Jon McCormack
Mark d'Inverno *Editors*

Computers and Creativity

Springer

Computers and Creativity

Jon McCormack • Mark d'Inverno
Editors

Computers and Creativity

Springer

Editors
Jon McCormack
Faculty of Information Technology
Monash University
Caulfield East, Victoria
Australia

Mark d'Inverno
Computing Department
Goldsmiths, University of London
New Cross, London
UK

ISBN 978-3-642-31726-2 ISBN 978-3-642-31727-9 (eBook)
DOI 10.1007/978-3-642-31727-9
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012946745

ACM Computing Classification (1998): I.2, J.5

Cover image by Fiammetta Ghedini

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Foreword

If I had to pick just one point out of this richly intriguing book, it would be something that the editors stress in their introduction: that these examples of computer art involve creative *computing* as well as creative *art.*

It's a happy—or perhaps an unhappy—coincidence that the book is going to press only a couple of weeks after the opening of David Hockney's one-man exhibition, "A Bigger Picture", at the Royal Academy of Arts in London.

A happy coincidence, in that such a famous traditional artist has chosen to link his most recent work with computers so publicly, and—according to the many favourable reviews—so successfully. This effectively puts paid to the all-too-common view that creative art cannot depend in any way on computers. For the "bigger pictures" that inspired the exhibition's title weren't produced with Hockney's oils, paintbrush, and easel, but with the help of computer software designed for colour graphics—specifically, Adobe's *Photoshop* and the iPad's app *Brushes.* Hockney used *Brushes,* for example, to move and blend colours, and—using his fingers on the tiny screen—to draw lines of varying thickness on the developing image.

An unhappy coincidence, however, in that Hockney's fame, alongside the critical success of this particular exhibition, will very likely lead people to think that his latest work is an iconic example of computer art. "And what's wrong with that?"— Well, Hockney's software is due to Adobe and Apple, not to Hockney himself. Even more to the point, the novelty, skill, and creativity—and the aesthetic judgements— evident in the huge images hanging on the Academy's walls aren't due to, or even reflected in, the software as such.

Photoshop can be—and has been—used to produce images of indefinitely many different styles. Years ago, to be sure, Adobe's professional programmers created (sic) the then-novel code that would eventually enable anyone to buy it off the shelf and use it in their own art-making. But that code wasn't intrinsically connected with the specific nature of any of the artworks that would be produced with its help. That is, it involved no aesthetic judgements on its creators' part.

The computer art that's described in this book is very different. It's not merely computer-assisted (as Hockney's is), but computer-generated. In other words, the program—originally written by, or under the direction of, the human artist—is left to run with minimal or zero interference from the human being.

Sometimes, as in Harold Cohen's work, the program runs entirely by itself. The artworks that result are literally untouched by human hand—and, occasionally, untouched even by *post hoc* human choice, or selection. At other times, although the code "runs by itself" in the sense that it's not altered by human beings during the running process, *what it actually produces* depends partly on various sorts of interaction between the program and the human artist and/or observer. These interactions can range from bodily movements, through noises or temperature-changes caused by human beings, to conscious choices made by the observer in selecting certain images (or musical compositions) to be preferred over others. And of course, for *this* or *that* interaction to be possible, with *this* or *that* result, the code had to be created in the appropriate way in the first place. The program had to be aesthetically motivated, not just technically effective. Off-the-shelf software simply doesn't fit the bill.

As various chapters make clear, this raises many difficult questions about the locus of creativity in the overall human-computer system. And it makes the aesthetic appreciation of computer art more problematic than it is in the familiar halls of the Academy's current exhibition. In general, the more someone understands the processes involved in the production of an artwork (wielding a paintbrush, perhaps, or turning a potter's wheel), the better they are able to appreciate the artist's achievement. But the code, in computer art, is even less evident than the chemicals and brush-strokes of traditional fine art. Worse: even if the code were to be made evident, many people would find it hard, or impossible, to understand.

These points, and many others, are explored in this book. For its aim is not only to describe a wide range of computer art, but also to indicate the many philosophical and aesthetic problems raised by this new genre. The answers are hotly contested, so don't expect a calm consensus in the following pages.

One thing, however, is agreed: the computer, here, is being used by the human artist not as a mere tool, but as a partner (or perhaps a quasi-partner) in the creative endeavour.

Brighton, England — Margaret A. Boden
2012

Preface

Why Does Computing Matter to Creativity?

This book, *Computers and Creativity*, examines how computers are changing our understanding of creativity in humans and machines. It contains chapters from twenty-five leading researchers in this field, on topics ranging from machine-assisted art creation, music composition and performance to formal theories of creativity and the emergence of novelty in natural and artificial systems. Before introducing these contributions we thought it useful to reflect on why we feel this book is both timely and important.

In just a single generation, computers and information technologies have brought about seismic changes in the way we communicate, interact, learn and think. Yet while these technologies are now well integrated into the fabric of modern society, their operation, design, and potential is understood by relatively few people. This limited appreciation of computing might explain why there remains a general reluctance to see its practice as something creative, and computers as machines that present a radical new potential for extending our own creativity.

Whilst general society may not think of computing as being a creative enterprise, we find ourselves in a world where we are now dependent on computers in almost every aspect of contemporary culture. Computers have become an extension of ourselves and how we communicate and think, even changing the way we think. They form a complex network of dependencies around us, and are constantly and rapidly developing, ever expanding in their role as a dynamic cultural and creative partner.

However the majority of traditional computing education and training has struggled to keep abreast of these changes. In September 2011, Google chairman Eric Schmidt criticised UK education, claiming: "Your IT curriculum focuses on teaching how to use software, but gives no insight into how it's made. That is just throwing away your great computing heritage." Art and Science need to be brought back together if we are to better tackle the challenges this rich entanglement with technology brings. And that doesn't just go for art either, to be a successful sociologist, journalist or social entrepreneur, for example, a deeper understand of computing as a creative discipline is becoming increasingly indispensable.

Creativity is critical for our ability to function and change as a society. Yet until recently, the practice of computing has not formally situated itself around the exploration of creative artistic ideas. Rather it has been taught in the main from a scientific and engineering perspective, using data structures (how to represent data) and algorithms (how to process or manipulate data) to directly solve problems. One of the great challenges for computing is to achieve a fuller understanding of process and representations which are beyond those that are easily computable or even fully comprehensible by humans. Necessarily, human design of software requires reducing difficult and complex concepts to far simpler abstractions that can be practically implemented, in some cases even ignoring those aspects of a phenomena that are too complex to express directly in a program. One way to overcome this limitation is to design programs that are capable of initiating their own creativity—to increase their complexity and discover ways of interacting independently of human design. Yet people don't naturally think of creative expression in terms of formal algorithms, leading to a perceived gap between natural creative human expression and computation.

Despite these difficulties, a field known as "creative coding" has emerged as an artistic practice of rising popularity. Here, software is considered a medium for creative expression, and the field has been enthusiastically embraced by many artists, designers and musicians. Software undergoes development at a pace and complexity that far exceeds all prior tools humans have developed, so these practitioners see the computer as something more than a benign tool such as a chisel or paintbrush. However, many artists find their artistic expression limited by a lack of knowledge in how to program *creatively*. While social and information networks allow easy access to a vast repository of resources and examples, what is often missing is a cogent technical, historical and philosophical foundation that allows practitioners to understand the "how and why" of developing creativity with computers. We hope this book makes important contributions by engaging with these foundational issues.

It is our belief that we now need to embrace and support the new forms of creativity made possible by technology across all forms of human endeavour. This creativity is important because it provides opportunities that have not been previously available, and are necessary if we are to address the complex challenges we face in our increasingly technology-dependent world.

Many excellent titles that look at creativity in general already exist.[1] Similarly, many works on the technical or didactic aspects of creative coding can be found, and are becoming standard in many university computing and design departments. However, due to a growing interest in appreciating computing as a creative discipline, and as a means of exploring creativity in new ways, the time is right for an edited collection that explores the varied relationships between computers and creativity. This book differentiates itself from general books on creativity or artistic coding because it focuses on the role of computers and computation in defining,

[1]Here we would suggest titles such as the *Handbook of Creativity* (edited by Robert J. Sternberg, Cambridge UP, 1999) and Margaret Boden's *The Creative Mind: Myths & Mechanisms* (2nd edition, Routledge, London, 2004).

augmenting and developing creativity within the context of artistic practice. Furthermore, it examines the impact of computation on the creative process and presents theories on the origins and frameworks of all creative processes—in human, nature, and machine.

Many of the book's authors come from an interdisciplinary background. Indeed, the origins of this book arose from a 2009 seminar on interdisciplinary creativity organised by the editors (McCormack and d'Inverno) and Professor Margaret Boden (University of Sussex), held at Schloss Dagstuhl–Leibniz-Zentrum für Informatik in Germany (http://www.dagstuhl.de/09291). Participants included artists, designers, architects, musicians, computer scientists, philosophers, cognitive scientists and engineers. With such diversity you might wonder what, if anything, was able to be understood and discussed beyond the traditional interdisciplinary boundaries and misinterpretations. It turned out that everyone passionately supported the view that computers have a substantial role to play in developing new forms of creativity, and the value of better understanding creativity from computational models in all its varied guises.

This book will appeal to anyone who is interested in understanding why computers matter to creativity and creative artistic practice. It is a proudly interdisciplinary collection that is suited to both those with a technical or scientific background along with anyone from the arts interested in ways technology can extend their creative practice. Each chapter arose in response to group discussions at the Dagstuhl seminar, and has undergone extensive review and development over a sustained period since, leading to what we hope will be a seminal volume on this topic that will remain relevant for many years to come.

Summary of Contributions

The book is divided into four sections: Art, Music, Theory and an Epilogue. However, as we have tried to make each chapter self-contained, the reader may read chapters in any order if they wish.

Part I, *Art*, addresses the long-standing question of machine creativity: can we build a machine that is capable of making art? And not just art, but good or even great art. Art that is exhibited in major art museums, prized and respected for its creative brilliance. Since the earliest days of computing, the idea of a machine being independently creative has been challenged. As Ada Lovelace famously claimed, a computer cannot be an artist because a computer cannot *originate* anything. All the machine does is what it is told to do, so how can a machine be independently creative?

Of course these arguments are closely tied to the history of Artificial Intelligence (AI), a research effort now more than sixty years old. The most famous and celebrated example of a "creative painting machine" is the *AARON* system of Harold Cohen. Cohen's initial investigations followed the "GOFAI" (Good Old-Fashioned Artificial Intelligence) approach to automated painting, but over its forty year history has developed considerably, producing an impressive oeuvre of paintings in collaboration with its creator. Cohen remains reluctant to ascribe independent creativity to AARON and sees the software as an extension of his artistic process rather than an independent, autonomous creative entity (he also acts as a curator and filter, carefully selecting specific images from AARON's prolific output).

Simon Colton's *Painting Fool* (Chap. 1) is the 21st-century continuation of research pioneered with AARON. Colton's bold and ambitious goal is to build a computer painter recognised in its own right as an independent artist. He deftly uses a diverse array of methods from contemporary AI, and anticipates the use of many more if he is to achieve his goal. Like Cohen, this ambitious agenda may require a lifetime's work, and also similarly, Colton is not deterred by this prospect. His chapter also addresses a number of criticisms and philosophical issues raised in both the idea of creating a computer artist, and the exhibition and appreciation of paintings made by a machine.

The chapter by Jon McCormack takes a very different approach to the problem of machine creativity. He sees the processes of biological evolution as a creative algorithm that is eminently capable of being adapted by artists to allow a machine to originate new things. Importantly, these "new things" (behaviours, artefacts) were not explicitly stated by the programmer in authoring the program. Using ideas drawn from biological ecosystems, he illustrates the creative potential of biological processes to enable new kinds of machine creativity. Here the computer is able to discover new artistic behaviours that were not explicitly programmed in by the creator, illustrating one way in which Lady Lovelace's enduring criticism can be challenged.

Pioneering artist Frieder Nake has been working with computational art since the 1960s. Nake frames creativity as a "US American invention" and through a series of vignettes examines the processes of developing creative works from the earliest days of digital computer art. As one of the first artists to create work with computers, Nake is uniquely placed to appreciate and reflect on over 40 years of endeavour in this field. His evaluation of the work of Georg Nees, A. Michael Noll, Vera Molnar, Charles Csuri, Manfred Mohr, Harold Cohen and even himself is fascinating.

Both Nake and Cohen are highly sceptical about machines ever being autonomously creative, and this is explored in the final chapter of this section: a discussion on machine creativity and evaluation between Nake, Cohen and a number of other Dagstuhl participants. These informal, and sometimes frank discussions reveal the complexities and diversity of opinion on the possibility of developing machines capable of independent artistic creativity that resonates with human artists. This chapter has been included for both its insights and its historical significance in documenting a rare discussion between several of computer art's most experienced and significant practitioners.

Part II, *Music*, deals with issues related to computers, music and creativity. A major challenge for machine creativity is in musical improvisation: real time, live interaction between human and non-human performers. This not only sets challenges for efficiency and on-the-fly decision making, but also in articulating what encompasses musically meaningful interactions between players. The chapter by François Pachet draws on the concept of "virtuosity" as an alternative way of understanding the challenge of improvisation. Pachet aims to create a computational musician who, in its improvisational skill, would be as good as the best bebop jazz musicians. He describes in detail the construction of a system that is capable of competently improvising with, and challenging, professional jazz musicians. Many think of AI's most public successes as game playing (such as Deep Blue's defeat of world chess champion Garry Kasparov in 1997) or mathematical problem solving, but as demonstrated by a number of authors in this book, intelligent musical interaction with computers is now a real possibility.

The goal of musically meaningful interaction between human and machine performers is the basis of what has become known as "Live Algorithms". The chapter by Tim Blackwell, Oliver Bown and Michael Young summarises a series of frameworks for human-machine interaction and improvisation inspired by the Live Algorithms model. The authors detail the kinds of interactions necessary for musically meaningful exchanges to occur and document some recent projects and research in this area.

The idea of a computer as "creative partner" is a major topic of this book. In combination, how can humans and computers expand our creative consciousness? The chapter by Daniel Jones, Andrew Brown and Mark d'Inverno details how computational tools extend and modify creative practice: challenging old assumptions and opening up new ways to simply "be creative".

Rather than looking for a general theory of human creativity through the work of others, researcher and musician Palle Dahlstedt introspected deeply about his own creative processes. This has lead to his theory of how materials, tools and ideas all interact and affect the creative process in complex, layered networks of possibility. While the theory comes from a musical understanding, it is broadly applicable to any creative discipline based around computers and software.

Many artists working with computers do so at the level of writing their own code. Coding is a unique form of artistic endeavour, which is often poorly understood as it lacks the extensive mainstream critical analysis and heritage found in more traditional art practices. Alex McLean and Geraint Wiggins—both coders and composers—examine the special relationship between a computational artist and their programming environment. Borrowing the art idea of the bricolage, they examine how perceptions affect the creative process when working with code. It is interesting to compare the use of feedback processes discussed by McLean & Wiggins, Dahlstedt, Jones, Brown & d'Inverno in relation to the current design of creative software, which often does little to facilitate or enhance the types of feedback emphasised as crucial by these authors.

Personal- and practice-based understandings of creativity are contextualised next in Part III, *Theory*. As discussed in Part I, for any machine to be creative it is argued that it must have some way of evaluating what it is doing. Philip Galanter undertakes an extensive survey of methods used in computational aesthetic evaluation: considered a first step in designing machines that are able to produce aesthetically interesting output. Although the chapter focuses primarily on visual aesthetics, the techniques can be applied more broadly, and Galanter's chapter provides a distinctive and comprehensive survey for researchers entering this challenging field. Similarly, Juan Romero and colleagues look at perceptual issues in aesthetic judgement and discuss how a machine might take advantage of things like psychological models of creativity. Both these chapters provide a much-needed overview of the field that has previously been lacking.

While the computer has brought new creative possibilities for artists, designers and performers, computer science has challenged traditional definitions of creativity itself. Over the last two decades, Jürgen Schmidhuber has developed a formal theory of creative behaviour, one that he claims explains a wide variety of creative phenomena including science, art, music and humour. Schmidhuber sees creativity as the ability of an agent to create data that through learning becomes subjectively more compressible. What humans term "interesting" is a pattern (image, sculpture, poem, joke, etc.) that challenges our compression algorithm to discover new regularities from it. Similarly, the chapter by Alan Dorin and Kevin B. Korb challenges the long-held definition of creativity that relies on a concept of appropriateness or value. Dorin and Korb define a creative system as one that can consistently produce novel patterns, irrespective of their value. These definitions appear to accommodate a number of criticisms levelled at previous definitions of creativity. For example, that some discovery may lie dormant for decades or centuries before its "value" is recognised, or that aesthetic appreciation is a truly subjective thing. It is interesting to read these theories in light of the dialogue of Chap. 4.

A different approach is taken by Oliver Bown, who distinguishes two fundamentally different kinds of creativity: generative and adaptive. The main distinction is the teleology of each – generative creativity is not goal-directed, adaptive creativity is. Bown also looks at the role of social processes in determining creativity often (mistakenly) ascribed exclusively to individuals.

Finally Peter Cariani presents his theory of emergent creativity, which like Schmidhuber, he has been working on for over two decades. Cariani shows how new informational primitives arise in natural systems and presents a detailed and ambitious framework for developing creatively emergent artificial systems.

Throughout this book you will find many different definitions of creativity and opinions of what (if any) level of autonomy and creativity might be possible in a machine. For example, Nake and, to an extent, Pachet downplay the importance of creativity in individuals. In Pachet's case, he demonstrates a system that can competently improvise with professional jazz musicians to illustrate how virtuosity, rather than creativity, is the predominate factor in musical improvisation. In a sense Pachet (a jazz musician himself) has been able to begin "reverse engineering" the complex motifs employed by famous jazz musicians such as Charlie Parker and Dizzy Gillespie. His challenge is to compute the "99 % explainable stuff" of jazz music and make serious inroads into the "1 % magic" that we might intuitively call human creativity. Computer scientists such as Schmidhuber see the way forward in terms of formal, computable definitions, since in theory they can be implemented and verified practically on a computer. Of course, any formal model of creativity requires abstractions away from the complexity of real human creative practice, so any such model could never fully represent it. Conceivably, neuroscience will eventually provide a full understanding of the mechanisms of human creativity, potentially overcoming current difficulties in validating computer models of human creative processes.

To conclude the book, Part IV, *Epilogue*, contains a short chapter that poses questions that were raised while editing this volume. As is often the case with new and emerging research fields, we are left with many more questions than answers and here what we consider the twenty-one most interesting and critical questions that this book has inspired are summarised. Competently answering these questions will take decades of research and investigation, the results easily filling many more volumes like this.

Whatever your views on creativity are, and whether you think a machine is capable of it or not, this book presents many new and inspiring ideas—wonderfully written and passionately argued—about how computers are changing what we can imagine and create, and how we might shape things in the future. We hope you enjoy reading *Computers and Creativity* as much as we have enjoyed producing and editing it.

Melbourne, Australia and London, England Jon McCormack and Mark d'Inverno
2012

Acknowledgements

First, we would like to express our sincere gratitude to all the authors for their patience and dedication to this project. We are very grateful to all of them for the quality and insights of their contributions and their willingness to enter into a long process of review. In this book each chapter was peer reviewed by two independent reviewers in addition to our review as editors and we would like thank the reviewers (who include many of the authors) for their constructive and thorough reviews.

We would also like to acknowledge Schloss Dagstuhl–Leibniz-Zentrum für Informatik in Germany and all the participants at the seminar we organised in 2009, where the genesis of this book was formed. Even though not all were able to contribute a chapter, we're sure that their influence and ideas from the seminar will have found their way into many of the contributions to this volume.

We also thank our universities, Goldsmiths, University of London and Monash University, for supporting the editing and production of this volume. Indeed Goldsmiths has been a wonderfully inspiring place to develop many of the ideas around creativity and computing which is home to Mark and where Jon is a visiting research fellow. Much of the research and teaching at Goldsmiths is aligned with the spirit of this book in understanding the relationship between technology and creativity. We acknowledge the support of The Centre for Research in Intelligent Systems, and the Centre for Electronic Media Art (CEMA), Monash University, who provided funds and assistance for the original seminar. Fiammetta Ghedini did an excellent job designing the cover image. We would also like to thank our publisher, Springer, and in particular Ronan Nugent for his invaluable support and assistance in seeing this book through into print. We really enjoyed working with Margaret Boden in co-organising the Dagstuhl seminar and would like to thank her especially for writing the Foreword to this book—her influence is abundantly clear in so much of the work presented in the chapters that follow.

Finally, we dedicate this book to our families: Julie, Imogen, Sophie, Melly, Felix, Olive and Iris.

Contents

Contributors

Tim Blackwell Department of Computing, Goldsmiths, University of London, London, UK

Tim Blackwell is a senior lecturer in Computing at Goldsmiths, University of London. He has degrees in physics, theoretical physics and computer science and has researched a wide range of subjects including quantum field theory, condensed matter theory, computer music, digital art and swarm intelligence. He is well known for the application of swarms to improvised music, and his Swarm Music system has been the subject of numerous articles, radio programmes and a Discovery Channel documentary. His work in computational swarm intelligence has focused on dynamic optimisation problems, bare bones and discrete recombinant swarms and theoretical analysis of particle swarm optimisation. More recent work includes modelling animal social learning in spatial environments. He was Principal Investigator for the EPSRC funded Live Algorithms for Music research network, and was joint Principal Investigator for the EPSRC Extended Particle Swarms project.

Oliver Bown Design Lab, Faculty of Architecture, Design and Planning, University of Sydney, Sydney, NSW, Australia

Oliver Bown is an electronic musician, programmer and researcher in computing, evolutionary and adaptive systems, and music. He completed his PhD from Goldsmiths, University of London, in 2008 studying the evolution of human musical behaviour using multi-agent simulations, under the supervision of Geraint Wiggins and Tim Blackwell. From 2008 to 2010 he worked at the Centre for Electronic Media Art with Jon McCormack on the Australian Research Council funded project, *Computational Creativity, an Ecosystemic Approach*. His electronic music projects include the duo *Icarus*, the improvisation collective *Not Applicable* and the Live Algorithms for Music research group.

Andrew R. Brown Queensland Conservatorium of Music, Griffith University, Brisbane, Australia

Andrew R. Brown is Professor of Digital Arts at the Queensland Conservatorium of Music, Griffith University in Brisbane, Australia. His research interests include live algorithmic music, the aesthetic possibilities of computational processes, and the design and use of creativity support tools. He is an active computer musician, computational artist, and educator.

David C. Brown AI Research Group, Computer Science Department, Worcester Polytechnic Institute, Worcester, MA, USA

David C. Brown is Professor of Computer Science and Professor of Mechanical Engineering at Worcester Polytechnic Institute. From 2001-2011 he was the Editor in Chief of the Cambridge UP journal *AIEDAM: AI in Engineering, Design, Analysis and Manufacturing*. His research interests include computational models of engineering design, and the applications of AI to Engineering and Manufacturing. In addition he has interest in Intelligent Interfaces and in Computational Creativity.

Paul Brown Informatics, University of Sussex, Brighton, UK

Paul Brown is an artist and writer who has specialised in art, science and technology since the late 1960s and in computational and generative art since the mid-1970s. His early work included creating large-scale lighting works for musicians and performance groups like Meredith Monk, Music Electronica Viva and Pink Floyd. He has an international exhibition record that includes the creation of both permanent and temporary public artworks and has participated in shows at major venues, including the Tate, Victoria & Albert Museum and ICA in the UK, the Adelaide Festival, ARCO in Spain, the Substation in Singapore and the Venice Biennale. He is an honorary visiting professor and artist-in-residence at the Centre for Computational Neuroscience and Robotics, University of Sussex, UK and also Australia Council Synapse Artist-in-Residence at the Centre for Intelligent System Research, Deakin University, Australia.

Adrian Carballal Faculty of Computer Science, University of A Coruña, Campus de Elviña, A Coruña, Spain

Adrian Carballal holds a BSc and a PhD in Computer Science from the University of A Coruña (Spain) were he works as post-doctoral research associate at the Department of Information Technologies and Communications. His main research interests include Image Processing and Computer Graphics.

Peter Cariani Department of Otology & Laryngology, Harvard Medical School, Boston, MA, USA

Peter Cariani's training and work has involved theoretical biology, biological cybernetics, and neuroscience (BS 1978, MIT, biology; MS 1982, PhD 1989, Binghamton University, systems science). His doctoral work developed a semiotics and taxonomy of self-constructing adaptive systems, and explored epistemic implications of evolutionary robotics. For the last two decades Dr. Cariani has investigated temporal coding of pitch, timbre, and consonance in the auditory system and proposed neural timing nets for temporal processing. He is currently engaged in auditory scene analysis research. He has served as external scientific consultant for the John Templeton Foundation on emergence and consciousness. He is a Clinical Instructor in Otology and Laryngology at Harvard Medical School and teaches courses on music perception and cognition at MIT and Tufts. www.cariani.com.

Harold Cohen University of California, San Diego, CA, USA

Harold Cohen was born in London in 1928 and moved to the USA in 1968. He is a practising artist, having represented the UK at the Venice Biennale, 1966, and represented the US at the Tsukuba World Fair, 1985. He has exhibited at the Tate Gallery, London, Museum of Modern Art, San Francisco, Stedelijk Museum, Amsterdam, Brooklyn Museum, New York, Computer Museum, Boston, and the Ontario Science Center, Toronto. His artworks are held in many private and public collections worldwide. He is currently a distinguished Emeritus Professor, UCSD, and the Founding Director, Center for Research in Computing and the Arts, UCSD. Cohen is widely known as the creator of *AARON*, a semi-autonomous art-making program that has been under continuous development for nearly forty years.

Simon Colton Computational Creativity Group, Department of Computing, Imperial College, London, UK

Simon Colton is a Reader in Computational Creativity in the Department of Computing at Imperial College London, and an EPSRC Leadership Fellow. He leads the Computational Creativity Group (ccg.doc.ic.ac.uk) where researchers study fundamental notions of creativity in software via applications to graphic design, mathematical discovery, video game design and the visual arts. Dr. Colton has written more than 130 papers, and his research has won national and international awards. He is a co-organiser of the AI and Games Industry/Academic Research Network (www.aigamesnetwork.org) and is best known for his HR mathematical theory formation software and *The Painting Fool* automated artist (www.thepaintingfool.com).

João Correia Department of Informatics Engineering, University of Coimbra, Coimbra, Portugal

João Correia received an MSc degree in Computer Science from the University of Coimbra (Portugal) in 2011. He is currently a PhD candidate at this university. He is also a researcher for the Cognitive Media Systems group at the CISUC—Centre of Informatics and Systems of the University of Coimbra. His main research interests include Computer Vision, Evolutionary Computation, Neuroscience and Machine Learning.

Palle Dahlstedt Dept. of Applied Information Technology, University of Gothenburg, Göteborg, Sweden

Palle Dahlstedt is a composer, improviser and researcher from Sweden. He is Associate Professor in computer-aided creativity at the Department of Applied IT, and lecturer in composition at the Academy of Music and Drama, at the University of Gothenburg. He holds MFA and MA degrees in composition from there, and a degree in composition from the Malmö School of Music, Lund University. He holds a PhD in design and media from Chalmers University of Technology. As a composer, he received the Gaudeamus Prize in 2001, and he performs regularly as an improviser on piano or electronics, alone and in various constellations.

Mark d'Inverno Department of Computing, Goldsmiths, University of London, London, UK

Mark d'Inverno holds an MA in Mathematics and an MSc in Computation from Oxford University and a PhD from University College London in Artificial Intelligence. He is Professor of Computer Science at Goldsmiths, University of London and for four years between 2007 and 2011 was head of the Department of Computing which has championed interdisciplinary research and teaching around computers and creativity for nearly a decade. He has published over 100 articles including books, journal and conference articles and has led recent research projects in a diverse range of fields relating to computer science including multi-agent systems, systems biology, art, design and music. He is currently the principal investigator or co-investigator on a range of EU and UK projects including designing novel systems for sharing online cultural experiences, connecting communities through new techniques in video orchestration, building online communities of music practice and investigating new ways of integrating London universities with London's creative and cultural sectors. During the final editing of this book he was enjoying a research sabbatical shared between the Artificial Intelligence Research Institute in Barcelona and Sony Computer Science Laboratory in Paris. He is a critically acclaimed jazz pianist and composer and over the last 25 years has led a variety of successful bands in a range of different musical genres.

Alan Dorin Centre for Electronic Media Art, Monash University, Clayton, Victoria, Australia

Alan Dorin is a researcher in electronic media art and artificial life at the Centre for Electronic Media Art, Monash University, Australia. His interests include animation and interactive media, biology (artificial and natural), computer science, history, music, philosophy, self-assembly, visual art and the links which bind these fields together. Alan received his PhD in Computer Science in 1999 from Monash University and degrees in Applied Mathematics (Monash 1991) and Animation and Interactive Media (RMIT 1995).

Philip Galanter Department of Visualization, Texas A&M University, College Station, Texas, USA

Philip Galanter is an artist, theorist, curator, and an Assistant Professor at Texas A&M University conducting graduate studios in generative art and physical computing. His research includes the artistic exploration of complex systems, and the development of art theory bridging the cultures of science and the humanities. Philip creates generative hardware systems, video and sound art installations, digital fine art prints, and light-box transparencies. His work has been shown in the United States, Canada, the Netherlands, and Peru. Philip has written for both art and science publications, and was a collaborating curator for Artbots 2002 and 2003, and COMPLEXITY.

Daniel Jones Goldsmiths, University of London, London, UK

Daniel Jones is a doctoral researcher at Goldsmiths, University of London. His research focuses on the relationships between complexity, evolution and social dynamics, and the wider affordances of complex systems towards creative activity. With an MA in Sonic Arts and an honours degree in Philosophy with Computer Science, he has a committed belief in cross-fertilisation across domains. He lectures on process music, mathematics, and digital sociology, and has worked with the National Institute for Medical Research and Dresden's Institute for Medical Informatics.

Kevin B. Korb Clayton School of IT, Monash University, Clayton, Victoria, Australia

Kevin Korb is a Reader in the Clayton School of Information Technology, Monash University. He received his PhD in philosophy of science from Indiana University in 1992. His research interests include Bayesian philosophy of science, causal discovery algorithms, Bayesian networks, artificial life simulation, and evolutionary simulation. He is the author of 'Bayesian Artificial Intelligence' (CRC, 2010) and 'Evolving Ethics' (Imprint Academic, 2010) and co-founder of the journal *Psyche*, the Association for the Scientific Study of Consciousness and the Australasian Bayesian Network Modelling Society. He was an invited speaker at the Singularity Summit (Melbourne, 2010).

Jon McCormack Centre for Electronic Media Art, Monash University, Caulfield East, Victoria, Australia

Jon McCormack is an Australian artist and researcher in Artificial Life, Creativity and Evolutionary Music and Art. He holds an Honours degree in Applied Mathematics and Computer Science from Monash University, a Graduate Diploma of Art from Swinburne University and a PhD in Computer Science from Monash University. He has held visiting research positions at the University of Sussex (UK), Goldsmiths, University of London and the Ars Electronica Future Lab in Linz, Austria. He is currently Associate Professor in Computer Science and co-director of the Centre for Electronic Media Art (CEMA) at Monash University in Melbourne, Australia. CEMA is an interdisciplinary research centre established to explore new collaborative relationships between computing and the arts.

Alex McLean Interdisciplinary Centre for Scientific Research in Music (ICSRiM), University of Leeds, Leeds, UK

Alex McLean is a PhD candidate on the Arts and Computational Technology programme at the Department of Computing in Goldsmiths, University of London. His research applies an embodied approach to representation in improvised computer music, informed by his practice as a live coding musician. He is a member of slub, a live coding band making people dance to their algorithms at festivals across Europe. He is active across the electronic arts, co-founding the long running *dorkbotlondon* meetings, and the award-winning *runme.org* software art repository.

Penousal Machado Department of Informatics Engineering, University of Coimbra, Coimbra, Portugal

Penousal Machado, PhD, teaches Artificial Intelligence and Computational Art at the University of Coimbra, Portugal. He is the author of more than 50 refereed journal and conference papers in these areas and co-edited the book "The Art of Artificial Evolution". His research interests include computational art, artificial intelligence and nature-inspired computation. He is the recipient of several scientific awards, including the prestigious award for Excellence and Merit in Artificial Intelligence (PremeIA) granted by the Portuguese Association for Artificial Intelligence. His work was featured in Wired magazine and exhibited at the Museum of Modern Art (MoMA), USA.

Frieder Nake University of Bremen, Bremen, Germany

Frieder Nake is a professor of interactive computer graphics at the computer science department of the University of Bremen. He also teaches digital media at the University of the Arts, Bremen. He holds a Diplom and a Dr.rer.nat. degree in mathematics from the University of Stuttgart. He is recognised as a pioneer of computer art, with his first exhibition held in 1965. Nake has contributed to the aesthetics and the theory of digital art for more than 40 years. He has recently focussed his work on the *compArt* database of digital art and the aesthetic laboratory at the University of Bremen.

François Pachet Sony CSL-Paris, Paris, France

Francois Pachet received his PhD degree and Habilitation from University of Paris 6, Paris, France. He is a Civil Engineer (Ecole des Ponts and Chaussées) and was an Assistant Professor in Artificial Intelligence and Computer Science, Paris 6 University, until 1997. He then set up the music research team at SONY Computer Science Laboratory, Paris, which conducts research in interactive music listening and performance. He developed several innovative technologies and award winning systems (*MusicSpace*, constraint-based spatialisation, *PathBuilder*, intelligent music scheduling using metadata, *The Continuator* for Interactive Music Improvisation). He is the author of over 80 scientific publications in the fields of musical metadata and interactive instruments.

Juan Romero Faculty of Computer Science, University of A Coruña, Campus de Elviña, A Coruña, Spain

Juan Romero, PhD, is an associate professor at the University of A Coruña, Spain. He is founder of the "Working Group in Music and Art" of EvoNet—the European Network of Excellence in Evolutionary Computing—and of the European Workshop on Evolutionary Art and Music (evoMUSART). He is the author of more than 30 refereed journal and conference papers in the areas of evolutionary computation and artificial intelligence, and editor of a special issue of the MIT Press journal "Leonardo" and of the book "The Art of Artificial Evolution", published by Springer in its Natural Computing Series.

Jürgen Schmidhuber IDSIA, University of Lugano & SUPSI, Manno-Lugano, Switzerland

Jürgen Schmidhuber is director of the Swiss Artificial Intelligence Lab IDSIA (since 1995), Professor of Artificial Intelligence at the University of Lugano, Switzerland (since 2009), Head of the CogBotLab at TU Munich, Germany (since 2004, as Professor Extraordinarius until 2009), and Professor SUPSI, Switzerland (since 2003). He obtained his doctoral degree in computer science from TUM in 1991 and his Habilitation degree in 1993, after a postdoctoral stay at the University of Colorado at Boulder. In 2008 he was elected a member of the European Academy of Sciences and Arts. He has published more than 250 peer-reviewed scientific papers on topics such as machine learning, mathematically optimal universal AI, artificial curiosity and creativity, artificial recurrent neural networks, adaptive robotics, algorithmic information and complexity theory, digital physics, the theory of beauty, and the fine arts.

Geraint Wiggins School of Electronic Engineering and Computer Science, Queen Mary, University of London, London, UK

Geraint A. Wiggins was educated in Mathematics and Computer Science at Corpus Christi College, Cambridge, and then to PhD in Computational Linguistics at the University of Edinburgh. He took a second PhD in Musical Composition at Edinburgh, in 2005. Since 1987, Geraint has been conducting research on computational systems for music, with a strong emphasis on cognitively motivated approaches. He was Professor of Computational Creativity in the Department of Computing in Goldsmiths, before taking a new position at Queen Mary College, University of London in 2011.

Michael Young Department of Music, Goldsmiths, University of London, London, UK

Michael Young is a composer with interests in improvisation, generative media and artificial intelligence. He is a Senior Lecturer and Head of the Department of Music, Goldsmiths, University of London. His work explores real-time interaction and learning systems for performance, including the "_prosthesis" series for soloist and computer (oboe, flute, piano, cello). In "Argrophylax" (2006) and "ebbs-" (2008) players negotiate a musical score in dialogue with interpretation-sensitive electronics. He has collaborated in a number of joint science/visual arts projects: "Exposure" (2010), is a real-time generative installation and exhibition exploring sonification, human habitation and environmental change www.ground-breaking.net. Personal website: www.michaelyoung.info.

Part I
Art

Chapter 1
The Painting Fool: Stories from Building an Automated Painter

Simon Colton

Abstract The Painting Fool is software that we hope will one day be taken seriously as a creative artist in its own right. This aim is being pursued as an Artificial Intelligence (AI) project, with the hope that the technical difficulties overcome along the way will lead to new and improved generic AI techniques. It is also being pursued as a sociological project, where the effect of software which might be deemed as creative is tested in the art world and the wider public. In this chapter, we summarise our progress so far in The Painting Fool project. To do this, we first compare and contrast The Painting Fool with software of a similar nature arising from AI and graphics projects. We follow this with a discussion of the guiding principles from Computational Creativity research that we adhere to in building the software. We then describe five projects with The Painting Fool where our aim has been to produce increasingly interesting and culturally valuable pieces of art. We end by discussing the issues raised in building an automated painter, and describe further work and future prospects for the project. By studying both the technical difficulties and sociological issues involved in engineering software for creative purposes, we hope to help usher in a new era where computers routinely act as our creative collaborators, as well as independent and creative artists, musicians, writers, designers, engineers and scientists, and contribute in meaningful and interesting ways to human culture.

1.1 Introduction

Computational Creativity is the term used to describe the subfield of Artificial Intelligence research where we study how to build software that exhibits behaviours deemed creative in people. In more practical terms, we investigate how to engineer software systems which take on some of the creative responsibility in arts and science projects. This usage of computers in the creative process differs from the majority of ways in which software is used, where the program is a mere tool to

S. Colton (✉)
Computational Creativity Group, Department of Computing, Imperial College, 180 Queens Gate, London SW7 2RH, UK
e-mail: sgc@doc.ic.ac.uk

J. McCormack, M. d'Inverno (eds.), *Computers and Creativity*,
DOI 10.1007/978-3-642-31727-9_1, © Springer-Verlag Berlin Heidelberg 2012

Fig. 1.1 An example picture from The Painting Fool's *Dance Floor* series

enhance human creativity. In contrast, within Computational Creativity research, we endeavour to build software which is independently creative, either to act as a collaborator with people, or to be an autonomous artist, musician, writer, designer, engineer or scientist. Some members of the Computational Creativity research community are interested in simulating creative processes to discover more about human creativity, while others are more interested in the intellectual challenge of producing autonomous creativity in software. Others simply want to generate more interesting art, music, text, mathematics or scientific hypotheses, but have chosen to do so by enabling the software to act as more than a tool for creative people.

Within the Computational Creativity Group at Imperial College, London,[1] we engage in various projects where we aim to build software for creative purposes. We work from an Artificial Intelligence perspective, whereby the solutions to problems we encounter while trying to engineer creative behaviour help to improve existing AI techniques, or lead to the invention of new ones. In a major project within the group, we are building *The Painting Fool* program, which we hope will one day be taken seriously as a creative artist in its own right. The project has been ongoing for around seven years, driven largely by the author, but with input in recent years by PhD students, MSc students and research associates in the group. An example image from one of the most recent projects with The Painting Fool—as described in Sect. 1.4.3 below—is given in Fig. 1.1. We plan to work on The Painting Fool in perpetuity, that is, for as long as it takes to satisfy the intellectual challenge of building an autonomously creative system.

In one respect, we have fairly low standards: an automated painter doesn't have to produce art at the level of a great master, an esteemed professional, an art school

[1] The web pages for which are here: ccg.doc.ic.ac.uk.

graduate or even a talented amateur artist. At least to start with, The Painting Fool's art has been rather naive and of little cultural interest, but as we progress with the project, we hope the value of the artworks it produces will increase. In another respect, however, we have fairly high standards: to be called a painter, our software must simulate a range of both cognitive and physical behaviours common to human painters. Such behaviours naturally include practical aspects such as the simulation of making paint strokes on a canvas. However, we are also intent on simulating such cognitive behaviours as the critical appraisal of one's own work and that of others; cultural and art-historical awareness; the ability to express ideas and moods through scene composition, choice of art materials and painting style; and the ability to innovate in artistic processes.

For some in the art world, there is a discernible resistance to using a computer in art practice, and this is naturally heightened when mention is made of the software acting as a creative collaborator or an independent artist. It is therefore an interesting challenge to gain some level of acceptance for AI-based art producing software within mainstream artistic communities. One problem has been that the majority of artworks produced by software with some level of autonomy have limited appeal and the pieces largely exist for decorative purposes. For instance, once any aesthetic pleasure and possibly some awe at the power of modern computing has worn off, it is difficult to have a conversation (in the cerebral, rather than the literal sense) with an image of a fractal, or indeed many of the generative artworks that artists and engineers regularly produce. Also, as a community of Computational Creativity researchers, there has been the assumption (or perhaps hope) that the artefacts produced by our software—poems, pictures, theorems, musical compositions, and so on—will speak for themselves. In certain creative domains, this may be the case. For instance, it is possible that people will laugh at a funny joke regardless of how it was conceived (with the caveat of controversial jokes: there is a big difference in our appreciation of a racist joke told by a person of that race and of the same joke told by a person of another race). However, in other domains, especially the visual arts, there is a higher level of interpretation required for consumption of the artefacts. In such domains, we have somewhat neglected the framing of the artefacts being produced by our systems. Such framing includes providing various contexts for the work, offering digestible descriptions of how it was produced, and making aesthetic, utilitarian or cultural arguments about the value of the work. Only with this extra information can we expect audiences to fully appreciate the value of the artefacts produced autonomously by computers via more interesting, more informed, conversations.

With The Painting Fool, we are building a system that aims to address the shortcomings described above. In particular, we are overcoming technical challenges to get the software to produce more stimulating artworks which encourage viewers to engage their mental faculties in new and interesting ways. These techniques include new ways to construct the paintings, in terms of scene composition, choice of art materials, painting styles, etc. In addition, they also include new ways to frame the paintings, in terms of providing text about the artworks, putting them into context, etc. We are pioneering Computational Creativity approaches which adhere to principles designed to not only produce culturally valuable artefacts, but also to frame them in a way which makes them more interesting to audiences. We have argued in favour of these principles from a philosophical viewpoint in (Colton 2008b), and we have used them practically in the construction of The Painting Fool. Having said that, we are still a long way off achieving our goal, and The Painting Fool is not yet producing pictures of particularly high cultural value, or framing its work in interesting ways.

The purpose of this chapter is to present the current state of The Painting Fool project, to discuss some of the cultural issues raised, and to describe some ways in which the project will continue. It is beyond the scope of this chapter to give a full technical specification of the software, which runs to around 200,000 lines of Java code, and relies on numerous other pieces of software. In place of these details, we refer to various technical papers where the functionality of the software is described at length. In Sect. 1.2, we present our work in some artistic, engineering and scientific contexts. By placing our work in these contexts, in addition to studying state of the art practices in Computational Creativity and through discussions with numerous people about building an automated painter, we have put together a number of guiding principles which we adhere to in building and framing our software. These guiding principles are outlined in Sect. 1.3. To best describe our progress so far with The Painting Fool project, in Sect. 1.4 we present the motivation, cultural and social issues, technical difficulties and research results for a number of projects carried out within this research programme. In Sect. 1.5, we describe future projects that we intend to pursue towards the goal of building our automated painter, and getting it accepted into society. We conclude in Sect. 1.6 by summarising the issues which arise from the project and calling for collaboration on this project.

1.2 The Painting Fool in Context

Our personal preference is to think of computing as an engineering discipline which uses both scientific and artistic methodologies to evaluate the computer programs we design and engineer. Theoretical scientific methodologies are often employed in order to have the ideas for software in the first place, and then experimental scientific methodologies are employed to test the performance of software in terms of ability, efficiency, reliability, etc. In the visual arts, software is largely employed as enabling tools for artists to produce pieces of art or design. Increasingly, especially in so-called *new media* circles, this has led to software itself being assessed in terms of its cultural and artistic impact. This is most obvious with interactive digital art, where audience members are (hopefully) intellectually stimulated through interaction with software. In addition, video games are increasingly being seen as artistic artefacts, and art students are regularly presenting software, such as novel web browsers, as art objects in their degree shows. It is fairly rare to see computer programs shown in galleries or exhibitions, unless they are interactive art pieces or they generate visual and/or acoustic artworks for visitors. This shouldn't be taken as an indication that general (i.e. non-art producing) software cannot be artistically valuable, because acceptance of novel media such as software is generally rather slow. For instance, only in 2009 did the Royal Academy in London first accept video installations for its Summer Exhibition.

The visual art software we produce in Computational Creativity circles largely fits into the mould of art-generating programs. However, there are two important differences which set our programs aside from others in this mould. Firstly, there is the underlying assumption that our software has some creative input to the process. Sometimes, this creative input is in terms of the automatic assessment (and rejection or selection) of artefacts. Alternatively, the input may be in terms of searching a space of artworks which can lead to surprising and interesting results. As a loose rule of thumb, and without wanting to be too exclusive, if the software is not making some kind of decision (whether about assessment and/or avenues of exploration), it is unlikely to be considered to be within the realm of Computational Creativity. Secondly, Computational Creativity software can itself produce new programs, hence it can act at a meta-level. Sometimes, this meta-level is not obvious, for instance, the majority of evolutionary art systems produce programs (genotypes) which are compiled or interpreted and executed to produce artworks (phenotypes). However, the user is normally only ever shown the phenotypes, and in this sense the evolutionary software can be seen as an interactive art installation which enables the user to produce aesthetically pleasing artworks. Occasionally, the meta-level is more obvious, for instance in Colton and Browne (2009) we evolved simple art-based games, where the user could click on a spirograph being drawn in order to affect the drawing process. If the user clicked correctly, the spirograph would be drawn to look like a given one, which provided the game playing challenge. In this instance, therefore, our evolutionary software was employed to produce new interactive programs for artistic and playful purposes.

The Painting Fool is a generative art program with decision making abilities that place it in the realm of Computational Creativity. It is definitively not a tool for artists to use, and hence we do not make it available as such. Rather, we see it as a fledgling artist that is being trained to act increasingly more creatively. In this sense, our automated painter most closely resembles the *AARON* program written by Harold Cohen and described in McCorduck (1991). This is a very well known system that has been developed over 40 years to produce distinctive figurative art, according to Cohen's unique guidance at both generative and aesthetic levels. The software has been through a number of stages of development: early versions produced child-like simple line drawings, and in a recent novel development, AARON has started producing abstract images.

It is over-simplistic to say that AARON has been developed to paint in the style of Cohen, as he has been influenced himself by feedback from the software, so the process has been somewhat circular. However, it is fair to say that AARON has not been developed to be independent of Cohen. Taken together as a package, Cohen and AARON represent one of the biggest success stories of AI art, both in terms of critical appraisal, acceptance (to a certain level) by the art world and sales to collectors and galleries. Part of this success can be attributed to AARON being seen as creative by various groups of people (although only in a guarded way by

Cohen). This is because it invents scenes from imagination, i.e., each scene that it paints is different, and doesn't rely on digital images, etc. Moreover, the scenes are figurative rather than abstract, hence the software uses information about the way the world works, which is not often the case with generative computer art. Cohen has used AARON to raise issues about the nature of software in art, which has further increased the interest in the artworks it produces. For instance, he ends (Cohen 1995) by asking:

> If what AARON is making is not art, what is it exactly, and in what ways, other than its origin, does it differ from the "real thing?" If it is not thinking, what exactly is it doing?

The main difference between The Painting Fool and AARON is in terms of the range of artistic abilities in the two pieces of software. For instance, the range of scene types that can be painted by AARON have differed somewhat over the years, but are largely limited to figurative scenes involving multiple people, pot plants and tables in a room. We discuss later how, when properly trained, The Painting Fool can produce pieces which depict a wide variety of scenes, including ones similar to those produced by AARON. The notion of training highlights another difference between the two systems. To the best of our knowledge, AARON has only ever been programmed/trained by Cohen, and that is not likely to change. In contrast, again as described below, we have built a teaching interface to The Painting Fool which enables artists, designers and anyone else to train the software in all aspects of its processing, from the way in which it analyses digital photographs to the way in which it constructs and paints scenes. We hope that allowing the software to be trained by artists will ultimately enable it to produce more varied and culturally valuable pieces. In particular, while The Painting Fool will be able to draw on and refer directly to some of the training it has been given, with knowledge of the styles of those who have trained it, the software will also be able to find its own path, its own style. In addition to this, we have enabled the software to interact with online information sources, such as Google and Flickr and social networking sites such as Facebook and Twitter, as described below and in (Krzeczkowska et al. 2010). Again, the hope is that the software can be trained to harness this information to produce more culturally interesting paintings.

Future versions of The Painting Fool will be further distinguished from AARON by their ability to critically appraise their own work, and that of others. Cohen provides aesthetic guidance to AARON by programming it to generate pieces in a certain style. However, he has not supplied it with any critical ability to judge the value of the pieces it produces—and ultimately, Cohen acts as curator/collaborator by accepting and rejecting pieces produced by the system. In contrast, not only do we plan for The Painting Fool to use critical judgement to guide its processing, we also plan for it to invent and defend its own aesthetic criteria to use within these judgements. For instance, it will be difficult, but not impossible, to use machine vision techniques to put its own work into art-historical context, and appraise its pieces in terms of references (or lack thereof) to existing works of art. In addition, we plan a *committee splitting* exercise, whereby we use crowd sourcing technologies such as Facebook apps to enable members of the public to score pieces produced by The

Painting Fool. The software will derive aesthetic measures via machine learning techniques applied to the results of this crowd-sourcing activity. However, we will attempt to avoid so-called "creativity by committee" by enabling The Painting Fool to concentrate on those pictures which are liked and disliked by the crowd in equal measures. In this way, its first learned aesthetic will hopefully be able to tell whether a piece it produces is divisive or not, which is a start. We plan to enable the software to invent and employ various aesthetic measures in a similar fashion.

It's our intention for the art produced by The Painting Fool to cause audiences to engage their mental faculties, and not just to think about the fact that the pieces were computer generated (although we advocate full disclosure of how the software produces its artwork, as described below). This will be achieved through the production of intrinsically interesting work, which includes emotional content, interesting juxtapositions, social commentary, and so on. A level of audience engagement will also be made through the software framing its pieces in various art-historical and cultural contexts, and providing titles and wall-text to this extent. There are already art generating programs which achieve a good level of engagement with audiences, some of which are described in other chapters of this book. Indeed, there are many different kinds of conversations one can have with generative art pieces. For instance, in some of the pieces produced by the *NEvAr* evolutionary art system described in (Machado and Cardoso 2002), rather than being driven by the user, the software uses built-in fitness functions to search for art generating programs. When viewing these pieces, one might be tempted to try and determine what the fitness function was and how it is expressed in the pieces, in much the same way that one might try and work out the aesthetic considerations going through a human painter's mind when they painted their works. Concentrating on evolutionary art, other projects have appealed to the (un)natural world to evoke feelings in audiences. In particular, often the fact that generated creatures (by for instance, Sims 1994) and flora and fauna (by for instance, McCormack 2008) look so similar, yet dissimilar to real examples of the natural world can lead to feelings of other-worldliness. McCormack's work on evolutionary decay takes this further, via appeal to the art-historical mainstay of mortality. Similarly, the software in the Mutator project as originally described by Todd and Latham (1992) produces organic forms which can be unnerving—possibly because of a similar effect to the well-known uncanny valley effect in video games, where automated non-player characters get too close to being human-realistic, causing an uncanny, uneasy feeling in many people.

All of these approaches produce works which are thought-provoking independently of their evolutionary genesis, and there are numerous other generative art projects which produce interesting and culturally relevant artworks, with Romero and Machado (2007) providing a good starting point for further reading. However, authors such as Galanter (2010) point out that often the most interesting aspect of evolutionary artworks is the process which went into producing them. This follows a long line of art movements where the principle innovation has been the production process (e.g. impressionism: painting *en plein air* to catch fleeting light conditions; pointillism: painting with complimentary dots of paint, as per colour theories, to produce more vivid pieces, etc.). We certainly advocate providing a description of the processes at work when software produces pieces of art. However, our position is that how the work is produced should form only one part of the framing of generative artworks, and they would be culturally more important if the pieces themselves offered a reason for audiences to think about certain issues, or if they invoked certain feelings or moods.

Another context within which our project can be seen is that of the graphics subfield of *Non-Photorealistic Rendering* (NPR). Here, the emphasis is on producing software which simulates natural media such as paints, pencils, canvases, pastels, and their usage in paint strokes, filling regions of colour, etc. Much of the pioneering work in this area has ended up in software such as Adobe Illustrator, which give artists new digital tools and media to work with. As a good example, James Faure-Walker (2006) mixes simulated paint with real paint in his art practice. NPR software is designed along solid software engineering and Human-Computer Interaction lines to be useful and reliable tools for artists and designers. Moreover, given that the consumers of such software are largely within the creative industries (and hence possibly perceived to be worried about creative software taking over some of their responsibilities), there have occasionally been mistakes of judgement from NPR experts keen to downplay claims of creativity in their software. In particular, in a standard NPR textbook, Strothotte and Schlechtweg state that:

> Simulating artistic techniques means also simulating human thinking and reasoning, especially creative thinking. This is impossible to do using algorithms or information processing systems. (Strothotte and Schlechtweg, 2002, p. 113)

It is difficult to tell whether this statement is denying the subfield of Computational Creativity research, or the entire field of Artificial Intelligence. In any case, the statement attempts to reinforce the myth that creativity is beyond scientific study, which is one of the main issues addressed within creativity studies and Computational Creativity research in particular, as addressed most vocally by Boden (2003).

Other NPR researchers are more enlightened, however, and supply their techniques with more intelligent abilities. For instance, with their saliency-adaptive painting research, Collomosse and Hall (2006) enabled their NPR system to determine the most important regions in an image using an evolutionary search. This enabled the production of painterly renditions of digital images with special attention paid to the most salient regions, which is more in line with the way in which painters understand the content of the pictures they are painting.

To summarise our placing of The Painting Fool project into various contexts, we observe that it is generative art software which has evolutionary search and non-photorealistic rendering abilities, in addition to the ability to construct scenes in a similar fashion to AARON. It is being engineered and further trained to transcend most generative art projects by addressing higher level artistic behaviours such as critical ability and cultural awareness. As such, it is designed not as a tool for artists to employ, but rather as a creative collaborator, or even an independent artist.

1.3 Guiding Principles

The building of creative systems requires overcoming numerous technical problems both of a general nature and in the particular domain within which the system works. Given that the results arising from such systems are ultimately for general consumption, building AI systems to create culturally interesting artefacts also requires a certain amount of framing and promotion. Over the years, we have developed the following seven principles to which we try to adhere when building creative software and which we hope may be useful frames of reference for other people building similar systems. They stand as a paradigm within which to build, test, employ and promote the output of creative software.

1.3.1 Ever-Decreasing Circles

We start with the observation that it is much easier to put together artificially intelligent systems if we have something concrete to work towards, especially when there is a general and workable theory of human intelligence to guide us. This has led to a somewhat unspoken notion in Computational Creativity that we should be looking towards research about human creativity for guidance on how to get computers to behave creatively. While such natural creativity research influences Computational Creativity research to some extent, our efforts in building creative software similarly influences our understanding of creativity in general. So, we shouldn't wait for philosophers, psychologists, cognitive scientists or anyone else to give us a workable impression of what creativity is. We should embrace the fact that we are actually undertaking research into creativity in general, not just computer creativity. Hence, we should continue to build software which undertakes creative tasks, we should study these systems, and we should help in the goal of understanding creativity in general. In this way, there will be ever-decreasing circles of research where we influence the understanding of natural creativity, then it influences our research, and so on until we pinpoint and understand the main issues of creativity in both artificial and natural forms.

1.3.2 Paradigms Lost

The problem solving paradigm in AI research is well established and dominant. It dictates that when an intelligent task needs to be automated, we immediately ask the same questions: Does it involve proving something?; Does it involve generalising a pattern?; Does it involve putting together a plan? and so on. If it is possible to answer yes to any of these questions, then the task is pigeonholed forever as a theorem proving problem, or a machine learning problem, or a planning problem, etc. This often means that the original aim of the task is lost, because only researchers in the designated area will work on automating approaches with respect to particular problems. As people, we don't solve the problem of writing a sonata, or painting a picture, or penning a poem. Rather, we keep in mind the whole picture throughout, and while we surely solve problems along the way, problem solving is not our goal. The main push to resurrect the lost paradigm of artefact generation—whereby the production of culturally important artefacts is the point of the exercise—is coming from the Computational Creativity community, and we should educate the next generation of AI researchers in the need to embrace entire intelligent tasks which lead to the production of beautiful, interesting and valuable artefacts.

1.3.3 The Whole Is More Than a Sum of the Parts

We have observed that some of the more interesting pieces of software which undertake creative tasks are those where multiple systems have been combined. Certainly, the only systems we've personally built which might be called creative involve at least two pieces of AI software written for different tasks being brought together so that the whole is more than a sum of the parts. There is still a tendency to re-implement techniques to fit into the workflow of a creative system rather than investigating how existing AI software could be incorporated. Sub-tasks within creative systems are often being achieved sub-optimally with a bespoke system for, say, generalisation that could be solved with an off-the-shelf machine learning system. Similarly, we have seen some deductive tasks performed using a forward-chaining approach that would be laughed at by automated reasoning researchers. We should assume that anyone who has built software and made it available for others would be very pleased to see it used in a creative setting, and such usage might help attract more people to Computational Creativity research. It takes real effort to build systems which rely on other people's software, but the benefits are much greater, as the power and flexibility of the software vastly increases.

1.3.4 Climbing the Meta-mountain

Software is mostly a tool for humans to use, and until we can convince people that computer programs can act autonomously for creative tasks, the general impression will remain that software is of no more use to society than, say, a microwave. The main problem is that, even within Computational Creativity circles, we still build software that is intended to be used by, or at least guided by, us. A common way in which this manifests itself is that we let the software have some autonomy in the production of artefacts (which may involve the software assessing artefacts, for instance), but we retain overall creative responsibility by choosing which artefacts to present to the world. Moreover, a criticism that people often level at so-called creative software is that it has no purpose. That is, if people didn't run the software, analyse its output and publish the results, then nothing would happen—which is not a good sign that the software is autonomously creative. This is a valid criticism. However, it is one that we can manage by repeatedly asking ourselves: what am I using the software for now? Once we identify why we are using the software, we can take a step back, and write code that allows the software to use itself for the same purpose. We call this process climbing the meta-mountain. If we can repeatedly ask, answer, and code software to take on increasing amounts of creative responsibility, it will eventually climb a meta-mountain, and begin to create autonomously for a purpose, with little or no human involvement.

1.3.5 The Creativity Tripod

In many domains, in particular the visual arts, how an artefact is produced is very much taken into account when people assess the value of the artefact. This leads to a genuine, and understandable, bias towards human artefacts over computer generated ones, and this feeling is impervious to any Turing test, which demonstrates that people cannot tell the difference between human and computer generated artefacts when they are presented out of context. This isn't a fatal problem, though, as long as we are happy to manage the public's impression of how our software works. In our many dealings with (well meaning) critics of Computational Creativity, we have found that the main criticisms levelled at programs purporting to be creative is that they lack skill, or they lack appreciation, or they lack imagination. We should therefore manage these misconceptions by describing our software along these dimensions. Moreover, we should regularly ask ourselves: if I were to describe my software using this supporting tripod of creativity terms, where would the weakest link be? In identifying and addressing such weak links, we can build better software both in terms of what it is able to achieve practically, and what it appears to be doing.

As described in Colton (2008b), managing people's perception of creativity in software is as important as building more intelligent algorithms in domains where cultural, contextual and historical precedents play an important role. Hence, if you have software which doesn't appreciate its own work, or the work of others, or its subject material, etc., then you should write code which achieves this. If you have software which isn't particularly inventive, then you should implement some routines which could be described as imaginative, and so on. Using this tripod, we've managed to devise a baseline test for creativity in software which is defensible. We suppose that the software is regularly producing artefacts with certain behaviours being exhibited simultaneously or in sequence during the production process. If from these behaviours, one could genuinely be described as skillful, one could be described as appreciative, and one could be described as imaginative, then we argue that the software should be described as creative. There are two caveats here: firstly, this is not a prescription for creativity in people; secondly, this is a baseline test, i.e., it doesn't mean that the software is highly creative. Indeed, it is our responsibility to keep adding skillful, appreciative and imaginative behaviours so that the software is perceived as increasingly creative.

1.3.6 Beauty Is in the Mind of the Beholder

By changing the sentence “Beauty is in the eye of the beholder” to the one above, we want to emphasise that when people appreciate/buy artwork, the actual look of the finished piece is only one thing they take into consideration. Other things that occupy their mind may include details about the artist and their previous work, other pieces of art owned by the art appreciator, or they have seen in museums, whether the artwork will increase in value, etc. Most importantly, as argued previously (Sect. 1.3.5), people tend to take into account how a piece of art was produced when assessing the finished product. If no information pertaining to the production of an artwork is available, then people can fall back on general knowledge about the struggle artists have in taming paint on a canvas, and can try and reverse engineer the specifics of this from the paint strokes exhibited. These fallbacks are not available for software generated artefacts, as most people have little idea about how software works. Turing-test style experiments may seem attractive because it shows some level of success if the artefacts being generated by a creative system are vaguely comparable to those produced by people. However, computers are not humans, and this fact should be celebrated, rather than hidden through Turing tests. In the visual arts in particular, Turing-style tests ignore process and promote pastiche, both of which are done at great peril, as expanded on in Pease and Colton (2011).

We argue that Computational Creativity researchers should be loud and proud about the fact that our software is generating artefacts that humans might be physically able to produce, but might not have thought to actually bring into being. Many people have asked why The Painting Fool produces artworks that look like they might have been hand drawn/painted. It does seem like we are missing an opportunity to produce pieces that humans can’t produce, thus supplementing global art production, rather than producing more of what people are already good at producing. This is a valid point, which we address to some extent in Sect. 1.4.5 below. However, automatically producing images which can’t be produced by people is easy, but not necessarily enough to demonstrate creativity. We have largely chosen instead to aim at automatically producing images which look like they could have been produced by people (because they include figurative details, messages, intriguing references, skillful flourishes, etc.), but—importantly—have not yet been produced by people because no one has so far thought to do so. This has the advantage that audiences have a frame of reference, namely human painting, in which to appreciate the behaviour of the software. It is for this reason that The Painting Fool continues to produce images that look hand drawn. No self-respecting art school graduate wants to be mistaken for another artist, and should be horrified if they were mixed up with Picasso or Monet in a blind test. We should write software that similarly wants to produce uniquely interesting works of art, which are not confused with anyone else’s, whether human or computer.

Another reason we believe we should not hide the fact that the artefacts are generated by a computer is because this kind of deception can set the computer up for a fall. For instance, imagine a Turing-tester saying: “And so, I can now reveal that these are the paintings produced by a recent art school graduate, and these are the paintings produced by…a convicted murderer”. While this example may be a little crass, it makes the point: by stating that the aim is to produce artefacts which look like they might have been created by a person, it explicitly lowers the value of the artefacts produced by computer. By using Turing-style tests, we are seemingly admitting that pastiche is all that we aim for. At best, this shows that we don’t understand one of the fundamental purposes of creative endeavours, which is to produce something interesting which no one has produced before. In many domains, there is no right or wrong, there is only subjective impression, public opinion and the values of influential people in that domain. As there is no reason why we can’t change public opinion, there is no reason why we should compare our computer generated artefacts to those produced by people. We can change the mind of the beholder to more appreciate the value of the artefacts produced by our software, and in trying to do so, we can learn a lot about the general perception of creativity in society.

Taking all the above arguments into consideration, we advocate non-blind comparison tests of human and computer art, where full disclosure of the processes behind the production of each piece is given. It is not imperative that the software generated artefacts look like they could be physically human-produced, but it might help people to appreciate them. In such non-blind tests, if art lovers choose to buy computer generated art as much as human art, because the pieces they buy stimulate their mind as well as their eye, we can claim real progress in Computational Creativity.

1.3.7 Good Art Changes Your Mind

It is perhaps not useful to delve here into the debate about what is and what isn’t art. However, it is difficult to argue against the fact that some of the best scientific discoveries force us to think more about the Universe we inhabit, and some of the best works of art, music, and literature were explicitly designed to make their audience engage their brains more than usual. Sometimes, the artworks are designed to make most people engage their brains in roughly the same way, other times the artworks are meant to be interpreted in many different ways. Sometimes, the purpose is to engage people on a cognitive level, other times the purpose is to engage them on an emotional level. Given this, our software should produce artefacts with the explicit purpose of making the human audience think more. This can be achieved in a number of ways (disguise, commentary, narrative, abstraction, juxtaposition, etc.), and some of these are easier to achieve than others.

More than any other aspect of Computational Creativity research, this sets us apart from researchers in other areas of AI. In these other areas, the point of the exercise is to write software to think for us. In Computational Creativity research, however, the point of the exercise is to write software to make people think more. This helps in the argument against people who are worried about automation encroaching on intellectual life: in fact, in our version of an AI-enhanced future, our software might force us to think more rather than less. Note further that there are also powerful works of art which emphasise the phenomenological experience of the work, or which are best appreciated through types of meditation. Hence, as well as hoping to increase mental activity with some of the artefacts that our software produces—which would literally change peoples' minds, whether in terms of an long-held opinion or temporary feeling—we should also hope to change the state of the minds of audience members. In either case, it is clear that, if we want Computational Creativity software to have impact on people, it should have individual and collective models of the minds of audience members.

1.4 Illustrative Projects

Our purpose with The Painting Fool project is to build an automated painter which is one day taken seriously as a creative artist in its own right. In order to do this, we have developed a roadmap based on the notion of climbing a meta-mountain as described above. That is, we have specified a sequence of very broad areas within which to research and implement improved versions of the software, by asking the question "what does a painter do?" and answering as follows:

1. Makes marks on a canvas
2. Represents objects and scenes pictorially
3. Paints scenes in different styles
4. Chooses styles in a meaningful way
5. Paints new scenes from imagination
6. Invents scenes for a purpose
7. Learns and progresses as an artist.

Naturally, this is a very subjective and quite naive breakdown of painterly progression, and is not intended for anything other than directing components within our research programme. As such, it serves its purpose well, and as we will see each component described below fits into one of the parts of this roadmap and contributes to the overall goal of producing an independent artist. For each component, our overriding aim is to implement more sophisticated versions of the software. However, determining what represents an improved program is often one of the more difficult aspects of the project, and we use both engineering standards and feedback from people who view the artworks produced to assess the level of success of each component. Hence, for the majority of the components, we present details of the motivations and aims; some implementation details; results from scientific testing of the software; and a gallery of images arising from running the software, along with some commentary on the value of the images, based on the feedback we have received.

In the sections below, the work on non-photorealistic rendering fits into the first three stages of the meta-mountain ascent given above, while the work on emotional modelling fits into stage 4. In the section on scene construction, we describe work towards stage 5 above, and the work on collage generation has been done with stage 6 in mind. Finally, the work on paint dances fits best into stage 7 of the above meta-mountain ascent.

1.4.1 Non-photorealistic Rendering

Starting with the notion of an artist simply making marks on a canvas, we implemented abilities for the software to simulate natural media such as pens, pencils, pastels, paints, brushes, papers and canvases. These tools allow the system to create the basis of an artwork, for example, applying paint strokes on a canvas, or making pencil marks on paper. To employ these simulations in useful ways, we implemented standard machine vision techniques to enable the software to identify regions of colour in a digital image, i.e., image segmentation. This led to a graphics pipeline whereby an image is first segmented into a set of paint regions, and then each region is filled and/or outlined a series of times in possibly differing styles. To enhance this pipeline, we enabled layers of different segmentations to be defined for possibly different areas of the original digital image (for instance, the user could define a region of the image as containing a person's eyes, and specify that it is segmented into more regions than the other areas, then painted differently). We were careful to ensure that each stage of the pipeline can be user-controlled by a fairly large number of parameters. For instance, image segmenting is controlled by 12 parameters, including: the number of segments required, the smallest segment area allowed, the amount of abstraction of the segment regions, whether to allow segments to have holes, etc. In addition, it is possible to map the colours in the segmentation to a set of colours from another palette, for example, art deco colours. The four different segmentations of a flower in Fig. 1.2 give some indication of the range of segmentations possible via different parameterisations.

Image segmenting and the simulation of natural media are all standard non-photorealistic rendering techniques, as described in textbooks such as (Strothotte and Schlechtweg 2002). We differed from the standard approach in one main respect, namely that we didn't implement different methods for different media types. For instance, the simulation of paints is usually treated differently to the simulation of pencils or pastels, etc. Instead, we saw each media type as applying varying amounts of pigment to a fixing medium such as paper or canvas. For instance, pencil strokes could be seen as paint strokes carried out with a very thin brush and a less than usual probability of the pigment sticking to the canvas (which gives the grainy look required). As the individual strokes are only ever used to fill in colour regions, we combined the parameterisation for the individual strokes with the parameterisation of the way in which the strokes were employed. There are 45 parameters controlling the way in which colour regions are rendered, and these include: aspects

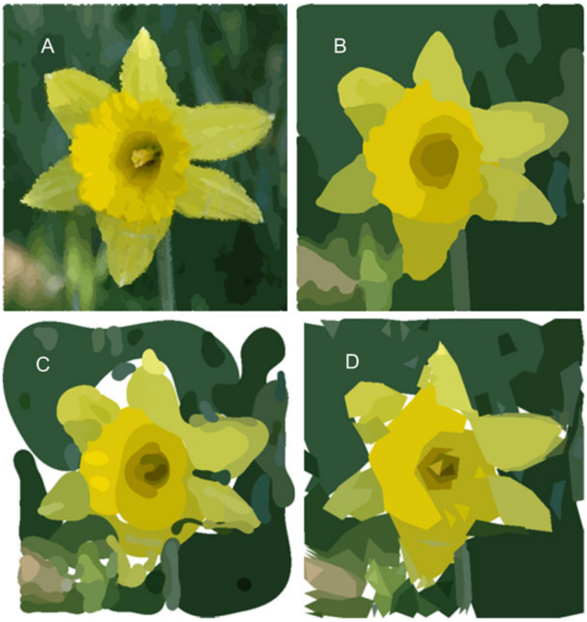


Fig. 1.2 Four different segmenting styles

of the natural media, e.g., brush and bristle size and colour variation; aspects of the individual strokes, e.g., length, taper, curvature; and aspects of the style in which the strokes are used, e.g., ever-decreasing circles versus parallel line fill, number of strokes required to fill a region, etc. The images in Fig. 1.3 give a flavour of the different types of strokes and filling mechanisms available, but this is only the tip of the iceberg—many more are possible.

Treating different media types with different models leads to more realistic looking paint strokes. However, for our purposes, treating all the natural media types and their usage as parameterisations of the same method essentially defines a search space of simulations, which has advantages. In particular, there are parts of the search space which fall in between particular natural simulations, such as paints and pencils. Hence, it was possible to specify ways of filling colour regions with unusual strokes which don't necessarily look like they could be naturally produced. Moreover, we were able to use search mechanisms to find natural media simulations for specific purposes, which we used to discover novel painting styles to enhance emotional content, as described below. Of course, these advantages would still be present if we had separate simulation models for each natural medium, but it would have seriously complicated the search mechanisms.

Full details of The Painting Fool's non-photorealistic rendering capabilities are available in Colton et al. (2008). In terms of the wider project, having the ability to turn images into painterly renditions of them enabled us to present some pictures in a group exhibition of computer generated art in 2007. An image from that exhibition is given in Fig. 1.4. It forms part of a series of eight images of buildings and cityscapes presented in the city series gallery at www.thepaintingfool.com.

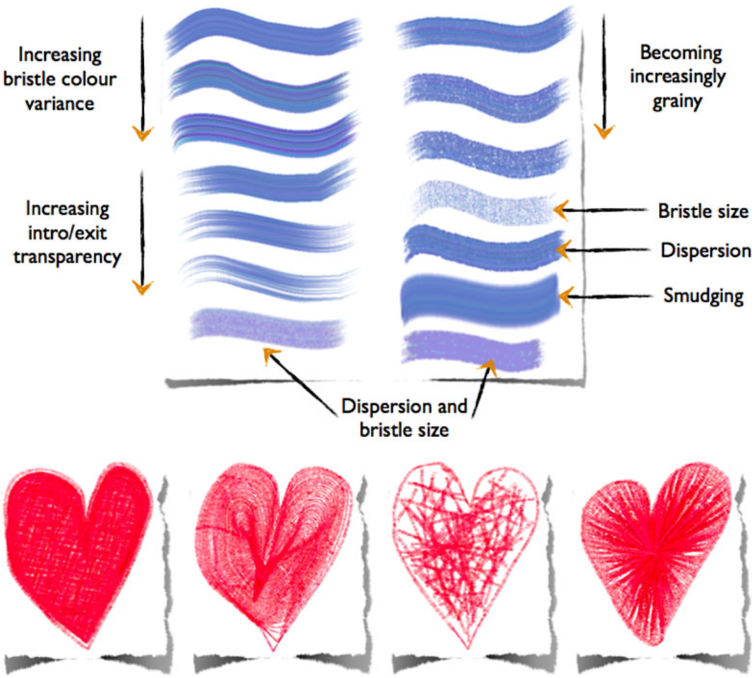


Fig. 1.3 Example paint strokes and region filling mechanisms

Fig. 1.4 A picture from the city series gallery by The Painting Fool

The exhibition gave us our first platform to introduce the notion of an independent software artist, which enabled us to identify the first of many non-technical issues related to this notion. In particular, the exhibition gained some press and media attention, and predictably led to stories about computers taking over the jobs of people in the arts. In one case, a news team told the story of computer generated art in a TV article prefixed by the phrase: "Is this some kind of hellish nightmare?" This is surely an over reaction, but it serves to highlight the public's perceived fear over the automation of human abilities, particularly in creative domains such as painting. In response to this, we argue that established artists have more to fear from the latest batch of art school graduates than from computers, because there will always be a premium in art for human involvement.

It does not diminish the potential for computer generated art to engage audiences in meaningful dialogues if we point out that many people appreciate art entirely because of the human aspect: art lovers want to get to grips with the mind and mood of the human behind the artwork. Hence, computer generated art may well occupy different niches to that produced by people, and so there is little to worry about in the automation of painting processes. More interestingly, the news team described above also interviewed Ralph Rugoff—director of the Hayward Gallery in London—and asked for his response to the notion of computer generated art. He pointed out that while software is good at playing games with fixed rules, such as chess, it is less obvious that computer programs can be playful in an artistic sense, where there are no such rules and where cultural knowledge plays an important role. Moreover, James Faure-Walker (another artist at the exhibition) pointed out that most of the research in non-photorealistic graphics was essentially photograph based, i.e., images are turned into painterly renditions. He added that this is rather a naive approach, and noted that an idea rather than an image should be the motivation for a piece of art. The issues raised by Rugoff and Faure-Walker led us to address the (lack of) imaginative aspects of the software, and ultimately provided the inspiration for the projects described under the scene invention and collage generation sections below.

1.4.2 Emotional Modelling

Human emotion plays an enormous role in the visual arts. Often, paintings are produced in order to convey the emotion of the painter, or to evoke a particular emotion in the viewer. In many cases, an important aspect of appreciating an artwork boils down to understanding the emotions at play. When building an automated painter, we have two choices with respect to emotional modelling. We could simply admit that computers are not human, and therefore any attempt for the software to simulate emotions or model the emotions of viewers would be facile and doomed to failure. In this case, we can emphasise that computer generated paintings can still evoke emotions in viewers without necessarily modelling human emotions, and that there are many other dialogues one can have with a painting other than trying to understand the emotional state of the painter who produced it. Given that we argue in the guiding principles given above that we should celebrate the difference between computers and people, it is certainly a defensible option to ignore emotion. However, this would miss an opportunity to use pioneering work from the field of affective computing, as described in Picard (2002), where software has been built to both simulate and detect human emotions. For this reason, we chose to implement some simple but foundational emotional modelling in The Painting Fool.

We first asked the question of whether we can train the software to paint in different styles, so that it can choose a particular style in order to heighten the emotional content of a painting. Note that this corresponds with part four of the meta-mountain described previously, i.e. choosing styles in a meaningful way. We worked on portraits of the actress Audrey Tatou as she portrayed Amélie Poulain in the film *Le Fabuleux Destin d'Amélie Poulain*. This source material seemed appropriate, as the film is largely about the emotional rollercoaster that Amélie finds herself on, and the actress portrays a full range of emotions during the film. Working with 22 stills from the film, we first annotated the images to specify where the facial features were, and then we repeatedly suggested painting styles to The Painting Fool. The descriptions of styles specified the level of abstraction to obtain through the image segmenting; the colour palette to map the regions of colour to; the natural media to simulate while filling/outlining the regions and the brush stroke style to employ while doing so. Largely through trial and error, we derived many of the styles by hand, by experimenting until the pictures produced were subjectively interesting. In addition to these hand-derived styles, we also enabled the software to randomly generate painting styles. Each time a style—whether randomly generated or derived by us—subjectively heightened the emotion portrayed by the actress in one of the stills, we recorded this fact. In this way, we built up a knowledge base of around 100 mappings of painting styles to emotions, with roughly half the styles provided by us, and the other half randomly generated (but evaluated by us).

Naturally, this tagging exercise was a very subjective endeavour, as all the emotional assessment was undertaken by us. Therefore, in order to gain some feedback about the knowledge base, we built an online gallery of 222 portraits produced from the 22 stills. The gallery is called Amélie's Progress and can be viewed at www.thepaintingfool.com. The portraits are arranged from left to right to portray emotions ranging from melancholy on the left to mild euphoria on the right. Sometimes, the emotion portrayed is largely due to the actress, but at other times, the painting style has heightened the emotional content of the piece. Hence, on a number of occasions, we find the same still image painted in different ways on both the left and the right hand sides of the gallery. An image of the entire gallery and some individual portraits are presented in Fig. 1.5.

The Amélie's Progress project raises some issues. In particular, we decided to make the web site for The Painting Fool read as if The Painting Fool is a painter discussing their work. This has been mildly divisive, with some people expressing annoyance at the deceit, and others pointing out—as we believe—that if the software is to be taken seriously as an artist in its own right, it cannot be portrayed merely as a tool which we have used to produce pictures. In addition, we chose to enable people working with The Painting Fool to see it paint its pictures stroke by stroke, and we put videos of the construction of 24 of the Amélie portraits onto a video wall, as part of the online gallery. This construction involves the sequential placing of thousands of paint strokes on a canvas. In another area of the web site, there are live demonstrations of paintings being constructed (as a Java Applet, rather than as

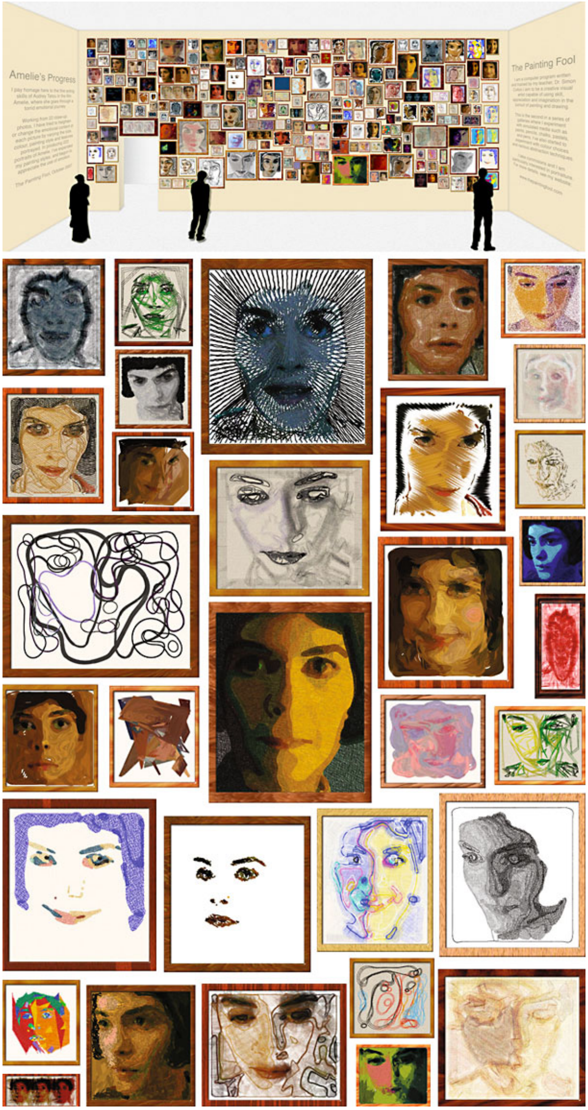


Fig. 1.5 An overview of the Amélie's Progress gallery and some sample portraits from it

a video of The Painting Fool at work). We have found that most people appreciate the painting videos, as they promote empathy with the software to some extent. While we know at least an approximation of the painting process for humans, in most cases—especially with the complex mathematical machinations of Photoshop filters—we do not know how software produces painterly images. Hence, seeing each simulated paint stroke applied to the canvas enables viewers to project effort and decision making processes onto the software. We have argued above that the process behind art production is taken into account when people assess the value of pieces of art, and we have much anecdotal evidence to highlight how evaluation of

Fig. 1.6 Example portraits using styles to heighten (L to R) sadness; happiness; disgust; anger; fear and surprise

The Painting Fool's pieces is increased after people see the videos of it at work. Of course, this approach was pioneered by Harold Cohen, as it has always been possible to view AARON at work, and AARON was an inspiration for us in this respect. Moreover, in discussions with Palle Dahlstedt about how software can better frame and promote their own work, he suggested that the artefacts produced by music and visual art systems should contain at least a trace of the construction process (see Chap. 8). In simulating paint strokes and showing construction videos, we achieve this with The Painting Fool.

One criticism of most image manipulation software is that it has no appreciation of the images it is manipulating. Hence a Photoshop filter will apply the same techniques to an image of kitten as it would to an image of a skyscraper, which clearly has room for improvement. To address this, and following on from the Amélie project, we addressed the question of whether The Painting Fool can detect emotion in the people it is painting and use this information to produce more appropriate portraits. Detecting emotion in images and videos is a well researched area, and we worked with Maja Pantic and Michel Valstar in order to use their emotion detection software (Valstar and Pantic 2006), in conjunction with The Painting Fool. The combined system worked as follows: starting with the sitter for a portrait, we asked them to express one of six emotions, namely happiness, sadness, fear, surprise, anger or disgust, which was captured in a video of roughly 10 seconds duration. The emotion detection software then identified three things: (i) the apex image, i.e. the still image in the video where the emotion was most expressed, (ii) the locations of the facial features in the apex image, and (iii) the emotion expressed by the sitter—with around 80 % accuracy, achieved through methods described by Valstar and Pantic (2006). It was a fairly simple matter to enable The Painting Fool to use this information to choose a painting style from its database of mappings from styles to emotions and then paint the apex images, using more detailed strokes on the facial features to produce an acceptable likeness. We found subjectively that the styles for surprise, disgust, sadness and happiness worked fairly well in terms of heightening the emotional content of the portraits, but that the styles for anger and fear did not work particularly well, and better styles for these emotions need to be found. Sample results for portraits in the six styles are given in Fig. 1.6.

The combined system was entered for the British Computer Society's annual Machine Intelligence Competition in 2007, where software has to be demonstrated during a 15 minute slot. The audience voted for the Emotionally Aware Painting Fool as demonstrating the biggest advancement towards machine intelligence, and we won the competition. More importantly for The Painting Fool project, we can now argue that the software shows some degree of appreciation when it paints. That is, it appreciates the emotion being expressed by the sitter, and it has an appreciation of the way in which its painting styles can be used to possibly heighten the emotional content of portraits.

1.4.3 Scene Construction

Referring back to the creativity tripod described in the guiding principles above, we note that through the non-photorealistic rendering and the emotional modelling projects, we could claim that the software has both skill and appreciation. Hence, for us to argue in our own terms that the software should be considered creative, we needed to implement some behaviours which might be described as imaginative. To do so, we took further inspiration from Cohen's AARON system, specifically its ability to construct the scenes that it paints. It was our intention to improve upon AARON's scene generation abilities by building a teaching interface to The Painting Fool that allows people to specify the nature of a generic scene and the software can then produce instantiations. As described below, we have experimented with numerous techniques in order to provide people with a range of methods with which to train the software. The techniques include AI methods such as evolutionary search and constraint solving approaches; exemplar based methods, where the user teaches the software by example; and third party methods such as context free design grammars for generating parts of scenes.

We describe a scene as a set of objects arranged prior to the production of a painterly rendition. This could be the arrangement of objects for a still life, the orchestration of people for a photograph, or the invention of a cityscape, etc. In practical terms, this entails the generation of a segmentation prior to it being rendered with simulated paints and pencils. This problem is most naturally split into firstly, the generation of the overall placement of elements within a scene—for instance the positions of trees in a landscape; and secondly, the generation of the individual scene elements—the trees themselves, composed of segments for their trunks, their leaves, and so on. While this split is appealing, we did not develop separate techniques for each aspect. Instead, we implemented a layering system whereby each segment of one segmentation can be replaced by potentially multiple segments repeatedly, and any segmentation generation technique can be used to generate the substitutions. This adds much power, and, as shown in the example pictures below, allows for the specification of a range of different scene types.

Our first exploration of scene generation techniques involved evolving the placement of scene elements according to a user-defined fitness function. Working with the cityscape scene of the tip of Manhattan as an *inspiring example* (in the words of Ritchie (2007)), we defined a fitness function based on seven correlations between the parameters defining a rectangle, with a set of rectangles forming the cityscape scene. For instance, we specified that there needed to be a positive correlation between a building's height and width, so that the rectangles retained the correct proportions. We similarly specified that the distance of a rectangle from the centre of

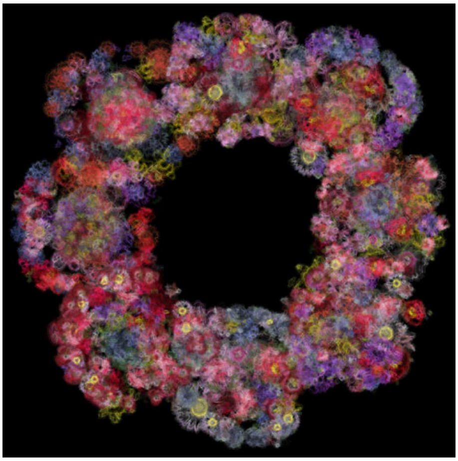


Fig. 1.7 A flower arrangement piece from the "Pencils, Pastels and Paint" gallery by The Painting Fool

the scene should be negatively correlated with the rectangle's height, width and saturation, so that buildings on the left and right of the scene were smaller and less saturated, leading to a depth effect. The genome of the individuals were the list of rectangles making up the scene. Crossover was achieved by swapping contiguous sublists, i.e. splitting the genomes of parents into two at the same point and producing a child by taking the left hand sublist from one parent and the right hand sublist from the other parent (and vice-versa for another child). Mutation was achieved by randomly choosing an individual with a particular probability, the mutation rate, for alteration. This alteration involved changing one aspect of its nature, such as position, shape or colour.

We experimented with one-point and two-point crossover, and with various mutation rates, population sizes and number of generations, until we found an evolutionary setup which efficiently produced scenes that looked like the tip of Manhattan (Colton 2008a). We turned each rectangle into a segment of a segmentation, and The Painting Fool was able to use these invented scenes as the subject of some pictures. Moreover, we used the same techniques to evolve the placement of flowers in a wreath effect, with the rectangle position holders replaced by segmentations of flowers. When rendered with pencil and pastel effects, these arrangements became two of the pieces in the "Pencils, Pastels and Paint" permanent exhibition, as described at www.thepaintingfool.com, with an example given in Fig. 1.7.

In an attempt to climb the meta-mountain somewhat, we realised that in defining the fitness function, we had ultimately performed mathematical theory formation. This suggested that we could employ our HR mathematical discovery system (Colton 2002), to invent fitness functions in our place. Using the same parameters required to define the original correlations (rectangle width, height, hue, saturation, brightness, and co-ordinates) as background information, and by implementing a new concept formation technique involving correlations, we enabled HR to invent new fitness functions as weighted sums of correlations over the parameters. For each fitness function, we calculated the fitness of 100 randomly generated scenes. If the

Fig. 1.8 Ten scenes generated for different invented fitness functions and two randomly generated scenes

average fitness was greater than 0.8, then it was likely that optimal fitness was too easy to achieve, and if it was less than 0.4, then it was likely that there were some contradictions in the fitness function. Hence, we only accepted fitness functions with an average for the 100 random scenes of between 0.4 and 0.8.

For each of ten acceptable invented fitness functions, we evolved a scene to maximise the fitness, and on each occasion, the scenes exhibited visually discernible properties. Moreover, two of the scenes genuinely surprised us, because the fitness functions had driven the search towards scenes which we didn't expect. In particular, for one fitness function, the fittest scene involved clumping together the rectangles in three separate centres (scene G in Fig. 1.8), and for another fitness function, the fittest scene had buildings placed on top of each other (scene C), which was not expected at all. The ten scenes arising from the fitness functions are given in Fig. 1.8, along with two randomly generated scenes, for comparison (R1 and R2). This approach to the invention and deployment of fitness functions is described fully in Colton (2008a). It raises the issue of software defining, employing and defending its own aesthetic considerations, something we will come back to in future work. It also highlights one of the accepted tenets of Computational Creativity research—that creative software should surprise its programmers.

Specifying correlation-based fitness functions for evolutionary scene generation worked well, but it had two main drawbacks: (i) for artistic purposes, sometimes the scene must fully adhere to some constraints, yet there is no guarantee that it will be possible to evolve a scene scoring 100 % for fitness, (ii) specifying a fitness function is not always a particularly natural thing to do and it would be better if someone using The Painting Fool's teaching interface were able to express their desires for a scene in a visual manner. To address these issues, we investigated the usage of constraint solving, whereby the requirements for a scene, or an element within a scene, are expressed by dragging, scaling and changing the colour of a set of rectangles. Following this, the constraints expressed in the scene are induced and translated into a constraint satisfaction problem (CSP, as described by Abdennadher

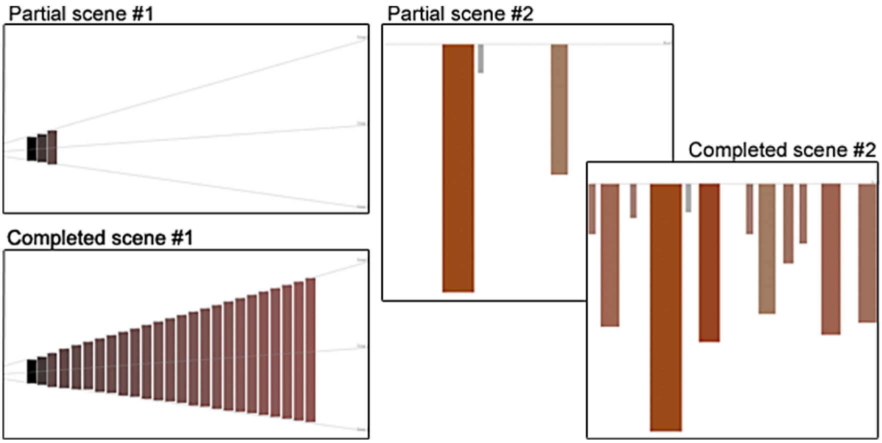


Fig. 1.9 Partial scenes provided by the user and completed scenes produced by solving the induced constraint satisfaction problem

and Frühwirth (2003)) and then the CSP is solved to give one or more instances of the scene which differ from the one defined by the user, while still satisfying all the required constraints. Full details of the implementation and our experimentation with it are given in Colton (2008c).

In summary, the user is able to visually express constraints involving: (a) the ranges of properties of rectangles, such as their co-ordinates, colours, dimensions, etc., (b) co-linearity of points on rectangles, (c) propositional notions describing pairs of rectangles, such as the constraint that if the width of rectangle 1 is greater than that of rectangle 2, then its height should also be greater, (d) correlations between the properties of a rectangle, and (e) constraints specifying overlap or disjointness of pairs of rectangles. The software then induces the constraints and asks the user to check whether each one is there by design, or has risen co-incidentally, in which case it can be deleted. At this stage, the user can also describe which aspects of the scene are to be generated randomly, for instance they can specify that the X co-ordinate of the rectangles should be chosen randomly from within a range. The constraints are then interpreted as a CSP for the Sicstus CLPFD solver (Carlsson et al. 1997). The variables of the CSP are the co-ordinates, dimensions and colour values of a set of rectangles, with the set size also specified by the user. Hence a solution to the CSP represents a scene of rectangles, and if a random element has been introduced, each scene will be different. To test the constraint solving approach, we worked with an inspiring example of trees in a forest, which ultimately led to the "PresidENTS gallery" as described below. The guiding scene and an example generated scene are provided in Fig. 1.9 for two constructions.

Unfortunately, for scenes of ten or more elements, we found that the constraint solver could take a prohibitively long time to find a perfect solution, and hence we re-integrated the evolutionary approach so that the fitness function could be defined as the number of singletons or pairs of scene elements adhering to the constraints. This means that the visual specification of the scene constraints can be used with a faster evolutionary approach, although the resulting scene may not fully satisfy all the constraints (which in scenes with more elements may actually be desirable).

To supplement the constraint-based and evolutionary approaches, we wanted the teaching interface to enable the user to simply draw an example of the scene or scene element that they wanted to specify, and for the software to use this as an exemplar in order to generate similar looking examples. To do this, we implemented a drawing interface that records key anchor points of each line drawn by a user. The anchor points are recorded as variables rather than fixed values, so they can vary within ranges in order to produce similar looking shapes in a scene. Additionally, we allow the user to specify the hue, saturation and brightness ranges within which the colour of each shape can vary, and to specify allowable linear transformations (such as translations and rotations) and non-linear transformations (such as perspective warping) that entire shapes, or even the entire scene can be subjected to.

To further supplement the scene generation abilities of the teaching interface, we integrated the CFDG generative art software,[2] and our own evolutionary art software (Hull and Colton 2007, Colton et al. 2011). The former system is able to generate representational and abstract artworks by using context free design grammars, and there are thousands of grammars available for use in art projects. The latter system is able to generate abstract art forms in a number of styles, including pixel-based (similar to fractals), particle based, and spirograph based (Colton and Browne 2009). Finally, the sixth scene generation method available within the teaching interface is to take a digital image and turn it into a segmentation, as described in the non-photorealistic rendering section above. We further enabled the image to be filtered before it is segmented, as per our *Filter Feast* software (Torres et al. 2008).

In addition to a screen for each of the segmentation generation methods described above, the teaching interface has a screen to describe how the different methods are to be used in layers to form the overall scene. It also has a screen to describe how the different elements of the scene are to be rendered using NPR techniques and a screen to describe how to generate paint dance animations (see below). The teaching interface is currently in beta development. While it is not yet ready for general usage, it is possible to define and render scenes. Given that we hope to attract other people to use the software to define their own pictures, it was important to provide example projects that produce interesting images. This has led us to the production of a series of galleries, collectively entitled: “Ever so Slightly…”. The rather strange name recognises the fact that it is not feasible that anyone will project a great deal of imagination onto software which is able to produce novel scenes using a template provided by people, but it may be possible to project a slight amount of imagination onto the software, and this is our aim.

There are currently four galleries in the series, named: “PresidENTS”, “Fish Fingers”, “After AARON”, and “Dance Floor”. An example from the “Dance Floor” series has been given in Fig. 1.1, and we give examples from the others in Fig. 1.10. The titles of the first two are fairly awful puns which reflect

[2] Available at www.contextfreeart.org.

Fig. 1.10 Pictures from the "PresidENTS", "Fish Fingers" and "After AARON" galleries in The Painting Fool's "Ever So Slightly..." exhibition

their content, and we won't spoil the fun of working out the wordplay here (see www.thepaintingfool.com). The fourth one reflects the influence that Cohen's AARON system has had on the scene generation aspects of The Painting Fool—as we see in the images shown, the pictures strongly reference the contents of the pictures produced by AARON, although we did not try to capture its visual style. Note that the human figures were produced by context free design grammars, and the abstract images on the walls of the room were similarly produced. The gradient effect of the ceiling and floor used a constraints approach to the generation of rectangles which were filled in using simulated pencils.

1.4.4 Collage Generation

The pictures produced in the "Ever So Slightly..." series represent a step in the right direction towards imaginative behaviour. However, looking at the meta-mountain we have described for The Painting Fool, the software needs to construct scenes for a purpose. Moreover, while the paintings in the series may be amusing and mildly thought-provoking as simple word/art puzzles, they are certainly not the most provocative of works. One aspect of the human painting process that is rarely simulated in computer art programatically is the ability to construct paintings to convey a particular message within a cultural context. We looked at using text and image resources from the internet as source materials for the production of artwork that might have a cultural impact (Krzeczkowska 2009). In particular, the software began by downloading headline news stories from the websites of the Guardian newspaper and other news sources. Using text extraction software (El-Hage 2009), based on the TextRank algorithm (Mihalcea and Tarau 2004), the most important nouns were extracted from the text of the news story. These were then used as keywords for searches in image repositories, including Google images and Flickr. The resulting images were juxtaposed in a collage which was turned into a segmentation, and the non-photorealistic rendering software from The Painting Fool was used to produce a painterly rendition of the subject material. With the exception of the text extraction software, the process was largely devoid of AI techniques, and this is something we plan to work on. However, the results were often remarkably salient. As an example, one morning, the software downloaded the lead story from the Guardian, which was

Fig. 1.11 Collage produced in response to a news story about the war in Afghanistan

covering the war in Afghanistan, and used images from Flickr to illustrate it. The final collage was quite poignant, as it contained a juxtaposition of a fighter plane, an explosion, a family with a small baby, a girl in ethnic headwear, and—upon close inspection—a field of war graves. The collage is given in Fig. 1.11.

In Krzeczkowska et al. (2010), we used this project to raise issues of intent in generative software. Usually, the intent for a piece is supplied by a human user, possibly through the expression of an aesthetic judgement and/or tailoring the content to fit the intent. However, with the Afghanistan collage, we were not users of the software in the traditional sense. Firstly, the software ran as a timed batch process, hence we didn't hit the start button. Secondly, we had no idea that the software would find a story about war, and thirdly, we had no idea which keywords it would extract or which images it would retrieve for the collage. Hence, it is difficult to say that we supplied much intentionality for the collage, even though the painting does achieve a purpose, which is to force the viewer to think about the Afghanistan war. We argue that it is possible to say that the software provided some of the intent, but we acknowledge that this is controversial. In fact, as described in Cook and Colton (2011), it seems clear that five parties contributed intent in the construction of the Afghanistan collage: (i) the programmer, by enabling the software to access the left-leaning, largely anti-war Guardian newspaper, (ii) the software, through its processing, the most intelligent aspect of which was the extraction of the keywords, (iii) the writer of the original article, through the expression of his/her opinions in print, (iv) individual audience members who have their own opinions forming a context within which the collages are judged, and (v) the Flickr users whose images were downloaded to use in the collage, by tagging many negative images such as explosions and fields of graves with the kinds of neutral words that were extracted from the newspaper article, such as “Afghanistan”, “troops” and “British”.

We also used the collage generation project to raise the issue of playfulness in the software, as the collages would often contain strange additions, such as an image of Frank Lloyd-Wright’s *Falling Water building* being included in a collage arising from a news story about the England cricket team (see Krzeczkowska et al. (2010) for an explanation of how this happened). We don’t claim that the word “playful” should be used to describe the software, but it does show potential for this kind of behaviour.

1.4.5 Paint Dances

In many respects, it is an odd choice to build an automated artist that simulates traditional media such as paints and brushes. It would seem to be missing an opportunity for the software to invent its own medium of expression and exploit that. In future, we would have no problem with the software performing such medium invention, and we would see that as a very creative act. However, we argue that while the software is in its early stages it is more important for its behaviour to be understood in traditional artistic terms, so that its creativity can be more easily appreciated. In particular, as described in Sect. 1.3.6, we want the software to produce paintings that look like they could have been physically produced by a human, but simultaneously look like they would not have been painted by a person because they are so innovative in technique and in substance.

Notwithstanding this notion, we wanted to follow an opportunity for the software to work in a new medium, albeit one which is not far removed from painting, namely, *paint dances*. We define a paint dance as an animation of paint, pencil, or pastel strokes moving around a canvas in such a way that the strokes occasionally come together to produce recognisable subject material. We worked with portraits, in particular our subject material was images of the attendees of the 2009 Dagstuhl seminar on Computational Creativity. The technical difficulties involved are discussed in Colton (2010). In summary, to achieve the paint dances, we first implemented a way to tell which pairs of strokes from two different paintings were most closely matched. Following this, from the 60,000 pencil strokes used in the 32 portraits of the Dagstuhl attendees, we used a K-means clustering method to extract just enough generic strokes to paint each picture in such a way that the fidelity would remain high enough for a likeness to be maintained. The final technical hurdle was to write software to perform the animations by moving and rotating the paint strokes in such a way as to come together at the right time to achieve a portrait. We have so far completed two paint dances: “meeting of minds”, where pairs of pencil portraits are shown together, with the strokes meeting in the centre of the picture as they move to form two new portraits, and “eye to eye”, where each painted portrait is formed individually, with spare paint strokes orbiting the scene. The images in Fig. 1.12 show the stills from a transition in both pieces. These videos formed part of an group

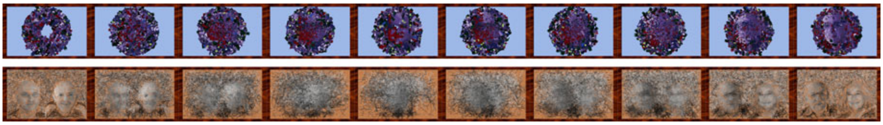


Fig. 1.12 A series of stills from the "Meeting of Minds" and the "Eye to Eye" paint dances produced by The Painting Fool

exhibition of work by members of the Sony Computer Science Laboratory, held at La Maison Rouge in Paris in September 2011 (see www.thepaintingfool.com for details).

This project raises the issue of software undertaking tasks that people cannot perform—in this case, the calculation of where to place thousands of strokes at thousands of time steps. As artists have been finding out since the advent of the computing age, getting software to undertake tasks beyond their capacity can enhance the space of possibilities in creative projects. Moreover, in an age of personalisation, such as personalised healthcare, personalised entertainment, etc., the only way to achieve personalised artistry will be to engage computers to produce artworks for people on an individual level. However, we note that the aesthetic decisions for the paint dance project were still undertaken by us, and not the software. Hence, in line with our guideline of climbing the meta-mountain, the next stage for this project will be to enable the software to invent animation techniques and make aesthetic judgements about what to apply while composing the paint dances.

1.5 Future Directions

We tend to undertake projects within quite large research programmes over a lengthy period of time, whereby multi-faceted intelligent systems are built by developing and combining our own techniques with existing AI systems (adhering to the principle that the whole is usually more than the sum of its parts). Ultimately, our aim is for the software we build to exhibit a range of intelligent behaviours while generating culturally interesting artefacts, all within a Computational Creativity setting. The Painting Fool is very much a work in progress, and we are not claiming that it should be taken seriously as an independently creative artist yet (and even if or when we do make that claim, it will be in reference to an artist of a low ability, at least to start with). In order to discuss future directions for The Painting Fool project, we address our current progress with respect to the guiding principles mentioned above, highlight areas where these principles can be used to assess the system negatively, and suggest ways forward.

With respect to the skill, appreciation and imagination requirements of the creativity tripod, it is easy to argue that the software is able to simulate the kinds of physical skills that are required of a painter. In addition, with the ability to produce paint dances, we can claim that the software has skills not possessed by human painters. There are numerous additional physical skills that we would like to simulate. For instance, the ability to produce smooth colour gradients through paint strokes is something that would certainly enhance the quality of the pieces produced by the software, and other physical simulations such as the use of a palette knife, the ability to spatter paint, etc., would all add value.

The software is more lacking in appreciative and imaginative behaviours than in skillful behaviours. We have argued that with the emotional modelling projects, the software is exhibiting some level of appreciation of its subject material and its painting styles. The fact that The Painting Fool cannot assess its own artworks and those of others against various aesthetic considerations is a major gap in its abilities. We have implemented abilities for the software to calculate objective measures of abstract art, for instance the location of symmetries, the distribution of colours, regions of high and low texture, etc. However, it is difficult to imagine training the software to appreciate works of art without essentially training it in a singularly subjective way (i.e. to become a "mini-me" for someone). In such circumstances, it would be difficult to argue against the software simply being an extension of the programmer, which we clearly want to avoid. An alternative approach is to build on the project to use mathematical theory formation to invent fitness functions, as described above. Rather than inventing a single fitness function, we hope to show that it is possible for software not only to invent more wide ranging aesthetic considerations, but also adhere to them, change them and discuss and possibly defend them within cultural contexts.

One aspect of this may involve getting feedback from online audiences, which will be used to tailor the image construction processes. However, as mentioned in Sect. 1.2, we are keen to avoid creativity by committee, which could lead to the software producing very bland pieces that do not offend anyone. Instead, we propose to use a committee splitting process, by which The Painting Fool will judge the impact that its pieces have on people, and choose to develop further only those techniques that led to pictures which split opinion, i.e. those which people really liked or really hated. Enabling the software to work at an aesthetic level will also involve endowing it with art-historical and cultural knowledge to allow it to place its work in context and to learn from the work of others. We are in discussions with artists and art educators about the best way to do this. In addition, we will draw on texts about creativity in both human and computer art, such as Boden (2010).

With the scene generation and collage generation abilities, we claim that the software is very slightly imaginative, and we aim to build on these foundations. Firstly, once the teaching interface is finished, we will deploy it by asking numerous people from academia, the arts, and from the creative industries to train the software. The payoff for people using the tool will be the production of pictures which hopefully characterise the ideas they had in mind and would paint if they were using more traditional methods. The payoff for The Painting Fool project will be a collection of potentially hundreds of scene descriptions. We plan to use these in order to perform meta-level searches for novel scenes in a more playful way than is currently possible. An important aspect of the teaching interface is the tagging of information, which is passed from screen to screen in order to cross-reference material for use in the overall scene construction. Hence, the software will in essence be taught about the visual representation of real-world objects and scenes (in addition, of course, to imaginary ones). We hope to build models which can subvert this information in playful and productive ways, building meaningful scenes different to those it was given.

We also intend to extend the collage generation approach described above, whereby online resources are employed as art materials. To this end, we have begun construction of a Computational Creativity collective, available at: www.doc.ic.ac/ccg/collective. The collective currently contains individual processes which perform creative, analytical, and information retrieval tasks, along with mashups, which combine the processes in order to generate artefacts of cultural interest. For instance, the project whereby news stories are turned into collages described above is modelled in the collective as a mashup of five processes which retrieve news stories, extract text, retrieve images and construct collages. The collective currently has processes which can link to Google, Flickr, the BBC, LastFM, Twitter and numerous other online sources of information. Our plans for the collective are ambitious: we hope to attract researchers from various areas of computing including graphics, natural language processing, computer music and audio, and AI to upload their research systems to expand the collective.

Systems built for Computational Creativity purposes such as The Painting Fool are beginning to have abilities of note in their particular domains of expertise, but rarely are they combined in order to increase the cultural value of their output. Hence we plan to paint pictures using the text produced by story generators like the Mexica system (Perez y Perez 2007) as input, and there is no reason why the pictures produced couldn't be used, for instance, as input to an audio generation system. This example highlights the masterplan for the collective, which is to have the output of one system continually consumed as the input to another system, thus providing a framework for experimentation with control mechanisms. In particular, the first mechanism we will implement will be based on Global Workspace Architectures, (Baars 1988), as per the PhD work of Charnley (2010). It is our hope that the increase in complexity of processing, coupled with the ability to access culturally important information from online resources will lead to more thought-provoking artefacts being generated.

An important part of our future research will be to continue to engage audiences on an artistic level, i.e., by organising exhibitions and reacting to the feedback we gain from such exercises. As an example, in April 2011, we exhibited artworks from The Painting Fool alongside those by traditional artist Eileen Chen, who worked in watercolours and graphic pens. The exhibition was entitled "No Photos Harmed/Growing Paths from Seed", and was a dialogue in which we explored the handing over of creative responsibility in artistic processes. In traditional painting approaches, with the subject matter and more pointedly with mediums such as watercolours, the artist has to occasionally go with the flow, hence doesn't retain full creative responsibility. We tried to emphasise the continuation of this with Computational Creativity projects, whereby such responsibilities are explicitly and wilfully handed over to software.

Fig. 1.13 The Dancing Salesman Problem piece from the "No Photos Harmed" exhibition (pictured with the author). The piece is so named, because a solution to an instance of the travelling salesman problem was used to generate the brush strokes

One of the pieces from The Painting Fool in this exhibition is presented in Fig. 1.13. By calling our part of the exhibition "No Photos Harmed", we emphasised the fact that computer generated art can be representational without requiring digital photographs as input. For instance, the figurative piece presented in Fig. 1.13 has context free design grammars rather than photographs of people at its heart. This was in direct response to James Faure-Walker's comment mentioned above that the inception of paintings is via ideas rather than images. Given that the theme of the exhibition was handing over responsibility, we were asked to estimate how much of the creative process was supplied by The Painting Fool. In answer, we guessed that around ten percent of the creativity in the process came from the software. This is a measure of autonomy in the software that we hope will increase in future versions.

1.6 Conclusions

The Science Museum in London once exhibited some interesting machines made from Meccano which were able to perform fairly complex differential analysis calculations. As these machines were built in the 1930s, the Meccano magazine from June 1934 speculated about the future in an editorial article entitled: "Are Thinking Machines Possible?" (Anon 1934). They couldn't have possibly known the impact that the computing age would have on society, but they were already certain about one thing—at the very end of the article, the author states that:

> Truly creative thinking of course will always remain beyond the power of any machine.

At this stage in the development of modern computing, it is neither a foregone conclusion that we will never see truly creative machines, nor is it obvious that we will one day be working alongside creative individuals which happen to be computers. It is our job as Computational Creativity researchers to investigate the possibilities for creative software, but we do not underestimate the difficulty of engineering such systems, and we do not underestimate the difficulties we will face in getting such software accepted on equal terms in society. We have described the overall aim of The Painting Fool project and some of the components we've completed along the way in order to climb a meta-mountain. The next stages will involve enabling the software to learn and develop as a creative painter, and this will raise further issues. One litmus test for progress, or even completion of the project, will be when The Painting Fool starts producing meaningful and thought-provoking artworks that other people like, but we—as authors of the software—do not like. In such circumstances, it will be difficult to argue that the software is merely an extension of ourselves.

The project has always been driven by feedback from people around some of the issues that we have raised here, and we always welcome collaboration in this respect. It seems that creativity in software—and perhaps in people—is usually marked negatively. That is, while there is no sufficient set of behaviours that a computer program must exhibit in order to be deemed creative, there is a necessary set of behaviours that it must exhibit to avoid the label of being uncreative. By adhering to the guiding principles described above in undertaking projects with The Painting Fool, we hope to manage people's perceptions of creativity, most obviously through (i) the notion of climbing the meta-mountain, whereby we describe the ways in which the creative responsibilities we have as programmers and users have been bestowed upon the software, and (ii) the notion of the creativity tripod, whereby we describe The Painting Fool's behaviours in terms of the skills it has, the appreciation that it exhibits and the imagination it exercises. It is our hope that one day people will have to admit that The Painting Fool is creative because they can no longer think of a good reason why it is not.

Acknowledgements We would like to thank the organisers and participants of the 2009 Dagstuhl seminar on Computational Creativity for their very interesting discussions, debates and performances, and for permission to use their images in the paint dances. We would also like to thank the Dagstuhl staff for their efforts in making the event very enjoyable. The anonymous reviewers for this chapter provided some excellent food for thought with relation to the arguments that we put forward. These comments have greatly enhanced our understanding of the issues, and have led to a much improved chapter. Many members of the Computational Creativity community have expressed support and provided much input to The Painting Fool project, for which we are most grateful. We owe a great deal of gratitude to the many collaborators who have contributed time and expertise on The Painting Fool and related projects. These include Anna Krzeczkowska, Jenni Munroe, Charlotte Philippe, Azalea Raad, Maja Pantic, Fai Greeve, Michel Valstar, John Charnley, Michael Cook, Shafeen Tejani, Pedro Torres, Stephen Clark, and Stefan Rüger.

References

Abdennadher, S., & Frühwirth, T. (2003). *Essentials of constraint programming*. Berlin: Springer.
Anon (1934). Are thinking machines possible? *Meccano Magazine, June*.

Baars, B. (1988). *A cognitive theory of consciousness*. Cambridge: Cambridge University Press.
Boden, M. (2003). *The creative mind: myths and mechanisms* (2nd ed.). London: Routledge.
Boden, M. (2010). *Creativity and art. Three roads to success*. London: Oxford University Press.
Carlsson, M., Ottosson, G., & Carlson, B. (1997). An open-ended finite domain constraint solver. In *Proceedings of programming languages: implementations, logics, and programs*.
Charnley, J. (2010). *A global workspace framework for combined reasoning*. PhD thesis, Department of Computing, Imperial College, London, UK.
Cohen, H. (1995). The further exploits of AARON, painter. *Stanford Humanities Review*, *4*(2).
Collomosse, J., & Hall, P. (2006). Salience-adaptive painterly rendering using genetic search. *International Journal on Artificial Intelligence Tools (IJAIT)*, *15*(4), 551–576.
Colton, S. (2002). *Automated theory formation in pure mathematics*. Berlin: Springer.
Colton, S. (2008a). Automatic invention of fitness functions, with application to scene generation. In *Proceedings of the EvoMusArt workshop*.
Colton, S. (2008b). Creativity versus the perception of creativity in computational systems. In *Proceedings of the AAAI spring symposium on creative systems*.
Colton, S. (2008c). Experiments in constraint-based automated scene generation. In *Proceedings of the 5th international joint workshop on computational creativity*.
Colton, S. (2010). Stroke matching for paint dances. In *Proceedings of computational aesthetics*.
Colton, S., & Browne, C. (2009). Evolving simple art-based games. In *Proceedings of the EvoGames workshop*.
Colton, S., Cook, M., & Raad, A. (2011). Ludic considerations of tablet-based evo-art. In *Proceedings of the EvoMusArt workshop*.
Colton, S., Valstar, M., & Pantic, M. (2008). Emotionally aware automated portrait painting. In *Proceedings of the 3rd international conference on digital interactive media in entertainment and arts*.
Cook, M., & Colton, S. (2011). Automated collage generation—with more intent. In *Proceedings of the international conference on computational creativity*.
El-Hage, J. (2009). *Linguistic analysis for the painting fool*. Master's thesis, The Computer Laboratory, University of Cambridge, UK.
Faure-Walker, J. (2006). *Painting the digital river*. New York: Prentice Hall.
Galanter, P. (2010). The problem with evolutionary art is .... In *Proceedings of the EvoMusArt workshop*.
Hull, M., & Colton, S. (2007). Towards a general framework for program generation in creative domains. In *Proceedings of the 4th international joint workshop on computational creativity*.
Krzeczkowska, A. (2009). *Automated collage generation from text*. Master's thesis, Department of Computing, Imperial College, London, UK.
Krzeczkowska, A., El-Hage, J., Colton, S., & Clark, S. (2010). Automated collage generation—with intent. In *Proceedings of the 1st international conference on computational creativity*.
Machado, P., & Cardoso, A. (2002). All the truth about NEvAr. *Applied Intelligence*, *16*(2), 101–118.
McCorduck, P. (1991). *AARON's code: meta-art, artificial intelligence, and the work of Harold Cohen*. New York: Freeman.
McCormack, J. (2008). Evolutionary L-systems. In P. Hingston, L. Barone & Z. Michalewicz (Eds.), *Design by evolution: advances in evolutionary design* (pp. 168–196). Berlin: Springer.
Mihalcea, R., & Tarau, P. (2004). TextRank: bringing order into texts. In *Proceedings of the conference on empirical methods in natural language processing*.
Pease, A., & Colton, S. (2011). On impact and evaluation in computational creativity: a discussion of the Turing test and an alternative proposal. In *Proceedings of the AISB symposium on AI and philosophy*.
Perez y Perez, R. (2007). Employing emotions to drive plot generation in a computer-based storyteller. *Cognitive Systems Research*, *8*(2), 89–109.
Picard, R. (2002). *Affective computing*. Cambridge: MIT Press.
Ritchie, G. (2007). Some empirical criteria for attributing creativity to a computer program. *Minds and Machines*, *17*, 67–99.

Romero, J., & Machado, P. (Eds.) (2007). *The art of evolution: a handbook on evolutionary art and music*. Berlin: Springer.

Sims, K. (1994). Evolving virtual creatures. In *Proceedings of SIGGRAPH* (pp. 15–22).

Strothotte, T., & Schlechtweg, S. (2002). *Non-photorealistic computer graphics*. San Mateo: Morgan Kaufmann.

Todd, S., & Latham, W. (1992). *Evolutionary art and computers*. San Diego: Academic Press.

Torres, P., Colton, S., & Rüger, S. (2008). Experiments in example-based image filter retrieval. In *Proceedings of the cross-media workshop*.

Valstar, M., & Pantic, M. (2006). Biologically vs. logic inspired encoding of facial actions and emotions in video. In *Proceedings of the IEEE international conference on multimedia and expo*.

Chapter 2
Creative Ecosystems

Jon McCormack

Abstract Traditional evolutionary approaches to computer creativity focus on optimisation, that is they define some criteria that allows the ranking of individuals in a population in terms of their suitability for a particular task. The problem for creative applications is that creativity is rarely thought of as a single optimisation. For example, could you come up with an algorithm for ranking music or painting? The difficulty is that these broad categories are shifting and subjective: I might argue that Mozart is more musically creative than Lady Gaga, but others may disagree. Objective, fine-grained ranking of all possible music is impossible, even for humans. I will show how reconceptualising the exploration of a creative space using an "ecosystemic" approach can lead to more open and potentially creative possibilities. For explanatory purposes, I will use some successful examples that are simple enough to explain succinctly, yet still exhibit the features necessary to demonstrate the advantages of this approach.

2.1 Creative Systems

In this book you will find a broad range of definitions of creativity. Dorin and Korb (Chap. 13), for example, emphasise a system's propensity to generate novelty irrespective of its perceived value, similarly Schmidhuber (Chap. 12) views creativity as a problem of learning information compression. Nake (Chap. 3) is more sceptical about formal computer models of creativity, seeing the popular concept of creativity today as "a US-American invention," one that may be considered as a *means* for activity, or as its *goal*. Pachet (Chap. 5) prefers to focus on "virtuosity", emphasising the thousands of hours that human artists must spend to master a discipline or instrument. Each of these views place a different emphasis on which qualities, properties or functions are important to understanding creativity precisely, and hence appreciating its worth or relevance in any given domain.

If we take Boden's popular definition—that creativity involves the generation of ideas or artefacts that are *new, surprising,* and *valuable* (Boden 2010)—then an

J. McCormack (✉)
Centre for Electronic Media Art, Monash University, Caulfield East, Victoria 3145, Australia
e-mail: Jon.McCormack@monash.edu

J. McCormack, M. d'Inverno (eds.), *Computers and Creativity*,
DOI 10.1007/978-3-642-31727-9_2,

interesting question to ask is: what are the mechanisms that enable this creativity? It appears likely that any such mechanisms are numerous and diverse. While creativity is commonly associated with the human individual, clearly societies and nature invent, too.

The psychologist David Perkins (1996) talks about "creative systems"; recognising that there are different mechanisms or classes of underlying systems that are all capable of producing creative artefacts. A creative system, in this view, is simultaneously capable of the production of novelty and adaptation in a given context. This suggests natural selection is a creative system, generating things like prokaryotes, multicellularity, eusociality and language, all through a non-teleological process of hereditary replication and selection. Social interaction is another creative system, having given rise to cultural customs such as shaking hands and a variety of grammatical forms in different human languages.

A number of authors have offered explanations of fundamental creative mechanisms based on evolution or evolutionary metaphors, e.g. Martindale (1999), Lumsden (1999), Dawkins (1999), Aunger (2002). George Basalla's *The Evolution of Technology* detailed a theory of technological evolution, offering an explanation for the creative diversity of human made artefacts: "*novelty* is an integral part of the made world; and a *selection* process operates to choose novel artifacts for replication and addition to the stock of made things" (Basalla 1998). Evolution has also played an important role in computer-based and computer-assisted creative systems (Bentley and Corne 2002), being able to discover, for instance, seemingly counterintuitive designs that significantly exceed any human designs in performance (Keane and Brown 1996, Eiben and Smith 2003, p. 10). Such results illustrate the potential of evolutionary systems to devise unconventional yet useful artefacts that lie outside the capabilities of current human creative thinking.

Defining a class of phenomena in formal, systemic terms allows for a transition to the computer. The purpose of this chapter is to look at what kinds of computational processes might qualify as "creative systems" in their own right. Here I draw my inspiration from natural systems, in particular evolutionary ecosystems. Biological evolution is readily accepted as a creative system, as it is capable of discovering "appropriate novelty". The computer science adaptation of evolution, a field known as *Evolutionary Computing* (EC), selectively abstracts from the processes of biological evolution to solve problems in search, optimisation and learning (Eiben and Smith 2003). It is important to emphasise *selectively abstracts* here, as only certain components of the natural evolutionary process are used, and these are necessarily highly abstracted from their physical, chemical and biological origins, for both practical and conceptual reasons. In the case of designing a creative system, the challenge is somewhat different than that of standard EC: understanding how a process that is creative in one domain (biology) can be transformed to be creative in another (e.g. the creation of art) requires different selective abstractions.

Generating the adaptive novelty exhibited in creative systems can be conceptualised as a process of exploration through a space of possibilities, searching for regions of high creative reward. Perkins (1996) uses the metaphor of the "Klondike space"—*Gold is where you find it.* Perkins identified four basic problem types in

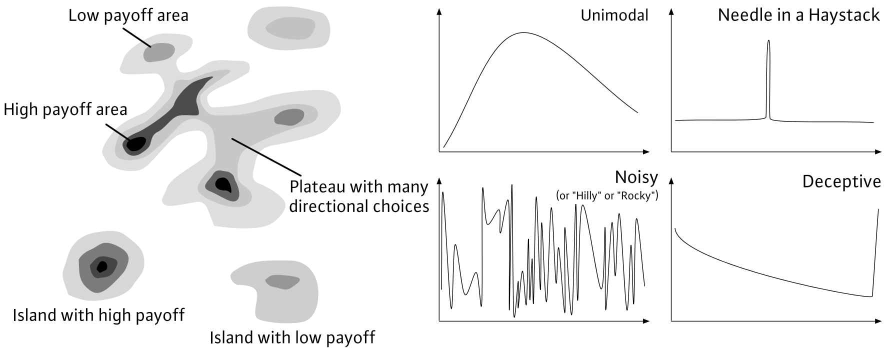


Fig. 2.1 Illustrative diagram of "Klondike spaces" (*left*, after Bell 1999) and, characterisation of archetypical search spaces in Evolutionary Computing (*right*, after Luke 2009)

the creative search of a conceptual space (Fig. 2.1, left): (i) *rarity:* viable solutions are sparsely distributed in a vast space of non-viable possibilities; (ii) *isolation*: places of high creative value in the conceptual space are widely separated and disconnected, making them difficult to find; (iii) *oasis*: existing solutions offer an oasis that is hard to leave, even though better solutions might exist elsewhere; (iv) *plateau*: many parts of the conceptual space are similar, giving no clues as to how to proceed to areas of greater creative reward.

This classification is similar to archetypical search and optimisation problems encountered in EC (Fig. 2.1, right), where algorithms search for optima in what are often difficult phenotypic spaces (Luke 2009). For example, "rarity" corresponds to "Needle in a haystack", "oasis" to "Deceptive". Noisy landscapes are particularly problematic, where evolutionary methods may do no better than random search.

Knowing as much as possible about the structure of the space you are searching is immensely important, as it allows you to strategically search using the most efficient methods. Additionally, being able to restructure the space can make it more intuitive for creative exploration. Hence the design of any creative system should take the structural design of the creative space very seriously. It is also important to emphasise that the search process is an explorative one. For most creative systems, this search space is *Vast* (McCormack 2008b), and there may be many isolated "Klondike spaces" of rich creative reward. The challenge is to efficiently and effectively find and explore them.

2.1.1 Spaces of Possibility

We should make further distinctions about creative spaces and spaces of possibility. As I have previously discussed (McCormack 2008b), in many domains there are large and crucial differences between the possible and actual. For example, consider a digital image defined by executing an arbitrary Lisp expression over some domain (x, y), where x and y are the co-ordinates of a rectangular grid of pixels that comprise the image. Iterating through each co-ordinate, the expression returns the corresponding pixel's colour. Different expressions will usually generate different images (although many different expressions will also generate the same image). In theory, this system is capable of generating *any* possible image, provided you have the appropriate Lisp expression to generate it.

This represents a space of possibilities than encompasses every possible image that can be represented by coloured pixels over (x, y). For any reasonable image dimensions, the size of this space is Vast, far beyond comparisons with astronomical maximums such as the age of the universe, or the number of basic sub-atomic particles estimated to exist in the universe.

However, the *actual* space of images that can be practically created with a Lisp expression is considerably smaller, limited by physical constraints. From the perspective of evolutionary creativity, if we evolve a Lisp expressions using, for example, an Interactive Genetic Algorithm (IGA, see Sect. 2.2), the actual images produced are all relatively similar and represent an infinitesimally small fraction relative to the possible space of which the system is theoretically capable.[1]

So while a representational system may theoretically cover a large range of possibilities, searching them—even with evolutionary methods—will only permit examination of insignificantly small regions. Furthermore, transformation or modification of the underlying generative mechanism[2] may open up new spaces not so easily found by the original, e.g. the addition of symmetry functions for the Lisp expression example would make it easier to generate images with symmetric elements. Of course we need some way of finding the "right" transformations or modifications to make. This is a kind of "meta-search" (a search of the different types of generative mechanisms that define a representational space). Further, this opens a hierarchy (meta-meta-search, meta-meta-meta-search, etc.), which effectively amounts to the same problem of the possible and actual in our original "flat" search.

What this means in practical terms is that there must be some human-defined generative mechanism as the basis for any computational creative system,[3] which will require serious human ingenuity and creativity if it's design is to be effective. I will return to this point in Sect. 2.4.3. While much research effort and discussion has focused on evaluation and judgement in computational creative systems, representation has received far less attention.

A somewhat analogous situation exists in biology. The space of possible DNA sequences is far greater than the space of viable, or possible, phenotypes.[4] The space of possible phenotypes (those which could exist) is again larger than the space of

---

[1]By my estimates, about 5×10^{-1444925} % for images of modest dimensions, far beyond astronomically small.

[2]By "generative mechanism" I am technically referring to the genotype and the mechanism that expresses it into a phenotype.

[3]The mechanism can include the ability to self-modify, change, or learn.

[4]We might think of "viable" as meaning being able to effectively express a living organism from a zygote or through mitosis of a parent cell. But this is problematic for many reasons, most of which are too tangential to the argument to list here.

actual phenotypes (those which have existed, or currently exist). In nature, what can be successfully expressed by DNA is limited materially by physical constraints and processes. In contrast to our Lisp expression example, once RNA and DNA were established evolution has not really experimented with different self-replication mechanisms. We think of DNA as being a highly successful self-replicating molecule, which might be true, but we have little to compare it with. Many factors affect the *variety* of life that has evolved on Earth. As evolution involves successful adaptations, the changing environment of the Earth is an important factor in determining evolutionary variety. In addition to geological events, environments change due to presence of species and their interactions, a point that I will return to later in this chapter.

2.2 Evolutionary Computing and Creativity

As noted in the previous section, EC methods (which include techniques such as Genetic Algorithms, Evolutionary Strategies and Genetic Programming) have demonstrated success in assisting users of complex creative systems to better locate regions of high creative reward (Bentley and Corne 2002, Romero et al. 2008). In broad terms they are "generate and test" algorithms that evolve a population of candidate solutions or artefacts. New, child artefacts are generated through random mutation and/or recombination with selected parents. Populations are tested or ranked by some measure, with the most highly valued individuals and their offspring more likely to survive in subsequent generations. Incrementally, the overall "quality" of the population *should* improve according to the fitness measure used. How well the method does depends on many factors, including the nature of the fitness landscape (determined in part by the representational scheme) and the evaluation of solution fitness in artefacts. Success or otherwise is dependent on (i) the structure of the phenotype space, and (ii) the effectiveness of the fitness evaluation in determining the quality of the artefacts produced.[5]

Evolutionary approaches and aesthetic evaluation are reviewed extensively in the chapter by Galanter (Chap. 10). So it is pertinent here to make just a few points. Firstly, it is important to differentiate between an evolutionary system that gives *creative* results and one that generates *aesthetically pleasing* results. The former does not preclude the latter, but they are in general, independent (i.e. it is possible for a machine or algorithm to generate aesthetically pleasing images without that system being creative). This distinction is often overlooked.

Some evolutionary systems use learnt or predefined measures of "creative" features in their generated artefacts (Baluja et al. 1994, Machado and Cardoso 2002), or rely on some form of aesthetic measure to evaluate an individual's fitness (Birkhoff 1933, Staudek 2002, Ramachandran 2003, Svangåard and Nordin 2004, Machado et al. 2008). Others use iterative human selection to rank individuals as part of the

[5]This issue is a topic of discussion in Chap. 4.

evolutionary process (Takagi (2001) provides a comprehensive survey). These approaches suffer from difficulties, however. Pre-defined measures of aesthetic properties, for example, risk implicit judgements as to which specific properties are of value (thus determining *what* will be measured). While a number of researchers describe "aesthetic universals" of evolutionary origin (Brown 1991, Dissanayake 1995, Martindale 1999, Ramachandran and Hirstein 1999, Dutton 2002), it is long proposed that aesthetic values also shift according to individual taste, time and culture. Moreover, aesthetics has many interpretations (Koren 2010), and in contemporary art surface aesthetic qualities are often downplayed or given little significance in appreciating the creativity of the work. Evolving artefacts exclusively for aesthetic value does not necessarily make them creative.

Some attempts have been made to expressly minimise or remove the aesthetic judgement of a particular individual. This is what is referred to as removing "the signature" of the artist (Boden 2010, Chaps. 9 & 10). The *Drawbots* system described by Bird et al. (2008) attempted to create a line-drawing robot using evolutionary robotics. Researchers defined "implicit" fitness measures that did not restrict the type of marks the robot drawer should make, including an "ecological model" involving interaction between environment resource acquisition and expenditure through drawing. However, the results demonstrated only minimal creativity, and the authors concluded that fitness functions which embodied "artistic knowledge about 'aesthetically pleasing' line patterns" would be necessary if the robot were to make drawings worthy of exhibition to humans.

Using human selection (known as the *Interactive Genetic Algorithm,* IGA) suffers from a "fitness evaluation bottleneck" that reduces the human operator role to that of a "pigeon breeder" who quickly fatigues (Takagi 2001, Dorin 2001). IGAs are generally more suited to explorations by a non-expert user, who is unfamiliar with the generative mechanism being evolved. Here the IGA allows limited navigation through a space of possibilities without necessarily understanding the underlying mechanisms that generate them.[6]

These standard evolutionary approaches, while historically important and capable of significant results, are not able to consistently generate convincingly creative results in many domains. Can we do better? Biology seemingly can. A useful insight is in recognising that finding the creative "Klondike spaces" is not simply an optimisation problem (i.e. finding a global optima using some fitness criteria). Indeed, for most creative domains the idea of evolving towards a single optimum is counterintuitive, as an artist or designer normally produces many new artefacts over their professional lifetime. New designs or techniques often "evolve" from previous ones, offspring of both the originating artist and his or her peers (Basalla 1998). As Basalla (1998) and others have pointed out using the example of technological evolution, the Western emphasis on individual creativity (reinforced socially through patents and other awards) obscures the important roles played in the evolutionary

[6]Although there are exceptions where the IGA has proved useful to expert users as well, e.g. Dahlstedt (2006), McCormack (2008a).

ecosystem of interactions between environment and prior work of many individuals. Hence:

> The trajectory through a creative space is not one of incrementally optimising towards a single goal or fitness measure, rather it is a complex pathway through a series of intermediate and changing goals, each of which may determine the pathway of the next, and may be creative in its own right.

If we are interested in discovering new creative spaces through the synergetic combination of human intelligence and intuitive structuring and representation of the conceptual space, then there are other possibilities. The evolution of species on earth involves a complex set of interrelated processes and events. For example, species do not exist in isolation from their environment or from other species; together they form a complex network of interdependencies that may impact on the evolutionary process significantly. Let us see what happens if we re-conceptualise the search of a creative space using insights from the structure and function of evolutionary biological ecosystems.

2.3 Ecosystems

Ecosystems are a popular yet somewhat nebulous concept increasingly adopted in contemporary culture. Environmental groups want to preserve them, businesses want to successfully strategise and exploit them, and the media is part of them. With recent sales of Nokia mobile smartphones on the decline, Nokia CEO Stephen Elop bemoaned that fact that his company, unlike its rivals, had failed to create an "ecosystem": one that encompassed smartphones, the operating system, services and users (Shapshak 2011). Media theorists speak of "media ecologies"—the "dynamic interrelation of processes and objects, beings and things, patterns and matter" (Fuller 2005). Philosopher Manuel De Landa emphasises the flows of energy and nutrients through ecosystems manifesting themselves as animals and plants, stating that bodies are "nothing but temporary coagulations in these flows: we capture in our bodies a certain portion of the flow at birth, then release it again when we die and micro-organisms transform us into a new batch of raw materials" (De Landa 2000).

In the broadest terms, the modern concept of an ecosystem suggests a community of connected, but disparate components interacting within an environment. This interaction involves dependency relationships leading to feedback loops of causality. The ecosystem has the ability to self-organise, to dynamically change and adapt in the face of perturbation. It has redundancy and the ability to self-repair. Its mechanisms evoke symbiosis, mutualism and co-dependency, in contrast to pop-cultural interpretations of evolution as exclusively a battle amongst individuals for fitness supremacy. Yet we also speak of "fragile ecosystems", implying a delicate balance or harmony between elements that can easily be broken by external interference. Any anthropomorphic projection of harmony or stability to ecosystems is naïve however. The history of evolution is the history of change: species, their diversity, morphology and physical distribution, the chemical composition of the biosphere, the geography of the earth—all have changed significantly over evolutionary time. The ecosystem's stability is seemingly transitory then, tied to the shifts in species distribution and environment.

2.3.1 Biological Ecosystems

Of course, ecosystems and Ecology are the domain of Biology, where we find a formal understanding, along with many inspirational ideas on the functional relationships found in real biological ecosystems. Modern Ecology is the study of species and their relations to each other and their environment. The term "Ecology" originated with the German Biologist and Naturalist, Ernst Haeckel,[7] who, in 1866, defined it as the "science of the relationship of the organism to the environment", signifying the importance of different species embedded in specific environments. The term "Ecosystem", from the Greek ($o\iota\kappa o\varsigma$, household; $\lambda o\gamma o\varsigma$, knowledge) is attributed to the British Ecologist, Sir Arthur Tansley, who coined it from fellow Botanist Arthur Clapham. It grew out of debates at the time about the similarity of interdependent communities of species to "complex organisms". Importantly, Tansley's use of the term ecosystem encompassed "the inorganic as well as the living components" (Tansley 1939), recognising that the organism cannot be separated from the environment of the biome, and that ecosystems form "basic units of nature" (Willis 1997).

Contemporary definitions of ecosystems begin with the work of American Ecologists Eugene and Howard Odum. Eugene wrote the first detailed Ecology text, *Fundamentals of Ecology*, published in 1953. Odum recognised energy flows, trophic levels,[8] functional, and causal relationships that comprised the ecosystem. Willis defines the modern concept of an ecosystem as "a unit comprising a community (or communities) of organisms and their physical and chemical environment, at any scale, desirably specified, in which there are continuous fluxes of matter and energy in an interactive open system" (Willis 1997).

In more modern terms, Scheiner and Willig (2008) nominate seven fundamental principles of ecosystems:

1. Organisms are distributed in space and time in a heterogeneous manner (inclusionary rule).

---

[7]Danish biologist Eugen Warming is also attributed as the founder of the science of Ecology.

[8]*Autotrophs*, such as plants, produce organic substances from simpler inorganic substances, such as carbon dioxide; *heterotrophs* unable to perform such conversions, require organic substances as a source of energy.

2. Organisms interact with their abiotic and biotic environments (inclusionary rule).
3. The distributions of organisms and their interactions depend on contingencies (exclusionary rule).
4. Environmental conditions are heterogeneous in space and time (causal rule).
5. Resource are finite and heterogeneous in space and time (causal rule).
6. All organisms are mortal (causal rule).
7. The ecological properties of species are the result of evolution (causal rule).

For those wanting to know more details on the contemporary science, a text such as that by Begon et al. (2006) provides a useful overview of Ecology science.

2.3.2 Ecosystem Models in the Creative Arts

A number of different "ecosystemic" approaches exist in the arts. Examination finds that they are quite diverse and only loosely drawn from biological concepts, probably due to multiplicitous and nebulous understandings of Ecology outside Biology, and various metaphoric interpretations of the ecosystem concept.

Design and Architecture. Given the state of human impact on the environment, much theory in landscape and architectural design has sought to bring ideas from Ecology and ecosystems into the design lexicon (see, e.g. Bell 1999). Through a greater understanding of nature's process and function, it is believed that designers can better integrate human interventions within the landscape, minimising their detritus impact, or at least appreciate how design decisions will effect change to the environment over the life of a project, and beyond. In architecture, *Design Ecologies* seeks connections between biological Ecology, human communication, instruction and aesthetics, with an emphasis on "novel concepts of ecologically informed methodologies of communication through design practice" (Murray 2011).

Generative design uses processes adopted from evolution as a source of design variation and customisation. It brings a number of desirable features to the design of artefacts, including a means to generate and manage complexity; self-maintenance and self-repair; design novelty and variation (McCormack et al. 2004). As discussed (Sect. 2.2), evolutionary methods such as the IGA are useful for generative design when the designer has only a rudimentary grasp of the underlying generative mechanism that is being evolved. They permit design changes without the need to understand in detail the configuration or parameter settings that generated the design. The application of generative design to customised manufacture has become feasible in recent years due to the availability of automated, programmable fabrication devices, such as 3D printers, laser cutters, etc. that can inexpensively translate computer representations into one-off physical objects. This allows physical generative designs to be customised to individual constraints or desires on commercial manufacturing scales.

Design associations with Ecology and ecological principles often suggest the superiority of natural over human design, and ecosystems embracing harmony and stable configurations, “in tune” with nature and natural surroundings. Ecological processes provide a certain cachet, appeal and authority that conveniently lend both a design and moral credibility to a project. Such views have been rightly criticised (Kaplinsky 2006). Evolution needs only to offer adequate solutions—ones that are sufficient for growth, survival and reproduction—not necessarily the best or globally optimal ones. “Optimality” for evolution is dependent on environment (obviously polar bears don’t do well in deserts). But it is not that nature has nothing useful to teach us. Moving beyond mimicry, a better understanding of the function and behaviour of real biological ecosystems offers new and rewarding possibilities for design, along with a greater awareness of how our activities ripple out through the environment and affect other species.

Music and Performance. Waters (2007) uses the concept of a “performance ecosystem”—one that encompasses composition, performance, performers, instruments and environment. Here music and music making are seen as part of a multi-layered, complex dynamical system, operating from the acoustic to the social. Emphasis is placed on the dynamical interactions and, importantly, feedback processes between components of the ecosystem. For example, the feedback between a performer and their instrument encompasses the body, tactility, vibrating materials, physical and acoustic properties of the room in which the instrument is played, along with the “psychological adaptations and adjustments” in the body of the performer, who is deeply connected to, and part of these interacting elements.

Such connections evoke the cybernetic: instruments can be considered part of a continuum that originates from the body, extending through instrument and environment. Italian composer, Agostino Di Scipio (2003) seeks a reformulation of what is meant by “interaction” in a technological performance context and invokes the cybernetic concept of ecosystems and feedback dependencies as a sonic interaction paradigm. This is indicative of a more general sense of failure, in creative contexts, of standard technical approaches to human-computer interaction. These traditional approaches emphasise the functional over the explorative and connected. An alternate view, advocated by Di Scipio and many others, sees interaction as “a by-product of lower level interdependencies among system components” (Di Scipio 2003). Components are *adaptive* to their surrounding external conditions and able to *manipulate* them. In the case of sound, this involves a sound ecosystem of sound-generating, sound-listening and sound-modifying components, connected in feedback loops with their acoustic environment. In this configuration sound itself is the medium in which the ecosystem exists. The coupling of components with their environment allows them to change and reconfigure in response to environmental variation: an environment that the components themselves may be modifying.

Visual and Installation Art. My own interactive installation, *Eden* (McCormack 2001), is a complex artificial ecosystem running in real-time on a two-dimensional lattice of cells, projected into a three-dimensional environment (Fig. 2.2). The simulation includes seasonal variation, planetary albedo modified by biomass composition (Lenton and Lovelock 2001), and a simulation of sound propagation and at-

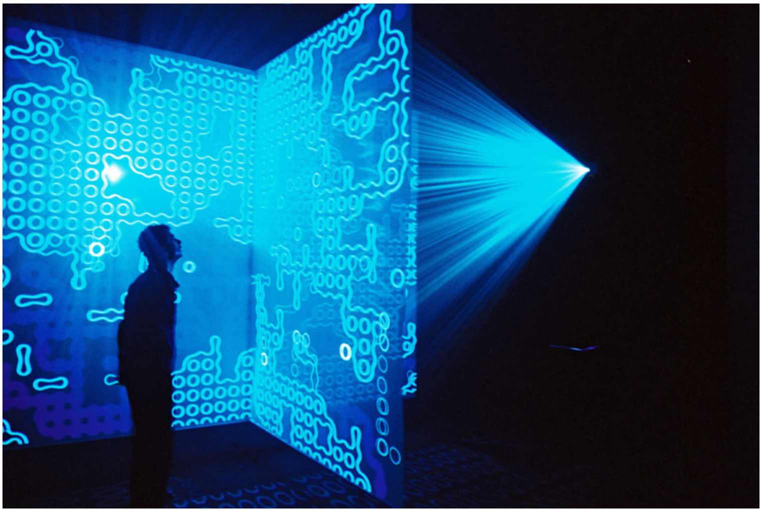


Fig. 2.2 The author's *Eden* installation: an evolving ecosystem of virtual creatures learn new behaviours based on interaction with their environment and with their human audience

tenuation. Evolving, learning agents modify and adapt to their surroundings. Interestingly, the agents learn a number of behaviours not explicitly programmed into the system, including hibernation during winter months when food resources are scarce, predation, and primitive signalling using sound. A computer vision system links human visitor presence to the generation of biomass (food for the agents), and over time agents learn to make interesting sequences of sound in order to keep visitors attracted near the work, thus increasing their supply of food and chances of reproductive success (McCormack 2005).

Over the last twenty years, Dutch artists Erwin Driessens and Maria Verstappen[9] have been experimenting with generative "processes of production" in their art practice. This has extensively encompassed the use of ecosystem metaphors in a number of their works. For example, *E-volver* is a generative visual artwork where a small collection of agents roam a gridded landscape of coloured pixels, choosing to modify the pixel underneath them based on it's colour, and those of the neighbouring pixels. Each agent has a set of rules that determine how to change the colour and where to move next (Driessens and Verstappen 2008). Through the interaction of these pixel-modifying agents and their environment (the pixels which comprise the image), E-volver is able to generate a fascinating myriad of complex and detailed images (Fig. 2.3 shows one example), all of which begin from a uniformly grey canvas. The images, while abstract, remind the viewer of landscape viewed from high altitude, or an alien mould overwhelming a surface, or electron micrographs of some unidentified organic structure. Importantly, they exhibit details on a variety of scales, with coherent structures extending far beyond the one pixel sensory radius of

[9] See their website at: http://www.xs4all.nl/~notnot/index.html.

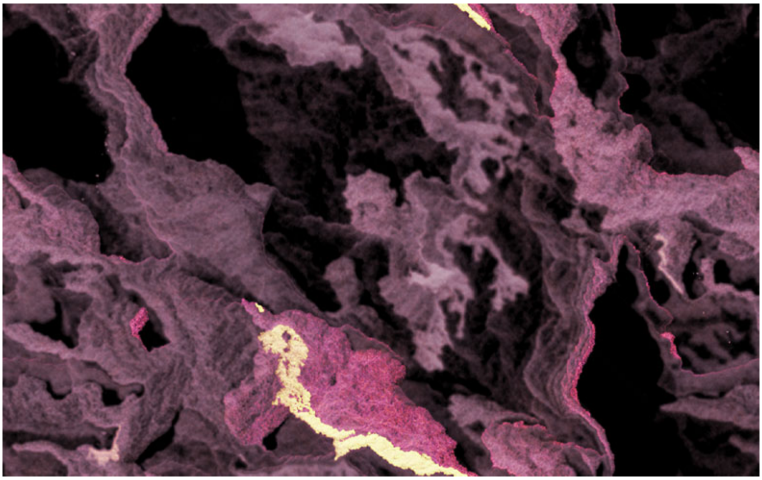


Fig. 2.3 An image produced by Driessens and Verstappen's E-volver. Eight pixel modifying agents build the image by modifying pixels. Notice the image contains coherent structures over multiple levels of detail

the agents that created them. This suggests a collective self-organisation achieved through agent-environment interaction, with the environment acting as a "memory" that assists agents in building coherent structures within the image.

Like Di Scipio's sonic ecosystems, E-volver's "environment" is *the medium itself* (an image comprised of coloured pixels). For Eden, the real and virtual environments are causally connected through sound, human presence and the production of resources. In both E-volver and Eden, agents modify their environment which, in part, determines their behaviour. Causally coupling agent to environment allows for feedback processes to be established, and the system thus becomes self-modifying. This iterative self-modification process facilitates the emergence of heterogeneous order and fractaline complexity from an environment of relative disorder and simplicity. For Eden this is further expanded by the use of an evolutionary learning system (based on a variant of Wilson's XCS (Wilson 1999)) that introduces new learning behaviours into the system. Learnt behaviours that have been beneficial over an agent's lifetime are passed onto their offspring.

Unlike Eden's learning agents, E-volver's agents are not evolutionary over the life of the ecosystem, yet they are evolved: a variation on the IGA allows the user of the system to evolve ecosystem behaviours through aesthetic rejection ("death of the unfittest"). The entire ecosystem (a set of eight agents and their environment) is evolved, not individual agents within a single image. Selection is based on the subjective qualities of the images produced by an individual ecosystem.

There are numerous other examples of successful artworks based on ecosystem metaphors and processes. To return to the central questions of this chapter: *how* and *why* do they work successfully?

Table 2.1 General properties of creative ecosystem models

Property	Features
Components & their environment	Together these constitute the ecosystem
Dynamical system	Enables the ecosystem to temporally adapt and change in response to internal and external conditions
Self-observation	Provides a link between component action and environment
Self-modification	Allows a component to adjust its behaviour within the system
Interaction	Components must interact with each other and their environment to give rise to emergent behaviours of the system as a whole
Feedback loops	Provide pathways of control, regulation and modification of the ecosystem
Evolution	Allows long term change, learning and adaptation

2.4 Ecosystem Design Patterns

Within our research group[10] at the Centre for Electronic Media Art we have investigated ecosystemic processes as a basis for designing or enhancing generative artworks (see e.g. McCormack (2001, 2007b, 2007a), Eldridge et al. (2008), Eldridge and Dorin (2009), Bown and McCormack (2010)). Our long-term aim has been to develop a catalogue of ecosystemic "design patterns" in the spirit of Gamma et al. (1995), which facilitate the building of creative evolutionary systems. Developing these patterns does not imply a "plug-and-play" approach where one just selects the appropriate patterns, connects them together, and then sits back to watch the creativity evolve. Rather, the patterns serve as starting points in conceptualising a specific creative system, documenting intermediate mechanisms and the typical behaviours they produce. Choosing *which* pattern to use and *how* to apply them remains a matter of significant creative judgement.

Di Scipio sees the artistic system as a "gathering of connected components", and it is these components and their interdependencies that must be carefully designed if successful system-level results are to ensue. Components must additionally be adaptive to surrounding external conditions and be able to manipulate them.

Table 2.1 summarises the basic properties we think are important to creative ecosystem models. The key to developing a successful ecosystem model is in the design of the system's components, their meaning, interpretation and interaction. In the following sections, I will explore some of these features in more detail, using completed ecosystem artworks as examples.

2.4.1 Environments: Conditions and Resources

In broad terms, biological environments have two main properties that determine the distribution and abundance of organisms: *conditions* and *resources*. Conditions are

[10] Which has included over the last few years: Oliver Bown, Palle Dahlstedt, Alan Dorin, Alice Eldridge, Taras Kowaliw, Aidan Lane, Gordon Monro, Ben Porter and Mitchell Whitelaw.

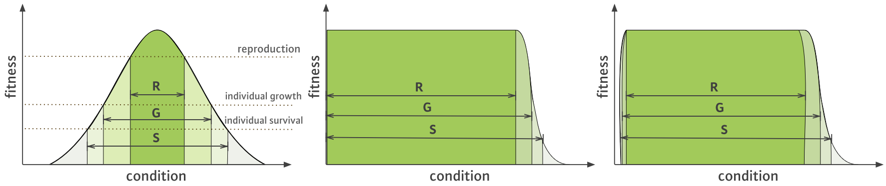


Fig. 2.4 Example organism viability curves for reproduction, growth and survival, from Begon et al. (2006)

physiochemical features of the environment (e.g. temperature, pH, wind speed). An organism's presence may change the conditions of its local environment (e.g. one species of plant may modify local light levels to that which another species is adapted for). Conditions may vary in cyclic patterns or be subject to the uncertainty of prevailing environmental events. Conditions can also serve as stimuli for other organisms. Resources, on the other hand, are consumed by organisms in the course of their growth and reproduction. One organism may become or produce a resource for another through grazing, predation, parasitism or symbiosis, for example.

For any particular condition or resource, an organism may have a preferred value or set of values that favour its survival, growth and reproduction. Begon et al. (2006) define three characteristic curves, which show different "viability zones" for survival, growth and reproduction (Fig. 2.4).

In developing artworks, we can abstract these concepts significantly as long as we are clear about the functional relationships between conditions, resources and organism. From here on we will consider the organism as a "component" of an ecosystem, this more genetic term useful to remind us of the abstractions in play. Components may often be called "agents" in a computer simulation, typically representing autonomous entities with parameterised, possibly evolving, behaviours.

2.4.2 Self-observation and Feedback

Self-observation gives rise a type of feedback process, similar to a governor or more simply "rein control" (Harvey 2004). Here "observation" means the system monitoring of environmental conditions or resources that are necessary for reproduction, growth and survival and shifting its configuration in response. A component is causally coupled to the environment through relevant conditions or resources within its environment. Observation may be implicit or explicit, local or global. Observation forms a critical connection between a component's effect on the environment and its ability to modify its behaviour in response, typically to retain homeostasis in local conditions or resources. The use of the term "observation" is deliberately a loaded one. It is used in the cybernetic sense and does not imply a necessary concept of agency (although it does not preclude it). It might be considered the most simple precursor to more complex observational intelligence. It also suggests a system-level (as opposed to an individual-level) ontology that emerges through the interaction of system components.

The well-known model of planetary homeostasis, *Daisyworld*, uses a simple form of system level self-observation (Lenton and Lovelock 2001). Planetary albedo is affected by proportions of black and white daisies, whose relative proportions change according to surface temperature. What is fascinating about Daisyworld is its ability to maintain a homeostatic surface temperature while the incoming radiant heat energy increases.

In the ecosystem artwork *Colourfield* (McCormack 2007a), individual components ("agents") are bands of colour occupying a 1D lattice of cells. Genetic information controls the colour the agent produces, along with its preference to adapt to the colour of its neighbours and its propensity to occupy vacant neighbouring cells (thus making a larger contribution to the overall colour distribution). A feedback mechanism uses a colour histogram of the overall colour distribution to allocate resources to each individual agent on a per-time step basis (Fig. 2.5). Here the observation mechanism—resource allocation based on the image histogram—is implicit and global (the system as a whole is observing itself). An individual agent's contribution to the overall image influences the production of its own resources and those of others. The more cells an individual occupies, the greater the reliance of other individuals to it. Here feedback is an environmental reward function that favours symbiotic adaptations because of its global nature (resources are equally divided between cells). As the system is evolutionary, as a whole it has the ability to modify its colour composition and distribution in response to the "self-observation" provided by this feedback mechanism.

A different self-observation mechanism is in operation in the ecosystem artwork *Niche Constructions* (McCormack 2010). Niche construction is the process by which organisms, through their activities, modify their heritable environment (and potentially the environments of others). Advocates of niche construction theory in biology argue that it is an initiator of evolutionary change, rather than simply an evolutionary outcome (Odling-Smee et al. 2003). The complete set of conditions and resources affecting an organism represent its *niche*, which can be conceptualised as a hypervolume in n-dimensional space.

In the Niche Constructions artwork, evolutionary line drawing agents draw on an initially blank canvas as they move around. A set of normalised scalar values forms an agent's genome, which directs its behaviour over its lifetime. Individual alleles control rate of drawing curvature, "irrationality" (Fig. 2.6), fecundity and mortality. Agents die if they intersect with any previously drawn line or run off the page. The canvas is seeded with a small initial population of *founder agents*—initialised with uniformly distributed random genomes and positions—that proceed to move, draw and reproduce. There is no limit to the number of offspring an agent may have, but in general the lifespan of agents decreases as the density of lines becomes greater, because it is increasingly difficult to avoid intersection with existing lines. Eventually the entire population dies out and the image is complete. This finished drawing represents the "fossil record" of all the generations of lines that were able to live over the lifetime of the simulation.

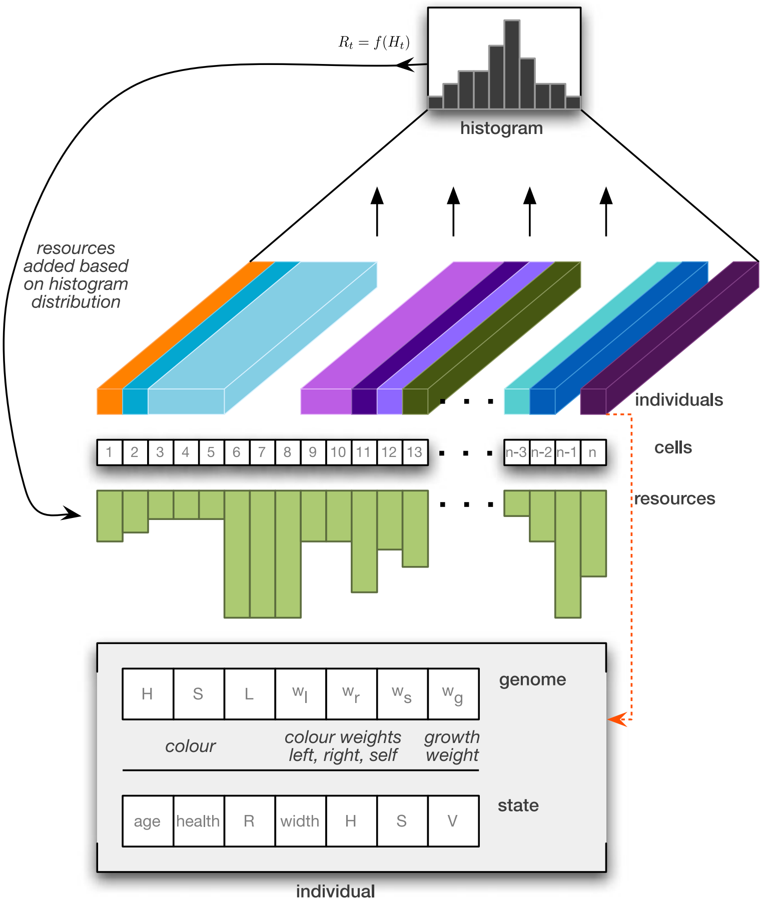


Fig. 2.5 Feedback relationships between component and environment creates a self-observation in the ecosystemic artwork "Colourfield"

Niche construction is enabled in this work through the addition of a self-observation mechanism that genetically links drawing behaviour to local conditions. As an individual agent draws on the canvas, the local density around it is measured. Each agent has an allele that represents its ideal density preference, i.e. the local line density that is most conducive to its survival, growth and reproduction. As the actual density shifts away from this ideal value, the agent finds it harder to reproduce, grow and survive. If the preferred density and actually density differ too greatly, the agent will die (see Fig. 2.7). Of course the actual value of this density preference is subject to evolutionary change and over the life of the drawing, average density preference increases in the population (McCormack 2010). The niche construction process influences agent behaviour: low density liking agents try and draw large, closed spaces to prevent other lines from decreasing their local density. High density seeking lines

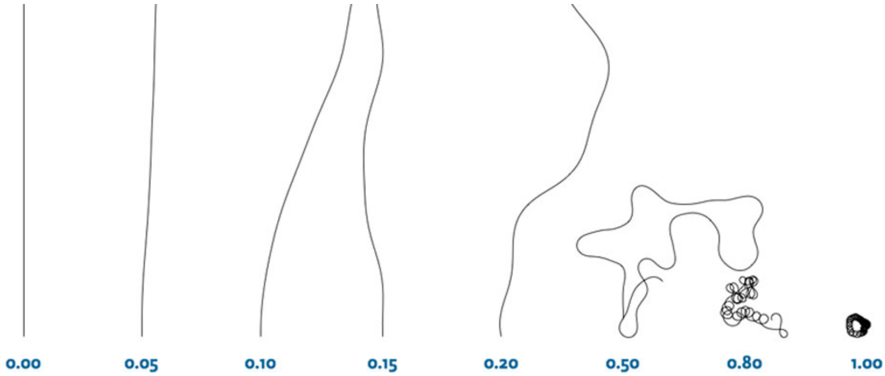


Fig. 2.6 Individual line drawing agents with different genetic values of irrationality. Note that the "die if intersect" rule has been turned off for these examples

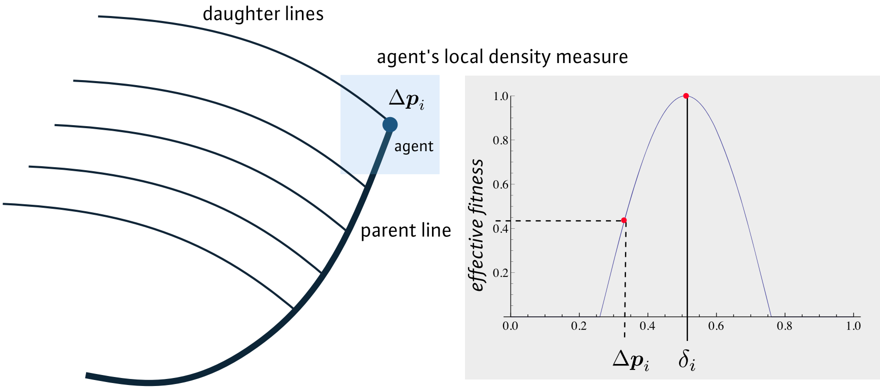


Fig. 2.7 The niche construction mechanism for drawing agents: a local line density measure, Δp_i, facilitates a self-observation mechanism. The agent's genome includes an allele that represents a preferred density (δ_i). The difference between preferred density and measured density affects the agent's effective fitness, hence its ability to survive, grow, and reproduce

give birth to large numbers of offspring, who quickly fill the canvas with lines of close proximity. Some examples are shown in Fig. 2.8.

This local, implicit self-observation plays a vital role in influencing the overall density variation and aesthetics of the images produced. We know this because turning the mechanism off produces images of significantly less density variation (statistically) and visual interest (subjectively).

2.4.3 Automation and the Creative Role of the Artist

automation (noun): the use of largely automatic equipment in a system of manufacturing or other production process

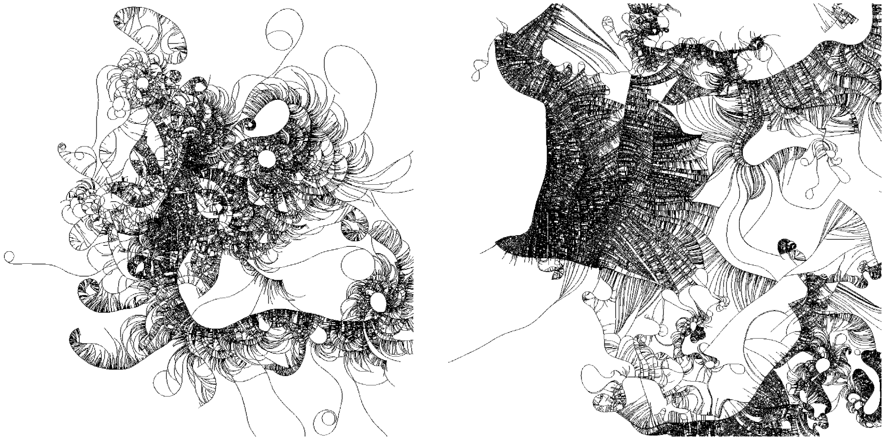


Fig. 2.8 Two sample outputs from the line drawing system with niche construction

The term "automation" originated in the USA, from the newly industrialised engineering of the 1940s, although similar concepts arose prior in different guises, both historically and geographically. The central idea was to create machines to perform tasks previously performed by humans. The rational was largely economic: machines that could replace and even out-perform their human counterparts will increase production efficiency. As a central driving force in US industrialisation and technologisation throughout the twentieth century, computers enabled the increasing sophistication and range of capabilities for automation within the capitalist economic system. The idea of machines automating human tasks still underpins many technology-driven approaches to "automating creativity". Traditional AI or EC approaches seek the automation of aesthetic or creative optima finding. In contrast, the ecosystemic approach, as outlined here, does not seek to automate the human out of the creative process, nor claim to equal or better human creative evaluation and judgement. It views creative search and discovery as an *explorative* process, as opposed to an optimisation.

Ecosystemic processes recognise the importance of the link between structure and behaviour. Ecosystem components must be embedded in, and be part of, the medium in which they operate. The design of the system—components and their interdependencies—requires skill and creativity. This design forms the conceptual and aesthetic basis by which the outcomes can be understood. So rather than removing the artist by automating his or her role, the artist's contribution is one of utmost creativity—creativity that is enhanced through interaction with the machine. As is also argued elsewhere in this book, forming an "ecosystem" that encompasses humans, technology and the socially/technologically mediated environment, opens up further ecosystemic possibilities for creative discovery.

There are of course, many reasons why we might seek some form of "automated creativity" or aesthetic judgement,[11] apart from replacing human labour. For example, automated creativity could lead to creative discovery that exceeds any human capability, or provides greater insights on the mechanisms of human creativity by attempting to model it. But these are "blue sky" speculations, and current technological advances in this area can just as easily homogenise and suffocate the creative decision-making process for human users, as they can expand or enhance it. A good example can be seen in recent digital camera technologies. Over the last ten years, as computational power has escalated, digital cameras have increasingly shifted creative decision making to the camera instead of the person taking the picture. We see modes with labels like "Intelligent Auto" or scene selection for particular scenarios ("Fireworks","Landscape","Sunset", "Beach"). These modes supposedly optimise many different parameters to achieve the "best" shot—all the photographer has to do is frame the image and press the button.[12] Recent advances even take over these decisions, choosing framing by high-level scene analysis and deciding when the picture should be taken based on smile detection, for example. Such functionality trends towards the removal of much human creative decision-making, subjugating the human photographer to an increasingly passive role.

As anyone who has used a entirely manual camera knows, hand-operated "slow technology" forces the user to think about all aspects of the photographic process and their implications for the final image. The user's role is highly active: experimentation, mistakes, and serendipitous events are all possible, even encouraged—well known stimuli for creativity. If the *design* of components and their interaction is good, then using such a device isn't marred by complexity or limited by inadequate functionality, which is often the rationalisation given in automation of creative functionality.

Shifting the thinking about the design of technology from one of "complexity automation" (where complexity is masked through "intelligent" simplicity) to one of "emergent complexity" (where interaction of well designed components generates new, higher-level functionality) allows the human user to potentially expand their creativity rather than have it subsumed and homogenised.

2.5 Conclusions

Ecosystemics represents an alternative, biologically-inspired approach to creative discovery over more traditional methods such as genetic algorithms or genetic programming. It offers an interesting conceptual basis for developing new creative systems and processes, even in non-computational settings. Incorporating an "environment", and allowing interactions between dynamic components and that environment, permits a rich complexity of creative possibilities for the artist wishing to

[11] Chapter 4 discusses this issue in more detail.

[12] Reminiscent of Kodak founder George Eastman's famous tag line of 1888 for the Kodak No. 1 camera: "You press the button, we do the rest".

exploit the generative nature of ecosystem processes. While ecosystemic methods don't offer a "magic bullet" in terms of searching the creative Klondike spaces of any generative system, they do make it easier to at least begin to conceptualise and design systems capable of high creative reward. As the complexity and sophistication of ecosystem artworks develop, we are likely to see further advances in the new creatively made possible with computers that use this approach.

Acknowledgements This research was supported by Australian Research Council Discovery Grants DP0877320 and DP1094064.

References

Aunger, R. (2002). *The electric meme: a new theory of how we think*. New York: Free Press.

Baluja, S., Pomerleau, D., & Jochem, T. (1994). Simulating user's preferences: towards automated artificial evolution for computer generated images. *Connection Science*, *6*, 325–354.

Basalla, G. (1998). *The evolution of technology*. Cambridge: Cambridge University Press.

Begon, M., Townsend, C., & Harper, J. (2006). *Ecology: from individuals to ecosystems*. New York: Wiley-Blackwell.

Bell, S. (1999). *Landscape: pattern, perception and process*. London: E & F N Spon.

Bentley, P. J., & Corne, D. W. (Eds.) (2002). *Creative evolutionary systems*. London: Academic Press.

Bird, J., Husbands, P., Perris, M., Bigge, B., & Brown, P. (2008). Implicit fitness functions for evolving a drawing robot. In M. Giacobini et al. (Eds.), *Lecture notes in computer science: Vol. 4974. Applications of evolutionary computing, EvoWorkshops 2008: EvoCOMNET, EvoFIN, EvoHOT, EvoIASP, EvoMUSART, EvoNUM, EvoSTOC, and EvoTransLog, Proceedings*, Naples, Italy, March 26–28, 2008 (pp. 473–478). Berlin: Springer.

Birkhoff, G. D. (1933). *Aesthetic measure*. Cambridge: Harvard University Press.

Boden, M. A. (2010). *Creativity and art: three roads to surprise*. London: Oxford University Press.

Bown, O., & McCormack, J. (2010). Taming nature: tapping the creative potential of ecosystem models in the arts. *Digital Creativity*, *21*(4), 215–231. http://www.csse.monash.edu.au/~jonmc/resources/DC2010/.

Brown, D. E. (1991). *Human universals*. New York: McGraw-Hill.

Dahlstedt, P. (2006). A mutasynth in parameter space: interactive composition through evolution. *Organised Sound*, *6*(2), 121–124.

Dawkins, R. (1999). *The extended phenotype: the long reach of the gene* (rev. ed.). Oxford: Oxford University Press.

De Landa, M. (2000). *A thousand years of nonlinear history*. Cambridge: MIT Press.

Di Scipio, A. (2003). 'Sound is the interface': from interactive to ecosystemic signal processing. *Organised Sound*, *8*(3), 269–277.

Dissanayake, E. (1995). *Homo aestheticus: where art comes from and why*. Seattle: University of Washington Press.

Dorin, A. (2001). Aesthetic fitness and artificial evolution for the selection of imagery from the mythical infinite library. In J. Kelemen & P. Sosík (Eds.), *LNAI: Vol. 2159. Advances in artificial life* (pp. 659–668). Prague: Springer. http://www.csse.monash.edu.au/~aland/PAPERS/aestheticFitness_ECAL2001.pdf.

Driessens, E., & Verstappen, M. (2008). Natural processes and artificial procedures. In P. F. Hingston, L. C. Barone & Z. Michalewicz (Eds.), *Natural computing series. Design by evolution: advances in evolutionary design* (pp. 101–120). Berlin: Springer.

Dutton, D. (2002). Aesthetic universals. In B. Gaut & D. M. Lopes (Eds.), *The Routledge companion to aesthetics*. London: Routledge. http://www.denisdutton.com/universals.htm.

Eiben, A. E., & Smith, J. E. (2003). *Introduction to evolutionary computing*. *Natural computing series*. Berlin: Springer.
Eldridge, A. C., & Dorin, A. (2009). Filterscape: energy recycling in a creative ecosystem. In M. Giacobini et al. (Eds.), *Lecture notes in computer science: Vol. 5484*. *Applications of evolutionary computing, EvoWorkshops 2009: EvoCOMNET, EvoENVIRONMENT, EvoFIN, EvoGAMES, EvoHOT, EvoIASP, EvoINTERACTION, EvoMUSART, EvoNUM, EvoSTOC, EvoTRANSLOG, Proceedings*, Tübingen, Germany, April 15–17, 2009 (pp. 508–517). Berlin: Springer.
Eldridge, A. C., Dorin, A., & McCormack, J. (2008). Manipulating artificial ecosystems. In M. Giacobini et al. (Eds.), *Lecture notes in computer science: Vol. 4974*. *Applications of evolutionary computing,EvoWorkshops 2008: EvoCOMNET, EvoFIN, EvoHOT, EvoIASP, EvoMUSART, EvoNUM, EvoSTOC, and EvoTransLog, Proceedings*, Naples, Italy, March 26–28, 2008 (pp. 392–401). Berlin: Springer.
Fuller, M. (2005). *Media ecologies: materialist energies in art and technoculture*. Cambridge: MIT Press.
Gamma, E., Helm, R., Johnson, R., & Vlissides, J. M. (1995). *Design patterns: elements of reusable object-oriented software*. *Addison-Wesley professional computing series*. Reading: Addison-Wesley.
Harvey, I. (2004). Homeostasis and rein control: from daisyworld to active perception. In J. B. Pollack, M. A. Bedau, P. Husbands, T. Ikegami & R. A. Watson (Eds.), *Ninth international conference on artificial life* (pp. 309–314). Cambridge: MIT Press.
Kaplinsky, J. (2006). Biomimicry versus humanism. *Architectural Design*, *76*(1), 66–71.
Keane, A. J., & Brown, S. M. (1996). The design of a satellite boom with enhanced vibration performance using genetic algorithm techniques. In I. C. Parmee (Ed.), *Conference on adaptive computing in engineering design and control 96, P.E.D.C.* (pp. 107–113).
Koren, L. (2010). *Which "Aesthetics" do you mean?: ten definitions*. Imperfect Publishing.
Lenton, T. M., & Lovelock, J. E. (2001). Daisyworld revisited: quantifying biological effects on planetary self-regulation. *Tellus*, *53B*(3), 288–305.
Luke, S. (2009). *Essentials of metaheuristics*. Lulu Publishing, Department of Computer Science, George Mason University
Lumsden, C. J. (1999). Evolving creative minds: stories and mechanisms. In R. J. Sternberg (Ed.), *Handbook of creativity* (pp. 153–169). Cambridge: Cambridge University Press. Chap. 8.
Machado, P., & Cardoso, A. (2002). All the truth about NEvAr. *Applied Intelligence*, *16*(2), 101–118.
Machado, P., Romero, J., & Manaris, B. (2008). Experiments in computational aesthetics. In J. Romero & P. Machado (Eds.), *The art of artificial evolution: a handbook on evolutionary art and music* (pp. 381–415). Berlin: Springer.
Martindale, C. (1999). Biological bases of creativity. In R. J. Sternberg (Ed.), *Handbook of creativity* (pp. 137–152). Cambridge: Cambridge University Press. Chap. 7.
McCormack, J. (2001). Eden: an evolutionary sonic ecosystem. In *Lecture notes in computer science: Vol. 2159*. *Advances in artificial life, proceedings of the sixth European conference, ECAL* (pp. 133–142).
McCormack, J. (2005). On the evolution of sonic ecosystems. In *Artificial life models in software* (pp. 211–230). London: Springer. http://www.springeronline.com/sgw/cda/frontpage/0,11855,5-40007-22-39144451-0,00.html.
McCormack, J. (2007a). Artificial ecosystems for creative discovery. In *Proceedings of the 9th annual conference on genetic and evolutionary computation (GECCO 2007)* (pp. 301–307). New York: ACM.
McCormack, J. (2007b). Creative ecosystems. In A. Cardoso & G. Wiggins (Eds.), *Proceedings of the 4th international joint workshop on computational creativity* (pp. 129–136).
McCormack, J. (2008a). Evolutionary L-systems. In P. F. Hingston, L. C. Barone & Z. Michalewicz (Eds.), *Natural computing series*. *Design by evolution: advances in evolutionary design* (pp. 168–196). Berlin: Springer.

McCormack, J. (2008b). Facing the future: evolutionary possibilities for human-machine creativity. In P. Machado & J. Romero (Eds.), *The art of artificial evolution: a handbook on evolutionary art and music* (pp. 417–451). Berlin: Springer.
McCormack, J. (2010). Enhancing creativity with niche construction. In H. Fellerman et al. (Eds.), *Artificial life XII* (pp. 525–532). Cambridge: MIT Press.
McCormack, J., Dorin, A., & Innocent, T. (2004). Generative design: a paradigm for design research. In J. Redmond, D. Durling, & A. de Bono (Eds.), *Futureground, vol. 1: abstracts, 2: proceedings* (p. 156). Melbourne: Design Research Society.
Murray, S. (2011). Design ecologies: editorial. *Design Ecologies*, *1*(1), 7–9.
Odling-Smee, F. J., Laland, K. N., & Feldman, M. W. (2003). *Niche construction: the neglected process in evolution*. *Monographs in population biology*. Princeton: Princeton University Press.
Perkins, D. N. (1996). Creativity: beyond the Darwinian paradigm. In M. Boden (Ed.), *Dimensions of creativity* (pp. 119–142). Cambridge: MIT Press. Chap. 5.
Ramachandran, V. S. (2003). The emerging mind. In *Reith lectures; 2003*. London: BBC in association with profile Books.
Ramachandran, V. S., & Hirstein, W. (1999). The science of art: a neurological theory of aesthetic experience. *Journal of Consciousness Studies*, *6*, 15–51.
Romero, J., & Machado, P. (Eds.) (2008). *The art of artificial evolution: a handbook on evolutionary art and music*. *Natural computing series*. Berlin: Springer.
Scheiner, S. M., & Willig, M. R. (2008). A general theory of ecology, *Theoretical Ecology 1*, 21–28.
Shapshak, T. (2011). Why Nokia got into bed with Microsoft. http://www.bizcommunity.com/Article/410/78/57030.html.
Staudek, T. (2002). *Exact aesthetics. Object and scene to message*. PhD thesis, Faculty of Informatics, Masaryk University of Brno.
Svangåard, N., & Nordin, P. (2004). Automated aesthetic selection of evolutionary art by distance based classification of genomes and phenomes using the universal similarity metric. In G. R. Raidl, S. Cagnoni, J. Branke, D. Corne, R. Drechsler, Y. Jin, C. G. Johnson, P. Machado, E. Marchiori, F. Rothlauf, G. D. Smith & G. Squillero (Eds.), *Lecture notes in computer science: Vol. 3005*. *EvoWorkshops 2004* (pp. 447–456). Berlin: Springer.
Takagi, H. (2001). Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation. *Proceedings of the IEEE*, *89*, 1275–1296.
Tansley, A. G. (1939). British ecology during the past quarter-century: the plant community and the ecosystem. *Journal of Ecology*, *27*(2), 513–530.
Waters, S. (2007). Performance ecosystems: ecological approaches to musical interaction. In *EMS07—the 'languages' of electroacoustic music*, Leicester.
Willis, A. J. (1997). The ecosystem: an evolving concept viewed historically. *Functional Ecology*, *11*(2), 268–271.
Wilson, S. W. (1999). *State of XCS classifier system research* (Technical report). Concord, MA.

Chapter 3
Construction and Intuition: Creativity in Early Computer Art

Frieder Nake

Abstract This chapter takes some facets from the early history of computer art (or what would be better called "algorithmic art"), as the background for a discussion of the question: how does the invention and use of algorithms influence creativity? Marcel Duchamp's position is positively referred to, according to which the spectator and society play an important role in the creative process. If creativity is the process of surmounting the resistance of some material, it is the algorithm that takes on the role of the material in algorithmic art. Thus, creativity has become relative to semiotic situations and processes more than to material situations and processes. A small selection of works from the history of algorithmic art are used for case studies.

3.1 Introduction

In the year 1998, the grand old man of German pedagogy, Hartmut von Hentig, published a short essay on creativity. In less than seventy pages he discusses, as the subtitle of his book announces, "high expectations of a weak concept" (Hentig 1998). He calls the concept of creativity "weak". This could mean that it is not leading far, it does not possess much expressive power, nor is it capable of drawing a clear line. On the other hand, many may believe that creativity is a strong and important concept.

Von Hentig's treatise starts from the observation that epochs and cultures may be characterised by great and powerful words. In their time, they became the call to arms, the promise and aspiration that people would fight for. In ancient Greece, Hentig suggests, those promises carried names like *arete* (excellence, living up to one's full potential), and *agon* (challenge in contest). In Rome this was *fides* (trust) and *pietas* (devotion to duty), and in modern times this role went to *humanitas*, enlightenment, progress, and performance. Hardly ever did an epoch truly live up to what its great aspirations called for. But people's activities and decisions, if only ideologically, gained orientation from the bright light of the epoch's promise.

F. Nake (✉)
University of Bremen, Bremen, Germany
e-mail: nake@informatik.uni-bremen.de

J. McCormack, M. d'Inverno (eds.), *Computers and Creativity*,
DOI 10.1007/978-3-642-31727-9_3,

If in current times we were in need of a single such concept, "creativity" would probably be considered as one of the favourites. Information, communication, sustainability, ecology, or globalisation might be competing. However, creativity would probably still win. It is a concept full of shining promise. Nobody dares criticise it as plastic and arbitrary. Everybody appears to be relating positively to it. Techno-freaks use it as well as environmentalists. No political party would drop it from their rhetoric.

Creativity may be considered as a *means* for activity, or as its *goal*. However, von Hentig is sceptical about the possibility of developing *more* creativity through education and training; he is also sceptical about creative skills independent of the context. Creativity as an abstract, general concept, taken out of context, is unlikely to exist. If a helpful concept at all, creativity is bound to situations and contexts. Only relative to them may our judgement evaluate an activity as creative. Creativity exists only concretely.

Leaving out ancient Greece, the Middle Ages, and the Renaissance, it seems that the way we understand "creativity" *today* is as a US-American invention (Hentig 1998, p. 12). It started with the fabulous definition of an IQ (Intelligence Quotient) and operational tests to measure it by Stern (1912) in Germany. His approach became an operational method in the USA by the end of World War I. J.P. Guilford (1950) and others made clear that IQ tests did not identify anything that might be called "creative". Current creativity research starts from this article. Like any other measure, a test of your IQ may at best say something about a standard behaviour within given boundaries, but not much about crossing boundaries. Often people do what they are *supposed* to do, and they do it well. Others do what they *want* to do, and do it to the dismay of their bosses, teachers, or parents.

When we consider creativity as an attribute, a property, or a feature that we may acquire by taking courses or joining training camps, we put creativity close to a thing, or a commodity. We inadvertently transform a subjective activity or behaviour into an objective thing. We may acquire many or few commodities, cheap or expensive ones. But is quantity important for understanding creativity, or for becoming a creative person? Doesn't it make more sense to associate the term "creativity" with behaviour, activity, situation, and context? The idea of attaching creativity to individuals is probably what we are immediately inclined to think. But it may still not be very helpful. Creativity seems to emerge in situations that involve several people, who interact in different roles with favourable and unfavourable conditions and events.[1]

We may align intelligence with making sense in a situation that makes sense. If we do so, creativity could be viewed as making sense in situations of nonsense. Dream and fantasy are, perhaps, more substantial to creative behaviour than anything else.

[1] We are so much accustomed to thinking of creativity as an individual's very special condition and achievement that we react against a more communal and cooperative concept. It would, of course, be foolish to assume individuals were not capable of creative acts. It would likewise be foolish to assume they can do so without the work of others.

With these introductory remarks I want to announce a sceptical distance to the very concept of creativity. With only little doubt, a phenomenon seems to exist that we find convenient to call by this name. A person engaged in a task that requires a lot of work, imagination, endurance, meetings, walks, days and nights, music, or only a flash in the mind, will use whatever means she can get hold of in pursuit of her task. Even computers and the Internet may be helpful, and they, indeed, often are. If the final result of such efforts is stamped as a "creative" product, is it then sensible to ask the question: what software and other technical means contributed to this creation? Not much, in my view. And certainly nothing that goes beyond their instrumental character. More interesting is to study changes in the role of the instrument as an instrument. The sorcerer's broom is more than a broom only in the eyes of the un-initiated. It is an expression of a human's weakness, not of the instrument's clever strength.

Therefore, I find it hard to seriously discuss issues of the kind: how to enhance creativity by computer? Or: how do our tools become creative? If anything is sure about creativity, it is its nature as a quality. You cannot come by creativity in a quantitative way, unless you reduce the concept to something trivial.

In this chapter, I will study a few examples of early computer art. The question is: How did the use of computers influence creative work in the visual arts? The very size and complexity of the computer, the division into hardware and software must, at the time, have had a strong influence on artistic creativity. The approach will be descriptive and discursive. I will not explain. Insight is with the reader and her imagination, not with the black printed material. I will simply write and describe. I cannot do much more.

The chapter is divided into four narrations. All four circle around processes of art or, in a less loaded expression, around aesthetic objects and processes. The art we will study here is, not surprisingly, algorithmically founded. It is done, as might be said, by *algorists*.[2] They are artists of a new kind: they *think* their works and let machines carry them out. These artists live between aesthetics and algorithmics and, insofar, they constitute a genuinely new species. They do art in postmodern times. When they started in the 1960s, they were often called computer artists, a term most of them hated. Meanwhile, their work is embraced by art history, they have conquered a small sector of the art market, and their mode of working has become ubiquitous.

The first narration will be about a kind of mathematical object. It is called a *polygon* and it plays a very important role. The narration is also about randomness, which at times is regarded as a machinic counterpart to creativity.

Three artists, Vera Molnar, Charles Csuri, and Manfred Mohr, will be the heroes of the second narration. It will be on certain aspects of their work pertaining to our general topic of creativity.

---

[2]There actually exists a group of artists who call themselves, "the algorists". The group is only loosely connected, they don't build a group in the typical sense of artists' groups that have existed in the history of art. The term *algorist* may have been coined by Roman Verostko, or by Jean-Pierre Hébert, or both. Manfred Mohr, Vera Molnar, Hans Dehlinger, Charles Csuri are some other algorists.

Two programs will be citizens first class in the third narration: Harold Cohen's *AARON* stands out as one of the most ambitious and successful artistic software development projects of all time. It is an absolutely exceptional event in the art world. Hardly known at all is a program Frieder Nake wrote in 1968/69. He boldly called it *Generative Art I*. The two programs are creative productions, and they were used for creative productions. Their approaches constitute opposite ends of a spectrum.

The chapter comes to its close with a fourth narration: on creativity. The first three ramblings lead up to this one. Is there a conclusion? There is a conclusion insofar as it brings this chapter to a physical end. It is no conclusion insofar as our stories cannot end. As Peter Lunenfeld has told us, digital media are caught in an aesthetics of the *unfinish* (Lunenfeld 1999, p. 7). I like to say the same in different words: the art in a work of digital art is to be found in the infinite class of works a program may generate, and not in the individual pieces that only *represent* the class.

I must warn the reader, but only very gently. There may occasionally be a formula from mathematics. Don't give up when you see it. Rather read around it, if you like. These creatures are as important as they are hard to understand, and they are as beautiful as any piece of art. People say, Mona Lisa's smile remains a riddle. What is different, then, between this painting and a formula from probability theory? Please, dear reader, enter postmodern times! We will be with you.

3.2 The First Narration: On Random Polygons

Polygons are often boringly simple figures when it comes to the generation of aesthetic, or even artistic objects. Nevertheless, they played an important role in the first days of computer art. Those days must be considered high days of creativity. Something great was happening then, something took on shape. Not many had the guts to clearly say this. It was happening at different places within a short time, and the activists were not aware of each other. Yet, what they did, was of the same kind. They surprised gallery owners who, of course, did not really like the art because, how could they possibly make money with it? With the computer in the background, this was mass production.

If the early pioneers themselves did not really understand the revolution they were causing, they left art critics puzzling even more. "Is it or is it not art?" was their typical shallow question, and: "Who (or what!) is the creator? The human, the computer, or the drawing automaton?" The simplest of those first creations were governed by polygons. Polygons became the signature of earliest algorithmic art. This is why I tell their story.

In mathematics, a *polygon* is a sequence of points (in the simplest case, in the plane). Polygons also exist in spaces of higher dimensions. As a sequence of points, the polygon is a purely mental construct. In particular and against common belief, you cannot *see* the polygon. As a polygon, it is invisible. It shares this fate with all of geometry. This is so because the objects of geometry—points, lines, planes—are pure. You describe them in formulae, and you prove theorems about them.

I cannot avoid writing down how a point, a straight line, and a plane are given explicitly. This must be done to provide a basis for the effort of an artist moving into this field. So the point in three-dimensional space is an unrestricted triple of coordinates, $P = (x, y, z)$. The straight line is constructed from two points, say P_1 and P_2, by use of one parameter, call it t. The values of t are real numbers, particularly those between 0 and 1. The parameter acts like a coordinate along the straight line. Thus, we can describe each individual point along the line by the formula

$$P(t) = P_1 + t(P_2 - P_1). \tag{3.1}$$

Finally, the points of a plane are determined from three given points by use of two parameters:

$$P(u, v) = uP_1 + vP_2 + (1 - u - v)P_3. \tag{3.2}$$

We need two parameters because the plane is spreading out into two dimensions whereas the straight line is confined to only one.

Bothering my readers with these formulae has the sole purpose that they should become aware of the different kind of thinking required here. Exactly describing the objects of hopefully ensuing creativity is only the start. It is parallel to the traditional artist's selection of basic materials. But algorithmic treatment must follow, if anything is going to happen (we don't do this here). The parameters u and v, I should add, can be any real numbers. The three points are chosen arbitrarily, but then are fixed (they must not be collinear).

As indicated above, all this is invisible. As humans, however, we want to see and, therefore, we render polygons visibly. When we do so, we interpret the sequence of points that make up the polygon, in an appropriate manner. The usual interpretation is to associate with each point a location (in the plane or in space). Next, draw a straight line from the first to the second point of the polygon, from there to the third point, etc. A closed polygon, in particular, is one whose first and last points coincide.

To draw a straight line, of course, requires that you specify the colour and the width of your drawing instrument, say a pencil. You may also want to vary the strokeweight along the line, or use a pattern as you move on. In short, the geometry and the graphics must be described explicitly and with utmost precision.

You have just learned your first and most important lesson: geometry is invisible, graphics is visible. The entities of geometry are purely mental. They are related to graphic elements. Only in them, they appear. Graphics is the human's consolation for geometry.

Let this be enough for a bit of formal and terminological background. We now turn to the first years of algorithmic art.[3] It is a well-established fact that between

[3]The art we are talking about, in the mid-1960s, was usually called *computer art*. This was certainly an unfortunate choice. It used a machine, i.e. the instrument of the art, to define it. This had not happened before in art history. *Algorithmic art* came much closer to essential features of the aesthetic endeavour. It does so up to this day. Today, the generally accepted term is *digital art*. But the digital principle of coding software is far less important than the algorithmic thinking in this art, at least when we talk about creativity. The way of thinking is the revolutionary and creative change. Algorithmic art is drawing and painting from far away.

1962 and 1964 three mathematicians or engineers, who on their jobs had easy and permanent access to computers, started to use those computers to generate simple drawings by executing algorithms. As it happened, all three had written algorithms to generate drawings and, without knowing of each other, decided to publicly exhibit their drawings in 1965. Those three artists are (below, examples of their works will be discussed):

- *Georg Nees* of Siemens AG, Erlangen, Germany, exhibited in the Aesthetic Seminar, located in rooms of the Studiengalerie of Technische Hochschule Stuttgart, Germany, from 5 to 19 February, 1965. Max Bense, chairing the institute, had invited Nees. A small booklet was published as part of the famous *rot* series for the occasion. It most likely became the first publication ever on visual computer art (Nees and Bense 1965).[4]
- *A. Michael Noll* of Bell Telephone Laboratories, Murray Hill, NJ, USA showed his works at Howard Wise Gallery in New York, NY, from 6 to 24 April, 1965 (together with random dot patterns for experiments on visual perception, by Bela Julesz; the exhibits were mixed with those of a second exhibition).
- *Frieder Nake* from the University of Stuttgart, Germany, displayed his works at Galerie Wendelin Niedlich in Stuttgart, from 5 to 26 November, 1965 (along with Georg Nees' graphics from the first show). Max Bense wrote an introductory essay (but could not come to read it himself).[5]

As it happens, there may have been one or two forgotten shows of similar productions.[6] But these three shows are usually cited as the start of digital art. The public appearance and, thereby, the invitation of critique, is the decisive factor if what you do is to be accepted as art. The artist's creation is one thing, but only a public reaction and critique can evaluate and judge it. The three shows, the authors, and the year define the beginning of algorithmic art.

From the point of view of art history, it may be interesting to observe that conceptual art and video art had their first manifestations around the same time. Op art had existed for some while before concrete and constructive art became influential. The happening—very different in approach—had its first spectacular events in the 1950s,

[4]The booklet, *rot 19*, contains the short essay, *Projekte generativer Ästhetik*, by Max Bense. I consider it to be the manifesto of algorithmic art, although it was not expressly called so. It has been translated into English and published several times. The term *generative aesthetics* was coined here, directly referring to Chomsky's generative grammar. The brochure contains reproductions of some of Nees' graphics, along with his explanations of the code.

[5]Bense's introductory text, in German, was not published. It is now available on the compArt Digital Art database at compart-bremen.de. Concerning the three locations of these 1965 exhibitions, Howard Wise was a well-established New York gallery, dedicated to avant-garde art. Wendelin Niedlich was a bookstore and gallery with a strong influence in the Southwest of Germany. The Studiengalerie was an academic (not commercial) institution dedicated to experimental and concrete art.

[6]Paul Brown recently (2009) discovered that Joan Shogren appears to have displayed computer-generated drawings for the first time on 6 May 1963 at San Jose State University.

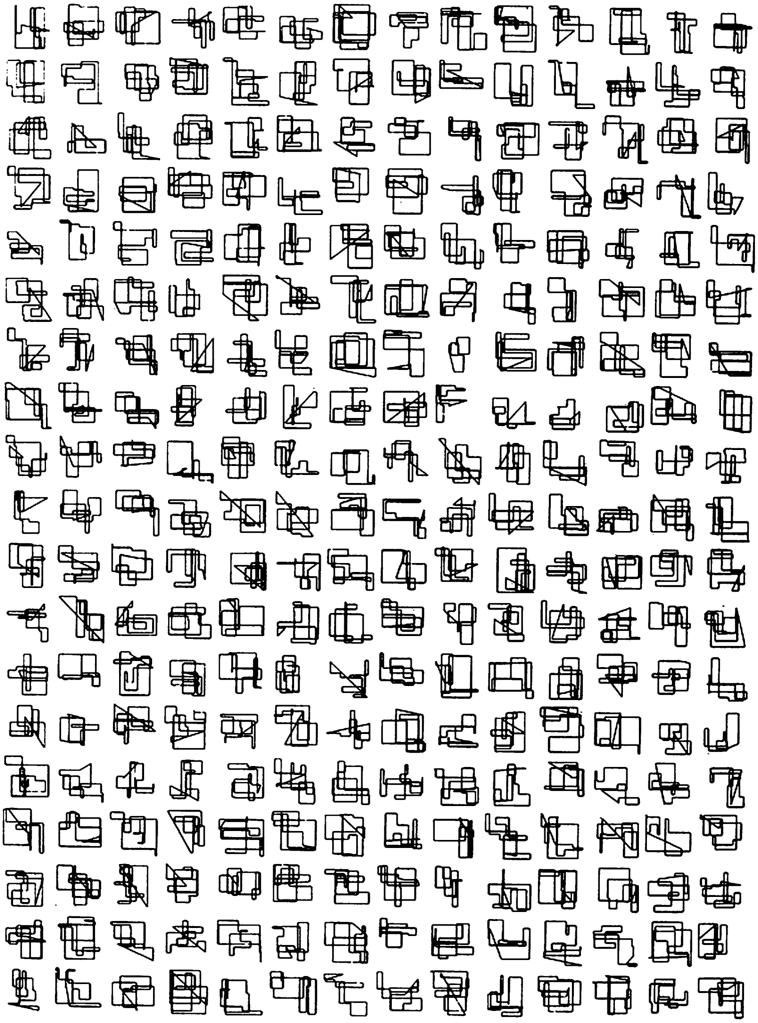


Fig. 3.1 Georg Nees: *23-Ecke*, 1965 (with permission of the artist)

and was continuing them. Pop art was, of course, popular. Serial, permutational, random elements and methods were being explored by artists. Kinetic art and light art were another two orientations of strong technological dependence. Max Bense had chosen the title *Programming the beautiful* (*Programmierung des Schönen*) for the third volume of his *Aesthetica* (Bense 1965), and Karl Gerstner had presented his book *Designing Programs* (*Programme entwerfen*, Gerstner 1963), whose second edition already contained a short section on randomness by computers.

But back to polygons! They appear in the works of the three above mentioned scientists-turned-artists among their very first experiments (Figs. 3.1, 3.2 and 3.3). We will now look at some of their commonalities and differences.

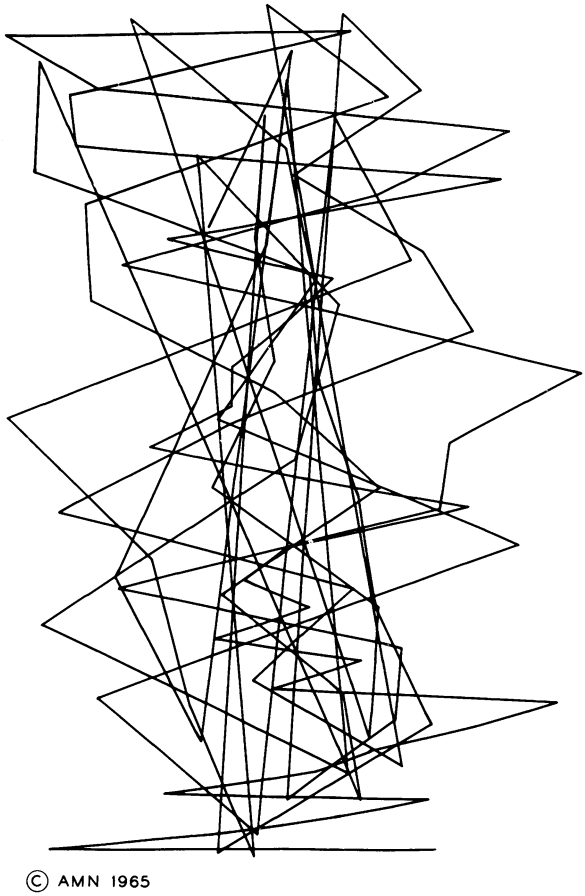


Fig. 3.2 A. Michael Noll: *Gaussian-Quadratic*, 1965 (with permission of the artist)

Assume you have at your disposal a technical device capable of generating drawings. Whatever its mode of operation may be, it is a mechanism whose basic and most remarkable operation creates a straight line-segment between two points. In such a situation, you will be quite content using nothing but straight lines for your aesthetic compositions. What else could you do? In a way, before giving up, you are stuck with the straight line, even if you prefer beautifully swinging curved lines.

At least for a start you will try to use your machine's capability to its very best before you begin thinking about what other and more advanced shapes you may be able to construct out of straight line-segments. Therefore, it was predictable (in retrospect, at least) that Nees, Noll, and Nake would come up with polygonal shapes of one or the other kind.

A first comment on creativity may be in order here. We see, in those artists' activities, the machinic limitations of their early works as well as their creative transcendence. The use of the machine: creative. The first graphic generations: boring. The use of short straight line-segments to draw bending curves: a challenge in creative

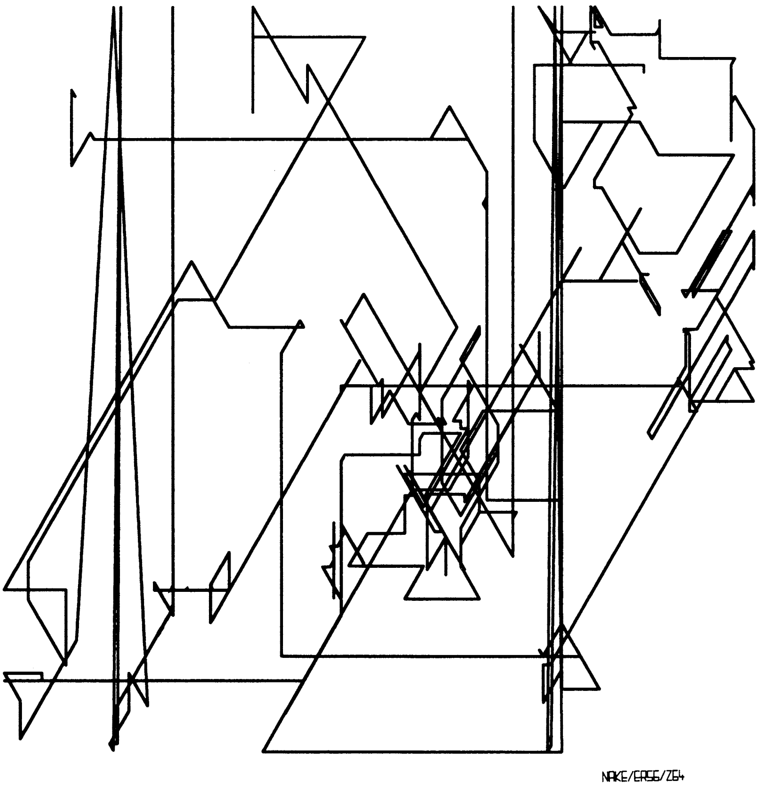


Fig. 3.3 Frieder Nake: *Random Polygon*, 1965

use of the machine. Turning to mathematics for the sake of art: creative, as well as nothing particularly exciting. Throughout the centuries, many have done this. But now the challenge had become to make a machine draw, whose sole purpose was calculation. How to draw when your instrument is not made for drawing?

3.2.1 Georg Nees

Although "polygons" were Nees', Noll's, and Nake's common first interest, their particular designs varied considerably. In six lines of ordinary German text, Nees describes what the machine is supposed to do (Nees and Bense 1965). An English translation of his pseudo-code reads like this:

> Start anywhere inside the figure's given square format, and draw a polygon of 23 straight line segments. Alternate between horizontal and vertical lines of random lengths. Horizontally go either left or right (choose at random), vertically go up or down (also random choice). To finish, connect start and end points by an oblique straight line.

Clearly, before we reach the more involved repetitive design of Fig. 3.1, this basic design must be inserted into an iterative structure of rows and columns. Once a specific row and a specific column have been selected, the empty grid cell located there will be filled by a new realisation of the microstructure just described. As we see from the figure, the composition of this early generative drawing is an invisible grid whose cells contain random 23-gons.

The random elements of Nees' description of the polygon guarantee that, in all likelihood, it will take thousands of years before a polygon will appear equal to, or almost equal to, a previous one. The algorithm creates a rich and complex image, although the underlying operational description appears as almost trivial. The oblique line connecting the first and last points adds a lot to the specific aesthetic quality of the image. It is an aberration from the rectilinear and aligned geometry of the main part of the polygons. This aberration from a standard is of aesthetic value: surprise.

There are $19 \times 14 = 266$ elementary figures arranged into the grid structure. Given the small size of the random shapes, we may, perhaps, not immediately perceive polygons. Some observers may identify the variations on a theme as a design study of a vaguely architectural kind.

The example demonstrates how a trivial composition can lead to a mildly interesting visual appearance not void of aesthetic quality. I postpone the creativity issue until we have studied the other two examples.

When some variable's value is chosen "at random", and this is happening by running a computer program, the concept of randomness must be given an absolutely precise meaning. Nothing on a computer is allowed to remain in a state of vagueness, even if vagueness is the expressed goal. And even if the human observer of the event does not see how he could possibly predict what will happen next, from the computer's position the next step must always be crystal clear. It must be *computable*, or else the program does nothing.

In mathematics, a *random variable* is a variable that takes on its values only according to a probability distribution. The reader no longer familiar with his or her highschool mathematics may recall that a formula like $y = x^2$ will generate the result $y = 16$ if $x = 4$ is given. If randomness plays a role, such a statement could only be made as a probability statement. This means the value of 16 may appear as the result of the computation, but maybe it does not, and the result is, say, 17 or 15.7.

Usually, when in a programming language you have a function that, according to its specification, yields random numbers, these numbers obey a so-called uniform probability distribution. In plain terms, this says that all the possible events of an experiment (like those of throwing dice) appear with the same probability.

But a random variable must not necessarily be uniformly distributed. Probability distributions may be more complex functions than the uniform distribution. In early algorithmic art, even of the random polygon variety, other distributions soon played some role. They simulated (in a certainly naïve way) the artist's intuition. (Does this sound like too bold a statement?)

3.2.2 A. Michael Noll

A. Michael Noll's "Gaussian-Quadratic" graphic makes use, in one direction (the horizontal, viz. Fig. 3.2), of the Gaussian distribution. The coordinates of vertices in the horizontal x-direction are chosen according to a Gaussian distribution, the most important alternative to the uniform distribution. The co-ordinates of vertices in vertical direction are calculated in a deterministic way (their values increase quadratically).

Whereas Nees' design follows a definite, if simple, compositional rule, Noll's is really basic: one polygon whose points are determined according to two distributions. It is not unfair to say that this is a simple visualisation of a simple mathematical process.

3.2.3 Frieder Nake

The same is true of Nake's polygon (Fig. 3.3). The algorithmic principle behind the visual rendition is exactly the same as that of Fig. 3.2: repeatedly choose an x- and a y-coordinate, applying distribution functions F_x and F_y, and draw a straight line from the previous point to the new point (x, y); let then (x, y) take on the role of the previous point for the next iteration.

In this formulation, F_x and F_y stand for functional parameters that must be provided by the artist when his intention is to realise an image by executing the algorithm.[7] Some experience, intuition, or creativity—whatever you prefer—flows into this choice.

The visual appearance of Nake's polygon may look more complex, a bit more like a composition. The fact that it owes its look to the simple structure of one polygon, does not show explicitly. At least, it seems to be difficult to visually follow the one continuous line that constitutes the entire drawing. However, we can clearly discover the solitary line, when we read the algorithm. The description of the simple drawing contains more (or other) facts than we see. So the algorithmic structure may disappear behind the visual appearance even in such a trivial case. Algorithmic simplicity (happening at the *subface* of the image, its invisible side) may generate visual complexity (visible *surface* of the image). If this is already happening in such trivial situations, how much more should we expect a non-transparent relation between simplicity (algorithmic) and complexity (visual) in cases of greater algorithmic effort?[8]

[7] Only a few steps must be added to complete the algorithm: a first point must be chosen, the total number of points for the polygon must be selected, the size of the drawing area is required, and the drawing instrument must be defined (colour, stroke weight).

[8] The digital image, in my view, exists as a double. I call them the *subface* and the *surface*. They always come together, you cannot have one without the other. The subface is the computer's view, and since the computer cannot see, it is invisible, but computable. The surface is the observer's view. It is visible to us.

This first result occurred at the very beginning of computer art. It is, of course, of no surprise to any graphic artist. He has experienced the same in his daily work: with simple technical means he achieves complex aesthetic results. The rediscovery of such a generative principle in the domain of algorithmic art is remarkable only insofar as it holds.

However, concerning the issue of creativity, some observers of early algorithmic experiments in the visual domain immediately started asking where the "generative power" (call it "creativity", if you like) was located. Was it in the human, in the program, or even in the drawing mechanism? I have never understood the rationale behind this question: human or machine—who or which one is the creator? But there are those who love this question.

If you believe in the possibility of answering such a question, the answer depends on how we first define "creative activity". But such a hope usually causes us to define terms in a way that the answer turns out to be what we want it to be. Not an interesting discussion.

When Georg Nees had his first show in February 1965, a number of artists had come to the opening from the Stuttgart Academy of Fine Art. Max Bense read his text on projects of generative aesthetics, before Nees briefly talked about technical matters of the design of his drawings and their implementation. As he finished, one of the artists got up and asked: "Very fine and interesting, indeed. But here is my question. You seem to be convinced that this is only the beginning of things to come, and those things will be reaching way beyond what your machine is already now capable of doing. So tell me: will you be able to raise your computer to the point where it can simulate my personal way of painting?"

The question appeared a bit as if the artist wanted to give a final blow to the programmer. Nees thought about his answer for a short moment. Then he said: "Sure, I will be able to do this. Under one condition, however: you must first explicitly tell me how you paint." (The artists appeared as if they did not understand the subtlety and grandeur, really: the dialectics of this answer. Without saying anything more, they left the room under noisy protest.)

When Nietzsche, as one of the earliest authors, experienced the typewriter as a writing device, he remarked that our tools participate in the writing of our ideas.[9] I read this in two ways. First, in a literal sense. Using a pencil or a typewriter in the process of making ideas explicit by formulating them in prose and putting this in visible form on paper, obviously turns the pencil or typewriter in my hand into a device without which my efforts would be in vain. This is the trivial view of the tool's involvement in the process of writing.

The non-trivial view is the observation that my thinking and attitude towards the writing process and, therefore, the content of my writing is influenced by the tool I'm using. My writing changes not only mechanically, but also mentally, depending on my use of tools. It still remains *my* writing. The typewriter doesn't write anything.

[9]Friedrich Kittler quotes Nietzsche thus: "Unser Schreibzeug arbeitet mit an unseren Gedanken." (Our writing tools participate in the writing of our thoughts.) (Kittler 1985), cf. Sundin (1980).

It is me who writes, even though I write differently when I use a pen than when I use a keyboard.

The computer is *not* a tool, but a machine, and more precisely: an automaton.[10] I can make such a claim only against a position concerning tools and machines and their relation. Both, machines and tools, are instruments that we use in work. They belong to the means of any production. But in the world of the means of production, tools and machines belong to different historic levels of development. Tools appear early, and long before machines. After the machine has arrived, tools are still with us, and some tools are hard to distinguish from machines. Still, to mix the two—as is very popular in the computing field where everything is called a "tool"—amounts to giving up history as an important category for scientific analysis. Here we see how the ideological character of so many aspects of computing presents itself.

Nietzsche's observation, that the tools of writing influence our thoughts, remains true. Using the typewriter, he was no longer forced to form each and every letter's shape. His writing became typing: he moved from the continuous flow of the arm and hand to the discrete hits of the fingers. We discover the digital fighting the analog: for more precision and control, but also for standardisation. Similarly, I give up control over spelling when I use properly equipped software (spell-checker). At the same time, I gain the option of rapid changes of typography and page layout.

If creation is to generate something that was not there before, then it is me who is creative. My creation may dwell on a trivial level. The more trivial, the easier it may be to transfer some of my creative operations onto the computer. It makes a difference to draw a line by hand from here to roughly there on a sheet of paper, as compared to issuing the appropriate command sequence, which I know connects points A and B. My thought must change. From "roughly here and there" to "precisely these coordinates".

My activity changes. From the immediate actor and generator of the line, I transform myself into the mediating specifier of conditions a machine has to obey when it generates the physical line. My part has become "drawing by brain" instead of "drawing by hand". I have removed myself from the immediacy of the material. I have gained a higher level of semioticity.

My brain helps me to precisely describe how to draw a line between any two points, whereas before I always drew just one line. It always was a single and particular line: this line right here. Now it has become: this is how you do it, independent of where you start, and where you end. You don't embark on the adventure of actually and physically drawing one and only one line. You anticipate the drawing of any line.

I am the creative one, and I remain the creator. However, the stuff of my creation has changed from material to semiotic, from particular to general, from single case to all cases. As a consequence, my thinking changes. I use the computer to execute a program. This is an enormous shift from the embodied action of moving the pencil. Different skills are needed, different thinking is required and enforced. Those who

[10] Cf. Sundin (1980).

claim the computer has become creative (if they do exist) have views that appear rather traditional. They do not see the dramatic change in artistic creation from material to sign, from mechanics to deliberate semiotics.

What is so dramatic about this transformation? Signs do not exist in the world. Other than things, signs require the presence of human beings to exist. Signs are established as relations between other entities, be they physical or mental. In order to appear, the sign must be perceived. In order to be perceivable, it must come in physical form. That form, however, necessary as it is, is not the most important correlate of the sign. Perceivable physical form is the necessary condition of the sign; the full sign, however, must be constituted by a cognitive act.

Semiotics is the study of sign processes in all their multitudes and manifestations. One basic question of semiotics is: how is communication possible? Semiotic answers to this question are descriptive, not explanatory.

3.3 The Second Narration: On Three Artists

It has often been pointed out that computer art originates in the work of mathematicians and engineers. Usually, this is uttered explicitly or implicitly with an undertone on “only mathematicians and engineers”.

The observation is true. Mathematicians and engineers are the pioneers of algorithmic art, but what is the significance of this observation? Is it important? What is the relevance of the “*only* mathematicians” qualification? I have always felt that this observation was irrelevant. It could only be relevant in a sense like: “early computer art is boring; it is certainly not worth being called art; and no wonder it is so boring—since it was not inspired by *real* artists, how could it be exciting”?

Frankly, I felt insulted a bit by the “only mathematicians” statement.[11] It implies a vicious circle. If art is only what artists generate, then how do you become an artist, if you are not born an artist? The only way out of this dilemma is that everyone is, in fact, born an artist (as not only Joseph Beuys has told us). But then the “only mathematicians” statement wouldn’t make sense any more.

People generate objects and they design processes. They do not generate art. Art, in my view, is a product of society—a judgement. Without appearing in public and thus without being confronted with a critique of historic and systematic origin, a work remains a work, for good or bad, but it cannot be said to have been included in the broad historic stream of art. Complex processes take place after a person decides to display his or her product in publicly accessible spaces. It is only in the public domain that art can emerge (as a value judgement!). Individuals and institutions in mutual interdependence are part of the processes that may merge to the judgement that a work is assessed and accepted as a work of “art”—often enough, as we all know, sparking even more controversy.

[11] This should read “mathematicians or engineers”, but I will stick to the shorter version.

In the course of time, it often happens that an individual person establishes herself or himself stably or almost irrevocably in the hall of art. Then she or he can do whatever they want to do, and still get it accepted as "art". But the principle remains.[12]

The "only mathematician" statement is relevant only insofar as it is interpreted as "unfortunately the pioneers were only mathematicians. Others did not have access to the machines, or did not know how to program. Therefore we got the straight-line quality of early works."

However, if we accept that a work's quality as a work of *art* is judged by society anyhow, the perspective changes. Mathematician or bohemian does not matter then. There cannot be serious doubt that what those pioneering mathematicians did caused a revolution. They separated the generation of a work from its conception. They did this in a technical way. They were interested in the operational, not only mental separation. No wonder that conceptual art was inaugurated at around the same time. The difference between conceptual and computational art may be seen in the computable concepts that the computer people were creating.

However, when viewed from a greater distance, the difference between conceptual artists and computational artists is not all that great. Both share the utmost interest in the idea (as opposed to the material), and Sol LeWitt was most outspoken on this. The early discourse of algorithmic art was also rich about the immaterial character of software. Immaterial as software may be, it does not make sense without being executed by a machine. A traditionally described concept does not have such a surge to execution.[13]

The pioneers from mathematics showed the world that a new principle had arrived in society: the algorithmic principle! No others could have done this, certainly not artists. It had to be done by mathematicians, if it was to be done at all. The parlance of "only mathematicians" points back to the speaker more than to the mathematician.

Trivial to note is that creative work in art, design, or any other field, depends on ideas on one hand, and skills on the other. At times it happens that someone has a great idea but just no way to realise it. He or she depends on others to do that. Pushing things a bit to the extreme, the mathematics pioneers of digital art may not have had great ideas, but they knew how to realise them.

---

[12]Marcel Duchamp was the first to talk and write about this: "All in all, the creative act is not performed by the artist alone; the spectator brings the work in contact with the external world by deciphering and interpreting its inner qualification and thus adds his contribution to the creative act. This becomes even more obvious when posterity gives a final verdict and sometimes rehabilitates forgotten artists." (Duchamp 1959). This position implies that a work may be considered a work of art for some while, but disappear from this stage some time later, a process that has often happened in history. It also implies that a person may be considered a great artist only after his or her death. That has happened, too.

[13]It is a simplification to concentrate the argument on conceptual vs. algorithmic artists. There have been other directions for artistic experiments, in particular during the 1960s. They needed a lot of technical skill and constructive intelligence or creativity. Recall op art, kinetic art, and more. Everything that humans eventually transfer to a machine has a number of precursors.

On the other hand, artists may have had great ideas and lots of good taste and style, but no way of putting that into existence. So who is to be blamed first? Obviously, both had to acquire new and greater knowledge, skills, and feelings. They had to learn from each other. Turning the argument around, we come up with "unfortunately, some were only artists and therefore had no idea how to do it." Doesn't this sound stupid? It sounds as stupid the other way around.

So let us take a look at what happened when artists wanted, and actually managed, to get access to computers. As examples I have chosen Vera Molnar, Charles Csuri, and Manfred Mohr. Many others could be added. My intent, however, is not to give a complete account, a few cases are enough to make the point.

3.3.1 Vera Molnar

Vera Molnar was born in Hungary in 1924 and lived in Paris. She worked on concrete and constructive art for many years. She tried to introduce randomness into her graphic art. To her great dismay, however, she realised that it is hard for a human to avoid repetition, clusters, trends, patterns. "Real" randomness does not seem to be a human's greatest capability.

So Vera Molnar decided that she needed a machine to do parts of her job. The machine would not be hampered by the human subjectivity that seems to get in the way of a human trying to do something randomly. The kind of machine she needed was a computer that, of course, she had no access to. Vera Molnar felt that systematic as well as hazardous ways of expressing and researching were needed for her often serial and combinatorial art. Since she did not have the machine to help her to do this, she had to build one herself. She did it mentally: "I imagined I had a computer" (Herzogenrath and Nierhoff 2006, p. 14). Her *machine imaginaire* consisted of exactly formulated rules of behaviour. Molnar simulated the machine by strictly doing what she had told the imaginary machine to do.

In 1968, Vera Molnar finally gained access to a computer at the Research Centre of the computer manufacturer, Bull. She learned programming in Fortran and Basic, but also had people to help her. She did not intend to become an independent programmer. Her interests were different. For her, the slogan of the computer as a tool appears to be justified best. She allowed herself to change the algorithmic works by hand. She made the computer do what she did not want to do herself, or what she thought the machine was doing more precisely.[14]

Figure 3.4 (left)[15] shows one of her early computer works. She had previously used repertoires of short strokes in vertical, horizontal, or oblique directions, sim-

[14]The catalogue (Herzogenrath and Nierhoff 2006) contains a list of the hardware Vera Molnar has used since 1968. It also presents a thorough analysis of her artistic development. The catalogue appeared when Molnar became the first recipient of the *d.velop digital art award.* A great source for Molnar's earlier work is Hollinger (1999).

[15]This figure consists of two parts: a very early work, and a much later one by the same artist. The latter one is given without any comment to show an aspect of the artist's development.

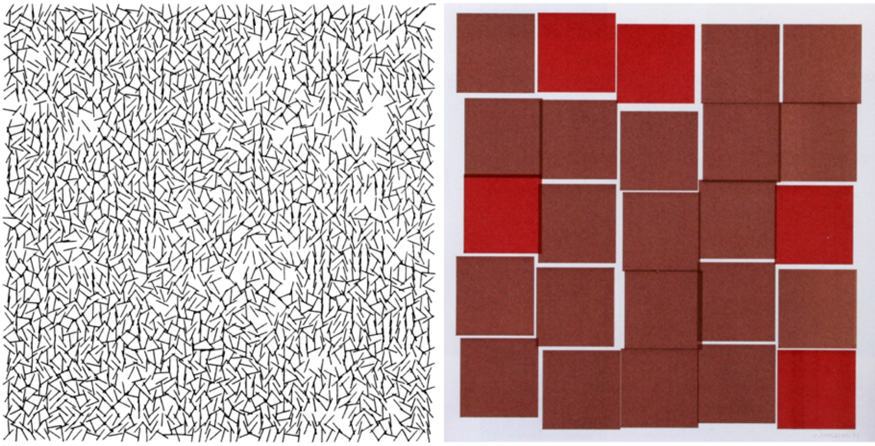


Fig. 3.4 Vera Molnar. *Left*: *Interruptions*, 1968/69. *Right*: *25 Squares*, 1991 (with permission of the artist)

ilar in style to what many of the concrete artists had also done. The switchover to the computer gave her the opportunity to do more systematic research. ("Visual research" was a term of the time. The avantgarde loved it as a wonderful shield against the permanent question of "art". Josef Albers and others from the Bauhaus were early users of the word.)

The *Interruptions* of Fig. 3.4 happen in the open spaces of a square area that is densely covered by oblique strokes. They build a complex pattern, a texture whose algorithmic generation, simple as it must be, is not easy to identify. The open areas appear as surprise. The great experiment experienced by pioneers of the mid-1960s shows in Molnar's piece: what will happen visually if I force the computer to obey a simple set of rules that I invent? How much complexity can I generate out of almost trivial descriptions?

3.3.2 Charles Csuri

Our second artist who took to the computer is Charles Csuri. He is a counter example to the "only mathematicians" predicament. Among the few professional artists who became early computer users, Csuri was probably the first. He had come to Ohio State University in Columbus from the New York art scene. His entry into the computer art world was marked by a series of exceptional pieces, among them *Sine Curve Man* (Fig. 3.5, left), *Random War*, and the short animated film *Hummingbird* (for more on Csuri and his art, see Glowski 2006).

Sine Curve Man won him the first prize of the *Computer Art Contest* in 1967. Ed Berkeley's magazine, *Computers and Automation* (later renamed to *Computers and People*), had started this yearly contest. It was won in 1965 by A. Michael Noll,

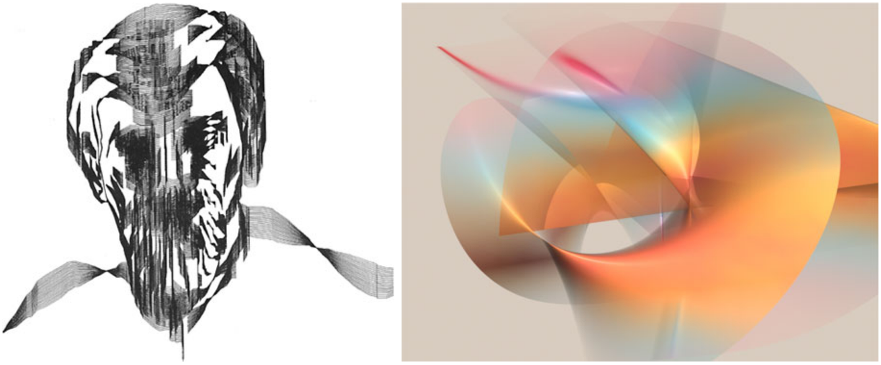


Fig. 3.5 Charles Csuri. *Left*: *Sine Curve Man*, 1967. *Right*: *yuck 4x3*, 1991 (with permission of the artist)

1966 by Frieder Nake, and then by Csuri, an educated artist for the first time. This award, by the way, never gained high esteem. It took many more years, until 1987, when the now extremely prestigious *Prix Ars Electronica* was awarded for the first time.

For his first programming tasks, Csuri was assisted by programmer James Shaffer. Similar to Vera Molnar, we see that the skill of programming may at the beginning constitute a hurdle that is not trivial to master. If time plays a role, an artist willing to use the computer, but still unable to do it all by himself, has almost no choice but to rely on friendly cooperation. Such cooperation may create friction with all its negative effects. As long as the technical task itself does *not* require cooperation, it is better to acquire the new technical skill. After all, there is no art without skillful work, and a steadily improved command of technical skills is a necessary condition for the artist. Why should this be different when the skill is not the immediate transformation of a corporeal material by hand, but instead the description only of relations and conditions, of options and choices of signs?

Csuri's career went up steeply. Not only did he become the head of an academic institute but even an entrepreneur. At the time of a first rush for large and leading places in computer animation, when this required supercomputers of the highest technological class and huge amounts of money, he headed the commercial Cranston Csuri Productions company as well as the academic Advanced Computing Center for the Arts and Design, both at Columbus, Ohio. In the year 2006, Csuri was honoured by a great retrospective show at the ACM SIGGRAPH yearly conference.

Sine Curve Man is an innovation to computer art of the first years in two respects: its subject matter is figurative, and it uses deterministic mathematical techniques rather than probabilistic. There is a definite artistic touch to the visual appearance of the graphic (Fig. 3.5), quite different from the usual series of precise geometric curves that many believe computer art is (or was) about.

The attraction of *Sine Curve Man* has roots in the graphic distortions of the (old?) man's face. Standard mathematics can be used for the construction. A lay person may, however, not be familiar with such methods. Along the curves of an original drawing, a series of points are marked. The curves may, perhaps, have been extracted from a photograph. The points become the fixed points of interpolations by sums of *sine* functions. This calculation, purely mathematical as it is, and without any intuitive additions triggered by the immediate impression of a seemingly half-finished drawing, is an exceptional case of the new element in digital art.

This element is the dialectics of aesthetics and algorithmics. *Sine Curve Man* may cause in an observer the impression that something technical is going on. But this is probably not the most important aspect. More interesting is the visual (i.e. aesthetic) sensation. The distortions this man has suffered are what attracts us. We are almost forced to explore this face, perhaps because we want to read the curves as such. But they do not allow us to do this. Therefore, our attention cannot rest with the mathematics. Dialectics happens, as well as semioses (sign processes): jumping back and forth between semantics and syntactics.

3.3.3 Manfred Mohr

Manfred Mohr is a decade younger than the first two artists. They belong to the first who were accepted by the world of art despite their use of computers. Do they owe anything to computers? Hard to say. An art historian or critic will certainly react differently if he doesn't see an easel in the artist's studio, but a computer instead.

The artist doesn't owe much to a computer. He has decided to use it, whatever the reason may have been. If to anything, he owes to the programs he is using or has written himself. With those programs, he calls upon work formerly spent that he now is about to set in action again. The program appears as canned labour ready to be resuscitated.

The relation between artist and computer is, at times, romanticised as if it were similar to the close relation between the graphic artist and her printer (a human being). The printer takes pride in getting the best quality out of the artist's design. The printing job takes on artistic quality itself. The computer, to the contrary, is only executing a computable function. It should be clear, that the two cases are as different as they could ever be.

If we characterise Vera Molnar, in one word, as the grand old lady of algorithmic art, and Charles Csuri as the great entrepreneur and mover, Manfred Mohr would appear as the strictest and strongest creator of a style in algorithmic art. The story says that his young and exciting years of searching for his place in art history were filled with jamming the saxophone, hanging out in Spain and France, and with hard edge constructivist paintings. Precision and rationality became and remained his values. They find a correspondence and a balancing force in the absolute individual freedom of jazz. Like many of the avant-garde artists in continental Europe during the 1960s, he was influenced by Max Bense's theory and writing on aesthetics, and when he read in a German news magazine (Anon 1965) that computers had appeared in fine art, he knew where he had to turn to.

K.R.H. Sonderborg and the art of *Informel*, Pierre Barbaud and electronic music, Max Bense and his theory of the aesthetic object constitute a triad of influences from

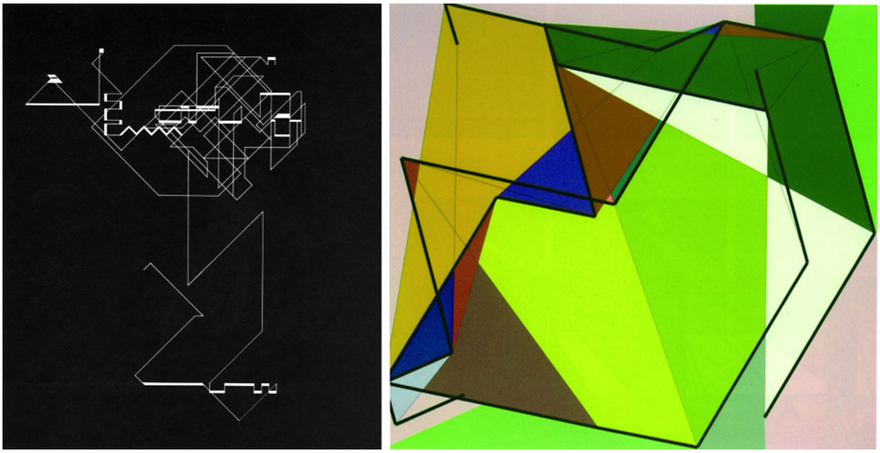


Fig. 3.6 Manfred Mohr. *Left*: *P-18* (*Random Walk*), 1969. *Right*: *P-707-e1* (*space.color*), 1999–2001 (with permission of the artist)

which Mohr's fascinating generative art emerged. From his very first programmed works in 1969 to current days, he has never betrayed his striving for the greatest transparency of his works. Never did he leave any detail of his creations open to hand-waving or to dark murmurs. He discovered the algorithmic description of the generative process as the new creation. The simplest elements can become the material for the most complex visual events.

After about four years of algorithmic experiments with various forms and relations, Manfred Mohr, in 1973, decided to use the cube as the source of external inspiration. He has continued exploring it ever since. There are probably only a few living persons who have celebrated and used the cube more than him (for further information see Keiner et al. 1994, Herzogenrath et al. 2007).

Figure 3.6 shows one event in the six-dimensional hypercube (right), and one of the earliest generative graphics of Mohr's career (left).

When we see a work by Mohr, we immediately become aware of the extraordinary aesthetic quality of his work. His decisions are always strong and secure. The random polygon of Fig. 3.6 is superior to most, if not all, of the others one could see in the five years before. The events of the heavier white lines add an enormous visual quality to the drawing, achieved in such strength here for the first time.

The decision, in 1973, to explore the three-dimensional cube as a source for aesthetic objects and processes, put Manfred Mohr in a direct line with all those artists who, at least for some part of their artistic career, have explored one and the same topic over and over again. It should be emphasised, however, that his interest in the cube and the hypercube[16] does not signify any pedagogical motif. He does not intend to explain anything about spaces of higher dimensions, nor does he visualise

[16]The hypercube is analogous to a three-dimensional cube in four or more dimensions. It is recursively defined as an intricate structure of cubes.

cubes in six or eleven dimensions. He takes those mental creatures as the rational starting points for his visual creation. The hypercube is only instrumental in Mohr's creative work; it is not the subject matter.

The cube in four or more dimensions is a purely mental product. We can clearly *think* the hypercube. But we cannot visualise it. We may take the hypercube as the source of visual aesthetic events (and Mohr does it). But we cannot show it in a literal sense of the word. Manfred Mohr's mental hikes in high dimensions are his inspiration for algorithmic concrete images. For these creations, he needs the computer. He needs it even more when he allows for animation.

Manfred Mohr's work stands out so dramatically because it cannot be done without algorithms. It is the most radical realisation of Paul Klee's announcement: *we don't show the visible, we make visible.* The image is a visible signal. What it shows is itself. It has a source elsewhere. But the source is not shown. It is the only reason for something visible.

Creativity? Yes, of course, piles of. Supported by computer? Yes, of course, in the trivial sense that this medium is needed for the activity of realising something the artist is thinking of. In Manfred Mohr's work (and that of a few others whose number is increasing) generative art has actually arrived. The actuality of his work is its virtuality.

3.4 The Third Narration: On Two Programs

Computer programs are, first of all, *texts*. The text describes a complex activity. The activity is usually of human origin. It has before existed as an activity carried out by humans in many different forms. When it becomes the source of an algorithmic description, it may gradually disappear as a human activity, until in the end, the computer's (or rather the program's) action appears as the first and more important than the human activities that may still be needed to keep the computer running: human-supported algorithmic work.

The activity described by a computer program as a text may be almost trivial, or it may be extremely complex. It may be as trivial as an approximate calculation of the *sine* function for a given argument. Or it may be as complex as calculating the weather forecast for the area of France by taking into account all available atmospheric measurements collected around the world.

The art of writing computer programs has become a skill of utmost creativity, intuition, constructive precision, and secrets of the trade. Donald Knuth's marvellous series of books, *The Art of Computer Programming*, is the best proof of this (Knuth 1968). These books are one of the greatest attempts to give an in-depth survey of the entire field of computing. It is almost impossible to completely grasp this field in totality, or even to finish writing the series of books. Knuth is attempting to do just this.

Computer programs have been characterised metaphorically as tools, as media, or as automata. How can a program be an automaton if it is, as I have claimed, a text? The answer is in the observation that the computer is a *semiotic machine* (Nadin 2011, Nöth 2002, Nake 2009).

The computer is seen by these authors as a semiotic machine, because the stuff it processes is of a semiotic nature. When the computer is running, i.e. when it is working as a machine, it is executing a program. It is doing this under the control of an operating system. The operating system is itself a program. The program, that the computer is executing, takes data and transforms it into new data. All these creatures—the operating system, the active program, and data—are themselves of semiotic nature. This chapter is not the place to go deeper into the semiotic nature of all entities on a computer.[17] So let us proceed from this basic assumption.

The assumption becomes obvious when we take a look at a program as a text. Leaving aside all detail, programming starts from a more or less precise specification of what a program should be doing. Then there is the effort of a group of programmers developing the program. Their effort materialises in a rich mixture of activities. Among these, the *writing of code* is central. All other kinds of activities eventually collapse into the writing of code.

The finished program, which is nothing but the code for the requested function, appears as a text. During his process of writing, the programmer must read the text over and over again. And here is the realisation: the computer is also reading the text! The kind of text that we call "computer program" constitutes a totally new kind of poetry. The poetics of this poetry reside in the fact that it is written for two different readers: one of them human, the other machine.

Their fantastic semiotic capabilities single out humans from the animal kingdom. Likewise, the computer is a special machine because of its fantastic semiotic capabilities. Semiotic animal and semiotic machine meet in reading the text that is usually called a *program*.

Now, reading is essentially interpreting. The human writer of the program materialises in it the specification of some complex activity. During the process of his writing, he is constantly re-reading his product as he has so far written it. He is convinced of the program's "correctness". It is correct as long as it does what it is supposed to do. However, how may a text be actively *doing* anything?

The text can do something only if the computer is also reading it. The reading, and therefore interpreting, of the program by the computer effectively transforms the text into a machine. The computer, when reading the program text (and therefore: interpreting it), cannot but execute it. Without any choice, reading, interpreting, and executing the text are one and the same for the computer. The program as a text is interesting for the human only insofar as the computer is brought to execute it. During execution, the program reveals its double character as text-and-machine, both at the same time. So programs are executable texts. They are texts as machine, and machine as text.

After this general but also concrete remark about what is new in postmodern times, we take a look at two specific and ambitious, albeit very different programs.

[17] A book is in preparation that takes a fundamental approach to this topic: P.B. Andersen & F. Nake, *Computers and signs. Prolegomena to a semiotic foundation of computing*.

We don't look at their actual code because this is not necessary for our discussion of creativity in early computer art. Harold Cohen's famous *AARON* started its astonishing career in 1973, and continued to be developed for decades. Frieder Nake's *Generative Aesthetics I* was written, completed, then discarded in the course of one year, 1968/69.

3.4.1 Harold Cohen: AARON

AARON is a rule-based system, an enormous expert system, one of the very few expert systems that ever made it to their productive phase (McCorduck 1990). In the end it consisted of so many rules that its sole creator, Cohen, was no longer sure if he was still capable of understanding well enough their mutual dependencies.

Everything on a computer must be rule-based. A rule is a descriptive element of the structure: `if` C `then` A, where C is a condition (in the logical sense of "condition"), and A is an action. In the world of computing, a formal definition must state precisely what is accepted as a C, and what is accepted as an A. In colloquial terms, an example could be: `if (figure ahead) then (turn left or right)`. Of course, the notions of "figure", "ahead", "turn", "left", "right" must also be described in computable terms, before this can make any sense to a computer.

A rule-based system is a collection of interacting rules. Each rule is constructed as a pair of a condition and an action. The condition must be a description of an event depending on the state (value) of some variables. It must evaluate to one of the truth-values `true` or `false`. If its value is `true`, the action is executed. This requires that its description is also given in computable form. The set of rules making up a rule-based system may be structured into groups. There must be an order according to which rules are tested for applicability. One strategy is to apply the first applicable rule in a given sequence of rules. Another one determines all applicable rules and selects one of them.

Cohen's AARON worked for many years during which it produced a large collection of drawings. They were first in black and white. Later, Cohen coloured them by hand according to his own taste or to codes also determined by AARON. The last stage of AARON relied on a new painting machine. It was constructed such that it could mimic certain painterly ways of applying paint to paper.

During more than three decades, AARON's command of subjects developed from collections of abstract shapes to evocations in the observer of rocks, birds, and plants, and to figures more and more reminiscent of human beings. They gave the impression of a spatial arrangement, although Cohen never really entered into three dimensions. A layered execution of figures was sufficient to generate a low-level of spatial impression.

Around the year 2005, Cohen became somewhat disillusioned with the figural subjects he had gradually programmed AARON to better and better create. When he started using computers and writing programs in the early 1970s, he was fascinated

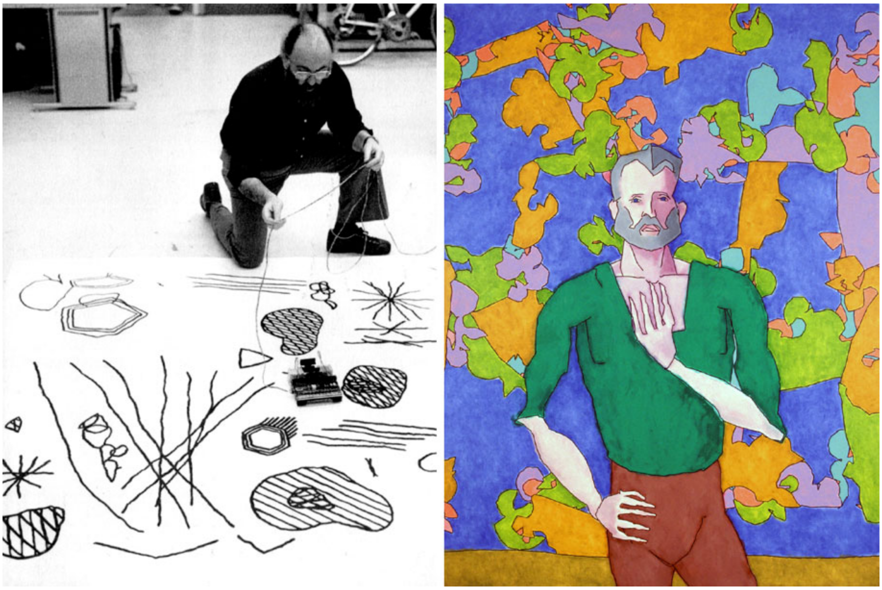


Fig. 3.7 Harold Cohen. *Left*: Early drawing by *AARON*, with the artist. *Right*: Drawing by *AARON*, 1992 (with permission of the artist)

by the problem of representation. His question then was: just how much, or little, does it take before a human observer securely recognises a set of lines and colours as a figure or pattern of something? How could a painting paint itself? (Cohen 2007).

But Harold Cohen has now stopped following this path any further. He achieved more than anyone else in the world in terms of creating autonomous rule-based art systems (Fig. 3.7 shows two works along the way). He did not give up this general goal. He decided to return to pure form and colour as the subject matter of his autonomous rule-based system.

For a computer scientist, there is no deep difference between an algorithm and a rule-based system. As Cohen (2007) writes, it took him a while to understand this. The difference is one of approach, not of the results. Different approaches may still possess the same expressive power. As Cohen is now approaching colour again in an explicitly algorithmic manner, he has shifted his view closer to the computer scientist's but without negating his deep insight into the qualities of colour as an artist.

This is marvellous. After a long and exciting journey, it sheds light on the alleged difference between two views of the world. In one person's great work, in his immediate activity, experience, and knowledge, the gap between the "two cultures" of C.P. Snow fades. It fades in the medium of the creative activity of one person, not in the complex management of interdisciplinary groups and institutes. The book must still be written that analyses the Cohen decades of algorithmic art from the perspective of art history.

Cohen's journey stands out as a never again to be achieved adventure. He has always been the lonely adventurer. His position is unique and singular. Artificial

Intelligence people have liked him. His experience and knowledge of rule-based systems must be among the most advanced in the world. But he was brave enough to see that in art history he had reached a dead-end. Observers have speculated about when would AARON not only be Cohen's favourite artist, but also its own and best critic. Art cannot be art without critique. As exciting as AARON's works may be, they were slowly losing their aesthetic appeal, and were approaching the only evaluation: oh, would you believe, this was done by computer? The dead-end.

Harold Cohen himself sees the situation with a bit more skepticism. He writes:

> It would be nice if AARON could tell me which of them [its products] it thinks I should print, but it can't. It would be nice if it could figure out the implications of what it does so well and so reliably, and move on to new definitions, new art. But it can't. Do those things indicate that AARON has reached an absolute limit on what computers can do? I doubt it. They are things on my *can't-do-that* list... (Cohen 2007).

The *can't-do-that* list contains statements about what the computer can and what it cannot do. During his life, Cohen has experienced how items had to be removed from the list. Every activity that is computable must be taken from the list. There are activities that are not computable. However, the statement that something cannot be done by computer, i.e. is not computable, urges creative people to change the non-computable activity into a computable one. Whenever this is achieved after great hardship, we don't usually realise that a new activity, a computable one, has been created with the goal in mind to replace the old and non-computable.

There was a time, when Cohen was said to be on his way to becoming the first artist of whom there would still be new works in shows after his death. He himself had said so, jokingly with a glass of cognac in hand. He had gone so far that such a thought was no longer fascinating. The Cohen manifesto of algorithmic art has reached its prediction.

But think about the controversial prediction once more. If true, would it not be proof of the computer's independent creativity? Clearly, Cohen wrote AARON, the program, the text, the machine, the text-become-machine. This was *his*, Cohen's creative work. But AARON was independent enough to then get rid of Cohen, and create art all by itself. How about this?

In a trivial sense, AARON is creative, but this creativity is a pseudo-creativity. It is confined to the rules and their certainly wide spectrum of possibilities. AARON will forever remain a technical system. Even if that system contained some meta-rules capable of changing other rules, and meta-meta-rules altering the meta-rules on the lower level, there would always be an explicit end. AARON would not be capable of leaving its own confines. It cannot cross borders.

Cohen's creativity, in comparison, stands out differently. Humans can always cross borders. A revolution has happened in the art world when the mathematicians demonstrated to the artists that the individual work was no longer the centre of aesthetic interest. This centre had shifted to descriptions of processes. The individual work had given way to the *class of works*. Infinite sets had become interesting, the individual work was reduced to a by-product of the class. It has now become an instance only, an index of the class it belongs to.

No doubt, we need the instance. We want to literally see something of the class. Therefore, we keep an interest in the individual work. We cannot see the entire class. It has become the most interesting, and it has become invisible. It can only be thought.

I am often confronted with an argument of the following kind. A program is not embedded into anything like a social and critical system, and clearly, without a critical component, it cannot leave borders behind. So wait, the argument says, until programs are embedded the proper way.

But computers and programs don't even have bodies. How then should they be able to be embedded in such critical and social systems? Purpose and interest are just not their thing. Don't you, my dear friends, see the blatant difference between yourself and your program, between you and the machine?

Joseph Weizenbaum dedicated much of his life to convincing others of this fundamental difference. It seems to be very tough for some of us to accept that we are not like machines and, therefore, they are not like us.

3.4.2 Frieder Nake: Generative Aesthetics I

A class of objects can never itself, as a class, appear physically. In other words, it cannot be perceived sensually. It is a mental construct: the description of processes and objects. The work of art has moved from the world of corporeality to the world of options and possibilities. Reality now exists in two modes, as actuality and virtuality.

AARON's generative approach is activity-oriented. The program controls a drawing or painting tool whose movements generate, on paper or canvas, visible traces for us to see. The program *Generative Aesthetics I*, however, is algorithm-oriented. It starts from a set of data, and tries to construct an image satisfying conditions that are described in the data.

You may find details of the program in Nake (1974, pp. 262–277). The goal of the program was derived from the theory of information aesthetics. This theory starts by considering a visual artefact as a sign. The sign is really a *supersign* because it is usually realised as a structure of signs.

The theory assumes that there is a repertoire of elementary or primitive signs. Call those primitive signs: $s_1, s_2, \ldots, s_r$. They must be perceivable as individual units. Therefore, they can be counted, and relative frequencies of their occurrence can be established. Call those frequencies, $f_1, f_2, \ldots, f_r$.

In information aesthetics, a schema of the signs with their associated relative frequencies is called a *sign schema*. It is a purely statistical description of a class of images. All those images belong to the class that use the same signs (think of colours) with the same frequencies.

In Shannon's information theory, the statistical measure of information in a message is defined as

$$H = -\sum_{i=1}^{r} p_i \log p_i . \tag{3.3}$$

The assumption for the derivation of this formula in Shannon and Weaver (1963) is that all the p_i are probabilities. They determine the statistical properties of a source sending out messages that are constructed according to the probabilities of the source.

This explanation may not mean much to the reader. For one, information theory is no longer popular outside of certain technical contexts. Moreover, it was overestimated in the days when the world was hoping for a great unifying theory. The measure H gives an indication of what we learn when one specific event (out of a set of possible events) has occurred, and we know what the other possible events could be.

Take as an example the throwing of dice in a typical board game. As we know, there are six possible events, which we can identify by the numbers 1, 2, 3, 4, 5, and 6. Each one of the six events occurs with the same probability, i.e. 1/6. Using Shannon's formula for the information content of the source "dice", we get

$$H = -\log(1/6) = -(\log 1 - \log 6) = \log 6 \approx 2.6 \tag{3.4}$$

(the logarithm must be taken to the base of 2). The result is measured in bits and must be interpreted thus: when one of the possible results of the throw has appeared, we gain between two and three bits of information. This, in turn, says that between two and three decisions of a "yes or no" nature have been taken. The Shannon measure of information is a measure of the uncertainty that has disappeared when and because the event has occurred.

Information aesthetics, founded by Max Bense and Abraham A. Moles (Bense 1965, Moles 1968) and further developed in more detail by others (Gunzenhäuser 1962, Frank 1964), boldly and erroneously ignored the difference between frequency and probability. To repeat, probabilities of a sign schema characterise an ideal source. Frequencies, however, are results of empirical measurement of several, but only finitely many messages or events (images in our case). As such, frequencies are only estimates for probabilities.

Information aesthetics wanted to get away from subjective value judgement. Information aesthetic criteria were to be objective. Aspects of the observer were excluded, at least in Max Bense's approach. Empirical studies from the 1960s and later were, however, not about aesthetic sources, but about individual pieces. In doing so, the difference of theory and practice, of infinite class and individual instance, of probability and frequency, had to be neglected by replacing theoretical probability by observed frequency, thus $p_i = f_i$. This opened up the possibility to measure the object without any observer being present. However, the step also gave up aesthetics as the theory of sensual perception.

Now, the program *Generative Aesthetics I* accepted as input a set of constraints of the following kind. For each sign (think of colour), a measure of surprise and a measure of conspicuity (defined by Frank in 1964) could be constraint to an interval of feasible values. Such requirements defined a set of up to $2r$ constraints. In addition, the aesthetic measure that Gunzenhäuser had defined as an information-theoretic analogue to Birkhoff's famous but questionable measure of "order in complexity" (Birkhoff 1931) could be required to take on a maximum or minimum value, relative to the constraints mentioned before. Requesting a maximum to be the goal of construction put trust on the formal definition of aesthetic measure actually yielding a good or even beautiful solution. Requesting a minimum, to the contrary, did not really trust the formalism.

With such a statement of the problem, we are right into mathematics. The problem turns out to be a non-linear optimisation problem. If a solution is possible, it had to be a discrete probability distribution. This distribution represents all images satisfying the constraints. it was called "the statistical pre-selector," since it was based only on a statistical view of the image. In a second step, a topological pre-selector took the sign schema of the previous step and created the image as a hierarchical structure of colour distribution, according to the probabilities determined before.

The type of structure used for this construction of the image was, in computer science, later called a *quadtree*. A quadtree divides an image into four quadrants of equal size. The generative algorithm distributes the probabilities of the entire image into the four smaller quadrants such that the sum total remains the same. With each quadrant, the procedure is repeated recursively, until a quadrant is covered by one colour only, or its size has reached a minimal length.

Generative Aesthetics I thus bravely started from specifying quantitative criteria that an image was to satisfy. Once the discrete probability distribution was determined as a solution to the set of criteria, an interesting process of many degrees of freedom started to distribute the probabilities into smaller and smaller local areas of the image but such that the global condition was always satisfied. Aesthetics happened generatively and objectively, by running an automaton.

The program was realised in the programming language PL/I with some support from Fortran routines. Its output was trivial but fast. I was working on this project in Toronto in 1968/69. Since no colour plotter was available, I used the line printer as output device. The program's output was a list of measures from information aesthetics plus a coded printout of the generated image. I used printer symbols to encode the colours that were to be used for the image.

This generative process was very fast, which allowed me to run a whole series of experiments. These experiments may constitute the only ones ever carried out in the spirit of generative aesthetics based on the Stuttgart school of information aesthetics. The program was intended to become the base for empirical research into generative aesthetics. Regrettably, this was not realised.

With the help of a group of young artists, I realised by hand only two of the printouts. From a printer's shop we got a set of small pieces of coloured cardboard. They were glued to a panel of size 128×128 cm. One of those panels has been lost (Fig. 3.8). The other one is in the collection Etzold at Museum Abteiberg in Mönchengladbach, Germany.

Besides the experience of solving a non-trivial problem in information aesthetics by a program that required heuristics to work, I did this project more like a scientist

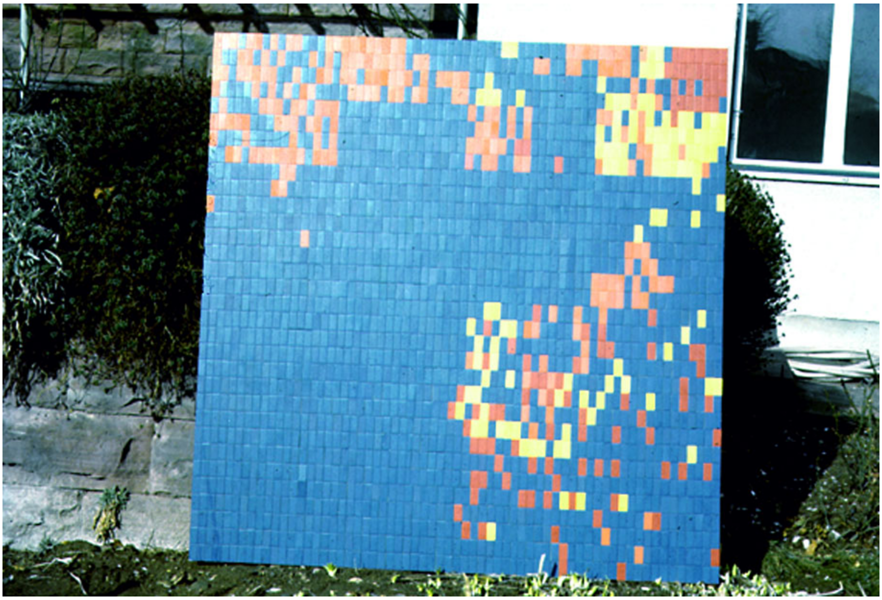


Fig. 3.8 Frieder Nake: *Generative Aesthetics I, experiment 4a.1*, 1969

than an artist. An artist would have organised, well in advance, a production site to transform the large set of the generated raster images into a collection of works. This collection would become the stock of an exhibition at an attractive gallery. A catalogue would have been prepared with the images, along with theoretical and biographical essays. Such an effort to propagate the most advanced and radically rational generative aesthetics would have been worthwhile.

Instead, I think I am justified in concluding that this kind of formally defined generative aesthetics did not work. After all, my experiments with *Generative Aesthetics I* seemed to constitute an empirical proof of this.

Was I premature in drawing the conclusion? It was the time of *Cybernetic Serendipity* in London, *Tendencies 4*, and later *Tendencies 5* in Zagreb. In Europe one could feel some low level, but increasing attention being paid to computer art. A special show was in preparation for the 35th Biennale in Venice, bringing together Russian constructivists, Swiss concrete artists, international computer artists, and kids playing. Wasn't this an indication of computer art being recognised and accepted as art. Premature resignation? Creativity not recognised?

I am not so sure any more. As a testbed for series of controlled experiments on the information-aesthetic measures suggested by other researchers, *Generative Aesthetics I* may, after all, have possessed a potential that was not really fathomed. The number of experiments was too small. They were not designed systematically. Results were not analysed carefully enough. And other researchers had not been invited to use the testbed and run their own, most likely very different, experiments.

It may well be the case that the project should be taken up again, now under more favourable conditions, and different awareness for *generative design*.

3.5 The Fourth and Last Narration: On Creativity

This chapter finds its origins in a Dagstuhl Seminar in the summer of 2009. Schloss Dagstuhl is a beautiful location hidden in the Southwest of Germany, in the province of Saarland. Saarland is one of the European areas where over centuries people from different nations have mixed. After World War II, Saarland belonged to France for some time until a public vote was taken (in 1955) about where people preferred to live, in West Germany or France. Was their majority decision in favour of the German side an act of collective creativity?

Mathematicians in Germany and beyond have had a wonderful institution ever since 1944, the *Mathematical Research Institute of Oberwolfach*. It is located at Oberwolfach in the Black Forest. Mathematicians known internationally for their interest in a specialised field, meet there to pursue their work. They come in international groups, with an open agenda leaving lots of time for spontaneous arrangements of discussion, group work, and presentations.

The German *Gesellschaft für Informatik*, after having established itself as a powerful, active, and growing scientific association in the field of computing, became envious of the mathematicians and decided that they also wanted to have such a well-kept, challenging and inviting site for scientific meetings of high quality. Soon enough, they succeeded. Was this creativity or organisation?

So Dagstuhl became a place for scientists and others, from computer science and neighbouring disciplines, to gather in a beautiful environment and work on issues of a specialised nature. They are supposed to come up with findings that should advance theory and practice of information technology in the broadest sense.

A week at a Dagstuhl seminar is a great chance to engage in something that we usually find no opportunity to do. The topic at this particular occasion was computational creativity—a topic of growing, if only vague interest these days.

Inspired by some of the debates at the seminar, I have tried in this chapter, to recall a few aspects from the early history of algorithmic art as a case from the fringes of computing that we would usually consider a case for creativity. We usually assume that for art to emerge, creativity must happen. So if we see any reason to do research into the relation between creativity and computers, a study of computer art seems to be a promising case.

People are, of course, curious to learn about human creativity in general. A special interest in the impact of computing on creativity must have its roots in the huge machine. As already indicated, I see the computer as a semiotic machine. The subject matter of computational processes must always already belong to the field of semiotics. The subject matter computers work on is of a relational character more than it is "thing-like".

This important characteristic of all computing processes exactly establishes a parallel between computable processes and aesthetic processes. But to the extent that computable processes are carried out by machinery, those processes cannot really reach the pragmatic level of semiosis. Pragmatics is central to purpose. Purpose is what guides humans in their activities. The category of purpose is strongly connected to interest.

I don't think it could be proved—in a rigorous mathematical meaning of the word "prove"—that machines do not (and can never) possess any form of interest and, therefore, cannot follow a purpose. On the other hand, however, I cannot see any common ground between the survival instinct governing us as human beings, and the endless repetition of always the same, if complex, operations the machine is so great and unique at. There is just nothing in the world that indicates the slightest trace of an interest on behalf of the machine. Even without such proof, I do not see any reason or situation where I would use a machine, and this machine developed anything I would be prepared to accept as "interest" and, in consequence, a purposeful activity.

What above I have called an interpretation by the machine is, of course, an interpretation only in a purely formal sense of the word. Clearly, the agent of such interpretation is a machine. As a machine, it is constructed in such a way that it has no freedom of interpretation. The machine's interpretation is, in fact, of the character of a determination: it must determine the meaning of a statement in an operational way. When it does so, it must follow strict procedures hard-wired into it (even if it is a program called a *compiler* that carries out the process of determination). This does not allow a comparison to human interpretation.

3.6 Conclusion

The conclusion of this chapter is utterly simple. Like any other tool, material, or media, computer equipment may play important roles in creative processes. A human's creativity can be enhanced, triggered, or encouraged in many ways. But there is nothing really exciting about such a fact other than that it is rather new, it is extremely exciting, it opens up huge options, and it may trigger super-surprise.

In the year 1747, Julien Offray de La Mettrie published in Leiden, the Netherlands, a short philosophical treatise under the title *L'Homme Machine* (The Human Machine).[18] This is about forty years before the French Revolution, in the time of the Enlightenment. La Mettrie is in trouble because of other provocations he published. His books are burned, and he is living in exile.

In *L'Homme Machine*, La Mettrie undertakes for the first time the radical attempt to reduce the higher human functions to bodily roots, even to simple mechanical explanations. This essay cannot be the place to contribute to the ongoing and, perhaps, never ending discourse about the machinic component in humans. It has been demonstrated often enough that we may describe certain features of human

[18] I only have a German edition. The text can easily be found in libraries.

behaviour in terms of machines. Although this is helpful at times, I do not see any reason to set both equal.

We all seem to have some sort of experienced understanding of construction and intuition. When working and teaching at the Bauhaus, Paul Klee observed and noted that "We construct and construct, but intuition still remains a good thing."[19] We may see construction as that kind of human activity where we are pretty sure of the next steps and procedures. Intuition may be a name for an aspect of human activity about which we are not so sure.

Construction, we may be inclined to say, can systematically be controlled; intuition, in comparison, emerges and happens in uncontrolled ways. Construction stands for the systematic aspects of work we do; intuition for the immediate, non-considerate, and spontaneous. Both are important and necessary for creation. If Paul Klee saw the two in negative opposition to each other, he was making a valid point, but from our present perspective, he was slightly wrong. Construction and intuition constitute the dialectics of creation. Whatever the unknown may be that we call intuition, the computer's part in a creative process can only be in the realm of construction. In the intuitive capacities of our work, we are left alone. There we seem to be at home. When we follow intuitive modes of acting, we stay with ourselves, implicit, we do not leave for the other, the explicit.

So at the end of this mental journey through the algorithmic revolution (Peter Weibel's term) in the arts, the dialectic nature of everything we do re-assures itself. If there is anything like an intuitively secure feeling, it is romantic. It seems essential for creativity.

In the first narration, I presented the dense moment in Stuttgart on the 5th of February, 1965, when computer art was shown publicly for the first time. If you tell me explicitly, Georg Nees told the artist who had asked him—if you tell me explicitly *how* you paint, then I can write a program that does it. This answer concentrated in a nutshell, I believe, the entire relation between computers, humans, and creativity.

The moment an artist accepts the effort of describing how he works, he reduces his way of working to that description. He strips it of its embedding into a living body and being. The description will no longer be what the artist does, and how he does it. It will take on its separate, objectified existence. We should assume it is a good description, a description of such high quality concerning its purpose that no other artist has so far been able to give. It will take a lot of programming and algorithmic skill before a program is finished that implements the artist's rendition. Nevertheless, the implementation will not be what the artist really does, and how he does it. It will, by necessity, be only an approximation.

He will continue to work, he will go on living his life, things will change, he will change. And even if they hire him as a permanent consultant for the job of his own de-materialisation and mechanisation, there is no escape from the gap between

[19](Klee 1928) Another translation into English is: "We construct and construct, but intuition is still a good thing."

a human's life and a machine's simulation of it. Computers just don't have bodies. Hubert Dreyfus (1967) has told us long ago why this is an absolute boundary between us and them.

The change in attitude that an artist must adapt to if he or she is using algorithms and semiotic machines for his or her art is dramatic. It is much more than the cozy word of "it is only a tool like a brush" suggests. It is characterised by explicitness, computability, distance, decontextualising, semioticity. None of these changes is by itself negative. To the contrary, the artist gains many potentials. His creative capacities take on a new orientation exactly because he or she is using algorithms. That's all. The machine is important in this. But it is not creative.

The creation of a work that may become a work of art may be seen as changing the state of some material in such a way that an idea or intent takes on shape. The material sets its resistance against the artist's will to form. Creativity in the artistic domain is, therefore, determined by overcoming or breaking the material's resistance. If this is accepted, the question arises what, in the case of algorithmic art, takes on the role of resistant material. This resistant material is clearly the algorithm. It needs to be formed such that it is then ready to perform in the way the artist wants it to do. So far is this material removed from what we usually accept under the category of form, that it must be built up to its suitable form rather than allow for something to be taken away. But the situation is similar to writing a text, composing a piece of music, painting a canvas. The canvas, in our case, turns out to be the operating system, and the supporting program libraries appear as the paints.

Acknowledgements My thanks go to the people who have worked with me on the compArt project on early digital art and to the Rudolf Augstein Stiftung who have supported this work generously. I have never had such wonderful and careful editors as Jon McCormack and Mark d'Inverno. They have turned my sort of English into a form that permits reading. I also received comments and suggestions of top quality by the anonymous reviewers. All this has made work on this chapter a great and enjoyable experience.

References

Anon (1965). Bald krumme linien. In *Der Spiegel* (xxxx, pp. 151–152).

Bense, M. (1965). *Aesthetica. Einführung in die neue Ästhetik*. Baden-Baden: Agis. This is a collated edition of four books on aesthetics that appeared between 1954 and 1960. Aesthetica has been translated into French and some other languages.

Birkhoff, G. (1931). A mathematical approach to aesthetics. In *Collected mathematical papers* (Vol. 3, pp. 320–333). New York: Am. Math. Soc.

Cohen, H. (2007). Forty-five years later.... http://www.sandiego.gov/public-library/pdf/cohencatalogessay.pdf.

Dreyfus, H. (1967). Why computers must have bodies in order to be intelligent. *The Review of Metaphysics*, *21*, 13–32.

Duchamp, M. (1959). The creative act. In R. Lebel (Ed.), *Marcel Duchamp* (pp. 77–78). New York: Paragraphic Books.

Frank, H. (1964). *Kybernetische Analysen subjektiver Sachverhalte*. Quickborn: Schnelle.

Gerstner, K. (1963). *Programme entwerfen*. Teufen: Arthur Niggli. Second ed. 1968, third ed. 2007 in English under the title *Designing programmes*. Baden: Lars Müller.

Glowski, J. M. (Ed.) (2006). *Charles A. Csuri: beyond boundaries, 1963-present*. Columbus: Ohio State University.
Guilford, J. P. (1950). Creativity. *American Psychologist*, *5*, 444–454.
Gunzenhäuser, R. (1962). *Ästhetisches Maß und ästhetische Information*, Quickborn: Schnelle.
Hentig, H. v. (1998). *Kreativität. Hohe Erwartungen an einen schwachen Begriff*. München: Carl Hanser.
Herzogenrath, W., & Nierhoff, B. (Eds.) (2006). *Vera Molnar. Monotonie, symétrie, surprise*. Bremen: Kunsthalle. German and English.
Herzogenrath, W., Nierhoff, B., & Lähnemann, I. (Eds.) (2007). *Manfred Mohr. Broken symmetry*. Bremen: Kunsthalle. German and English.
Hollinger, L. (Ed.) (1999). *Vera Molnar. Inventar 1946–1999*. Ladenburg: Preysing Verlag.
Keiner, M., Kurtz, T., & Nadin, M. (Eds.) (1994). *Manfred Mohr*. Weiningen-Zürich: Waser Verlag. German and English.
Kittler, F. (1985). *Aufschreibesysteme 1800/1900*. München: Fink. English: Kittler, F. (1990). *Discourse networks 1800/1900*. Stanford, with a foreword by David E. Wellbery.
Klee, P. (1928). Exakte versuche im bereich der kunst.
Knuth, D. E. (1968). *The art of computer programming*. Reading: Addison-Wesley. Planned for seven volumes of which three appeared from 1968 to 1973. Resumed publication with part of Vol. 4 in 2005.
Lunenfeld, P. (1999). *The digital dialectic. New essays on new media*. Cambridge: MIT Press.
McCorduck, P. (1990). *AARON's code: meta-art, artificial intelligence, and the work of Harold Cohen*. New York: Freeman.
Moles, A. A. (1968). *Information theory and esthetic perception*. Urbana: University of Illinois Press. French original 1958.
Nadin, M. (2011). *Semiotic machine*. An entry of the Semiotics Encyclopedia Online. http://www.semioticon.com/seo/S/semiotic_machine.html.
Nake, F. (1974). *Ästhetik als Informationsverarbeitung*. Vienna: Springer.
Nake, F. (2009). The semiotic engine. Notes on the history of algorithmic images in Europe. *Art Journal*, *68*, 76–89.
Nees, G., & Bense, M. (1965). *Computer-grafik* (19th ed.) Stuttgart: Walther.
Nöth, W. (2002). Semiotic machines. *Cybernetics & Human Knowing*, *9*, 5–21.
Shannon, C. E., & Weaver, W. (1963). *The mathematical theory of communication*. Chicago: University of Illinois Press.
Stern, W. (1912). *The psychological methods of intelligence testing*. Baltimore: Warwick and York. Transl. from the German.
Sundin, B. (Ed.) (1980). *Is the computer a tool?* Stockholm: Almqvist & Wiksell.

Chapter 4
Evaluation of Creative Aesthetics

Harold Cohen, Frieder Nake, David C. Brown, Paul Brown, Philip Galanter, Jon McCormack, and Mark d'Inverno

Abstract This chapter is an edited conversation on the topic of computational evaluation of artistic artefacts. The participants were Harold Cohen, Frieder Nake, David Brown, Jon McCormack, Paul Brown and Philip Galanter. It began at the Dagstuhl seminar on computers and creativity, held in Germany in 2009 and continued over a period of several months via email. The participants discuss their views on the prospects for computational evaluation of both the artistic process and the made artefact.

H. Cohen (✉)
University of California, San Diego, CA, USA
e-mail: hcohen@ucsd.edu

F. Nake
University of Bremen, Bremen, Germany
e-mail: nake@informatik.uni-bremen.de

D.C. Brown
AI Research Group, Computer Science Department, Worcester Polytechnic Institute, Worcester, MA, USA
e-mail: dcb@cs.wpi.edu

P. Brown
Informatics, University of Sussex, Brighton, BN1 9RH, UK
e-mail: paul@paul-brown.com

P. Galanter
Department of Visualization, Texas A&M University, College Station, Texas, USA
e-mail: galanter@viz.tamu.edu

J. McCormack
Centre for Electronic Media Art, Monash University, Caulfield East, Victoria 3145, Australia
e-mail: Jon.McCormack@monash.edu

M. d'Inverno
Department of Computing, Goldsmiths, University of London, London, UK
e-mail: dinverno@gold.ac.uk

J. McCormack, M. d'Inverno (eds.), *Computers and Creativity*,
DOI 10.1007/978-3-642-31727-9_4,

4.1 Introduction

This chapter documents a conversation on the prospects for computational evaluation of art, aesthetics and creativity. The dialogue began in July 2009 at the Dagstuhl seminar (Boden et al. 2009). At the seminar a small group of participants decided to explore the problem of evaluation of creative works. Here "evaluation" included the decision process during artwork production and that following an artwork's completion, including evaluation by others such as audiences and critics.

Following the seminar, the dialogue continued via email over a period of many months (from July to November, 2009) and covered a variety of fascinating issues. What follows here is an edited version of this correspondence, in chronological order. As the reader will understand, the dialogue is relaxed and conversational—points are not always justified, and the unplanned and improvisational nature of the conversation reveals different ideas than would be found in a formal authored chapter. But we hope readers will appreciate the sincerity and openness of all the contributions, the value of candid personal opinions, and the shared sense of trying to explore the complexity of the issues raised.

There is much here that philosophers may be familiar with (and possibly even aghast at). But it does serve as an important historical record, particularly from the perspective of a number of pioneering artists who have been working in this area for decades. Their wisdom and experience brings a compelling perspective to the conversation. The collective insight of these pioneers provides an important point of reference for the next generation of researchers and artists entering the field.

4.2 Background: Evaluation of Artistic Artefacts

Before presenting the edited dialogue, a short background is first provided, in order to establish the context from which these discussions began.[1] The discussion is centred around the idea of computational evaluation of creative artistic artefacts. There are a number of points to be made to flesh out this idea. Firstly, how something is evaluated depends on the evaluator's perspective and role. This role may be as creator or designer, viewer, experiencer, or interactive participant.

This leads to some initial questions:

- What are the main features of human creative and aesthetic evaluation?
- How do these features (and the methods that are used) change according to the evaluator's role in the process?
- What aspects of evaluation can be made computational?
- Is it necessary for computational evaluation to mimic the evaluation methods of humans?
- Does it make sense to automate a task that is so especially human?

[1] Elements of this section are based on the initial Dagstuhl group discussions (Boden et al. 2009).

Answers to these questions have many complex implications, leading to a myriad of further questions, several of which arise in the dialogue that follows. When reading this dialogue it is important to keep in mind that the *role* evaluation plays determines the *kind* of evaluation required. Evaluation of a work as it proceeds leads to changes in that work (and potentially to future works), creating a feedback loop between action, intent, material (physical, musical, virtual) and decision.

Perhaps the most basic evaluation of a work as it proceeds is to know when it is finished. Knowing when a work is "done" arises for artists in almost any medium, working alone or in collaboration. At the opposite end—when beginning a work of art—initial ideas, conditions, moods and decisions can have major impacts on what follows. Cezanne reportedly threw away paintings once an "incorrect" brushstroke was made. Improvising musicians do not have such luxuries: a pianist like Keith Jarrett is acutely aware that what he first plays will shape the rest of the performance.

Computers have the ability to "undo", backtrack, and trial many possible combinations very quickly. But knowing what to undo, how and when to backtrack, and which paths to pursue or abandon requires evaluation *appropriate to the task* if it is to be successful. Evaluation of a work as it proceeds is generally concerned with decision making and prediction, e.g. what are the implications of making this mark, playing this sequence of sounds, or using media in a specific way? Accomplished artists have a seemingly innate intuition about creative decision-making and its impact, developed and fine-tuned over many years of practice. But can such decisions alone lead to the transformational creativity (Boden 1991) we see in the best human artists?

Evaluation of a finished artwork as an "art object" presents a different set of criteria. This may include examination of the emotional response of people experiencing the work under consideration, or an evaluation based on (for example) some aesthetic principles. In this book we are inevitably interested in what aspects of evaluation might be captured in a computational system. One possibility is to employ machine learning techniques where the system is trained on existing art works in order to learn any underlying aesthetic criteria. Certainly, this forms the basis of much current research.

It is also crucial to understand what we are evaluating for: *quality* (artistic, conceptual, aesthetic), *value* (monetary, cultural, critical, emotional), or something else? An important distinction can be made between the evaluation of creativity (appropriate or valuable novelty) and, for example, aesthetics. Something that is aesthetically pleasing may not necessarily be creative (as evidenced by looking at any collection of picturesque wall calendars, for example).

Human evaluation of artistic works typically extends well beyond the artefact itself, encompassing implicit knowledge and cultural norms, such as the intention of the artist who created it, the situation and conditions—social, political, and cultural—under which it is made and presented, the observer's knowledge and experience of similar works, and the dominant social values and norms of the day.[2]

[2]For important considerations of these issues, we refer the reader to the contributions in Part III.

As discussed elsewhere in this book (e.g. Chap. 5), becoming an expert or virtuoso in a particular medium normally takes many years of intense practice and immersion. As expertise and virtuosity mature, so does evaluation: the two appear to go hand-in-hand. Knowledge and experience emerge as decisive factors in producing artefacts of high creative value.

With these statements and questions forming a background, let us now proceed to the discussion.

4.3 A Conversation on Evaluation

The participants are (in order of appearance and identified by their initials), Harold Cohen (HC), Frieder Nake (FN), David Brown (DB), Jon McCormack (JM), Paul Brown (PB) and Philip Galanter (PG).

The conversation begins with a discussion about the aesthetic evaluation of art by people and computers.

Harold Cohen (HC): I sometimes wonder whether Western culture hasn't generated more art evaluation than art over the past few hundred years. How much of it is known outside the art world is another matter. It is worthwhile to make clear that aesthetic evaluation has little to do with conformance to the set of "rules" still being widely taught in art colleges.

As to the evaluation of aesthetics computationally, I confess to paying little attention to what's going on outside my studio, but I'd be very surprised to learn that there's a rich enough history of practical work to fill a book. Why is there so little history? To begin with, AI is still not at a stage where a program could accumulate enough relevant knowledge about an object it didn't make itself to make a non-trivial evaluation, so the discourse is limited, necessarily, to art-making programs, of which there have been relatively few. (I'm unclear about whether the same limitation would apply in other forms: music, for example.)

All of my own sporadic forays until now have been non-starters. But once I relinquish the notion of program autonomy and accept that the program is working with and for me, it becomes clear that it is capable of exercising (my) aesthetic judgement. And it does, to a point. But it's exercised on the work-in-progress, not on the finished work. Thus, it doesn't wait to tell me that an image has too much grey; it evaluates and corrects as it proceeds, provided that I can tell it how much grey is enough. That's a trivial example; one step up I'd need to say how much grey is enough relative to something else. Even if I could find a way of identifying amount-of-grey as an evaluation issue, and say what to do about such issues generally, there is still the problem that they are a moving target. That target moves in the human domain.

Unfortunately, it's a lot easier to say what's wrong with an image than to say what makes it special. I'm looking for the images that are *transcendent;* which means, by definition, that I don't know what it is that makes them special and don't know how to describe what it is in computational (or any other) terms. The limitation is my own, not the program's.

Evaluation of a work in progress is directed to how to proceed. Evaluation of a finished work is directed to whether it's any good. The procedures required to satisfy the two are likely to be quite different, even when the same aesthetic is informing the procedure in each case.

I think it's very unlikely that "in-line" evaluation[3] can be done algorithmically. The simplest case I can think of would be to determine whether the work is finished. Even that is much harder than one might think. It could only be done algorithmically if one could provide an evaluation function—highly unlikely—which would be, in any case, a shifting target with respect to the many different goals a program (or human artist) may have.

For the general case the problem is much more difficult. The program can't determine how to proceed unless it knows what it has done, and knowing what it has done—the object so far—involves the notoriously troublesome "new term" problem. It knows it has done a, b and c, but can't know that it has introduced a novel and unanticipated relationship between a and c. Which is exactly what should be the determinant to the next step.

It is true, of course, that many human artists proceed algorithmically—you do this, then you do that, and after you've done all the thises and thats you have an artwork. No evaluation is required; your job is simply to do all the steps well. In human terms this algorithmic approach results in what we call "academic art", which I think has no place in a discussion on creativity.

Post-hoc evaluation is no less troublesome, and I suspect it's likely to be impossible for a program that didn't itself make the artwork. A formal colour evaluation[4] that doesn't take account of the possibility that all the well-balanced colour harmonies may add up to a portrait of an oddly-dressed man making rude hand gestures, for example. It also implies that there are canons for colour distribution and the rest, and evaluation simply measures conformance to those canons. (Impressionism good, German Expressionism bad?)

For most artists, making art is an on-going affair, not a series of isolated (artwork) events. Consequently, the completion of each work provides an extension of the feedback-driven consideration operating in the in-line evaluation. It is similarly concerned with direction, not aesthetics (except to the degree that, for the completed work, the artist must decide at least in part on aesthetic grounds, whether to accept it or reject it).

That is quite different from the aesthetic evaluation by any other agent, who is not engaged in that on-going process. In this case direction is clearly not an issue, acceptance/rejection is not an issue, and the aesthetic principles brought to bear on the work are unlikely to have much correspondence to those of the artist.

---

[3] By "in-line" Cohen is referring to evaluation of aesthetic decisions as a work proceeds.

[4] At the time of writing this statement, Cohen was very focused on translating his theories of colour and colour harmony into algorithms that AARON could use to colour abstract shapes.

I have some hope for the possibility of post-hoc evaluation by the generating program; no hope at all for evaluation by any other program.

Frieder Nake (FN): Aesthetics is, to a large extent, an evaluative discipline. We would probably not immediately equate evaluation with judgement. But the two are related. "Evaluation" is, quite likely, a more technical approach to aesthetic (or any other) judgement. However, we should be aware of the fundamental difference between value and measure. The temperature in a room can be measured because an instrument has been constructed that shows what physicists have defined as a quantitative expression of the quality of "warmth". The measured temperature is objective insofar as it has nothing to do with any human being present and experiencing the room in the actual situation and context. The human's value may be expressed as hot, warm, cool, or whatever else. Notice these are qualities.

So, in a first approximation, we may relate value with quality (human, subjective), and measure with quantity (instrument, objective).

The value judgement by a human may be influenced by the measured data delivered by an instrument. But the two are definitely and importantly to be kept apart (for intellectual rigour). Even more so in the complex situation of aesthetics.

Aesthetics itself is considered by many as being about our sensual perception of things, processes, and events in the environment. Hence, the subject matter of aesthetics is in itself intrinsically subjective. Those who start from this position cannot accept the claim that there are objective measures that would substantially contribute to human judgement.

HC: However, there have been times when number systems have had special cultural significance, and consequently aesthetics has been bound up with objective measures. For example, the Greek canon of human proportion was quite clear about how big the head should be in relation to the body, and I'm reasonably sure the sculpture critic would have regarded conformity to, or departure from, that canon as an aesthetic issue. There are many other examples.

Objective measures are a component of aesthetics when the measures themselves are important culturally. Today we have no such measures, and attempts to find them in contemporary artworks seem absurd to me, just as Ghikas's[5] attempts to find the golden mean in the art of a culture that knew nothing about incommensurable numbers seems absurd.

FN: Harold, you are absolutely right. By reminding me of some facts of history, you make me aware of a psychological hang-up that I now believe I have created in a dogmatic reaction against Max Bense.[6]

Bense, of course, allowed only objective aesthetic measures. He did so in reaction to German fascism where emotion was the only goal of their grandiose aesthetics

[5]Nikos Hadjikyriakos-Ghikas, a 20th-century Greek artist and academic.

[6]Max Bense was an influential German philosopher and Nake's teacher and mentor in his formative years as an artist exploring the generative possibilities of the computer in the 1960s.

for (against?) the masses. Bense was, at the same time, clear about subjective elements in the building of an aesthetic judgement. But that was outside of scientific investigation and research. It was purely private.

As a young man, I liked and loved this entire attitude. Everything in the world would be rational, mathematical, objective. Everything else was absolutely without interest.

I later adopted the view of aesthetics and sensual perception being tied together. From there it is a short step to my position. Your beautiful hints to some other times carry exactly the message that you summarise above. If some rule or law or proportion or other statement is culturally important, ruling, governing, then—of course—the individual sensual perception is, as always, largely determined by that "objectively" (i.e. culturally) dominating fact.

Having responded to Harold, Frieder now returns to his original discussion on developing algorithms for evaluation of aesthetics.

We seek algorithmic methods of evaluation that might have bearings on individual subjective aesthetic judgement. Yes—some researchers or even critics and artists want to find such measures, to define them, to construct instruments that tell us numbers on a scale. If we want to do this, if we neglect the deeply subjective character of a value judgement, we will try and find or define such measures to replace (or at least approximate) the subjective value. I am afraid, such heroic attempts will not get them very far.

It might be necessary to recall G.D. Birkhoff's definitions of aesthetic measure in the 1920s and 1930s. A lot of psychological work was done afterwards (in the form of empirical measures) with the unceasing intention of scientists to explain complex phenomena in an objective way.

The Birkhoff case is telling. He took up the old and popular idea of "order in complexity" or "unit in complexity" (a clearly subjective value). He put in a formula: $M = O/C$ (to me, this looks beautiful!). Here M is the aesthetic measure, O is the measure for order, C is the measure for complexity.

See how this works? You translate the words with all their connotations into variables. The variables stand for numbers, measured in appropriate units according to a measuring schema. What was a subjective interpretation all of a sudden has become reading scales. Great!

All that is left to do after this bold step is to "appropriately define" the measuring procedure. When you read Birkhoff's examples, you will be appalled. I was not, when I was young and did this (in the early 1960s). Birkhoff, as his famous example, chose polygons as the class of shapes to measure aesthetically. Complexity was for example the number of edges of the closed polygon. Order was, by and large, the degree of symmetry (plus a few additional features). The square is the best.[7] Wonderful!

When in those days, as a young guy using computers for production of aesthetic objects, I told people, small crowds perhaps, about this great measuring business,

[7]By Birkhoff's formula, the square evaluates to the polygon with the highest aesthetic value.

someone in the audience always reacted by indicating: "young man, what a hapless attempt to put into numbers a complex phenomenon that requires a living and experienced human being to judge".

My reaction then was, oh yes, I see the difficulties, but that's exactly what we *must* do! And we will, I threatened them. I guess, looking back without anger, they shut up and sat down and thought to themselves, let him have his stupid idea. Soon enough he will realise how in vain the attempt is.

He did realise, I am afraid to say.

In the early 1960s, Birkhoff's quotient of order over complexity was taken up again (by Bense, Frank, Gunzenhäuser, Moles, myself). It was given a promising interpretation in information theoretic terms. Helmar Frank, in his PhD thesis of 1959, defined measures of surprise and of conspicuousness (of a sign, like a colour, in an image). All these attempts were bold, strong, promising, radical. But they were really only heroic: the hero always dares a lot, more than anyone else, stupidly much, and always gets defeated and destroyed in the end.

I am sceptical about computer evaluations of aesthetics for many reasons. They are a nice exercise for young people who believe in one-sidedness. Human values are different from instrument measures. When we judge, we are always in a fundamental situation of forces contradicting each other. We should not see this fact as negative. It is part of the human condition.

Harold may be the one who, from his forty years of computational art practice that took him so close to the heroes of AI, would be able to pave the way. But even he is sceptical. "I don't know what it is that makes them (the computer-generated images coming from his program) special", he says. He continues to say he doesn't know how to describe "what it is in computational terms".

If we ever wanted to apply algorithmic methods to aesthetic evaluations, we must first be able to describe what we want to measure. Such a description must be formal and computable. So an explicitly formalised and algorithmic description is what would be needed. And those descriptions would be of works that we are used to calling "art". We all know the situation where five of us are around a small collection of pictures. We discuss them. We describe, bring in comparisons, develop our judgements against the background of our own lives, and of the current situation and discussion. We come up with a judgement in the end that doesn't totally satisfy any participant of the meeting. But all of us feel quite okay. We think we can justify the judgement. Tomorrow it could easily turn out to be different. This is how complex the situation of an evaluation is.

In Toronto in 1968/69, I wrote a program that I proudly called *Generative Aesthetics I*. It accepted as input a series of intervals for information aesthetic measures. They defined boundary conditions that must not be violated. The algorithm then tried to find a solution maximising the aesthetic measure against the boundary conditions. Its result was, of course, only a (probability based) distribution of the colours.

Just see what that program's task was: given a set of numeric (!) criteria, determine a "best" work that satisfies certain given evaluations. Isn't that great? I thought it was. And I was 29 years old.

A second program took this statistical description of an image (really: an infinity of images) and distributed colours into a quadtree structure such that the prescribed (just calculated) frequencies of colours were obeyed. I called the quadtree structure "the topology of the image".

I guess it was one of the most powerful programs ever in computer art, and certainly of its early phase. The program showed how little you achieve this way. As Harold says, you can use such dynamic evaluative measures during the generative process. That's all. Anything beyond this is human value judgement.

Phil Galanter has shared some of the scepticism of others, but says giant leaps are not to be expected. But baby-steps should be tried just to see where they get the baby. Yes, dear Phil, what is there left to do other than doing baby steps. So let us get into those pink knitted tiny shoes that mothers like to put their baby's feet into and move on from there.

David Brown (DB): I think that an analysis of existing methods in order to influence the output of computational systems—via some embedded knowledge (such as rules)—*is* a useful thing to do.

My experience in the design world suggests that you'll find a lot of people who had "techniques looking for a problem"—i.e. the method of evaluation is shaped by their tool.

I think it is better to analyse the problem and then look for techniques. For example, what kinds of evaluations affecting creativity are made during synthesis and what kinds of techniques can make these evaluations? Additionally, what kinds of evaluations can be applied to the descriptions of resulting artefacts, always assuming that all necessary sensing is in place.

For creative evaluation, newness and surprise are key to people judging something as being creative. But judging both of these computationally is tricky, especially during synthesis.

Focusing on learning is putting the cart before the horse. Focusing on a belief that something is "impossible" is not letting either out of the stable: a great way to reduce discovery of, and understanding about, the ingredients that lead to creative artefacts. By taking each challenge and looking at how it might be tackled we can make systematic progress.

Can we get a system to figure out that a blue widget isn't much different from a green widget, even if in some sense it is "new"? How can different types of newness be evaluated? Can a system predict how much a "newer" choice during synthesis will affect the judgement of the creativity of the finished product?

We take questions such as these and look for techniques that might help. For example, could we use the web for assessing newness? Could we take a representation of an artefact that has structural, behavioural and functional components and use that to decide a degrees of newness? Could fuzzy matching techniques be used to detect similarity and therefore newness? And so on...

Jon McCormack (JM): This discussion has made a number of claims as to why objective aesthetic measures seem impossible for an individual or machine. Nevertheless, I do think there is some basis for looking at aesthetic commonality particular to a specific culture, social group, style or individual. After all, what is taught at art schools? Students learn the basic craft of their medium, they are exposed to many exemplars, they try and fail, try again, receive critique and feedback with a hope of improving with experience. But as has been pointed out by Harold, rule following isn't enough, art is an ongoing dialogue.

A lot of generative art software encodes specific forms of aesthetic judgement. The artist/programmer carefully chooses specific rules so as to create a system that generates pleasing aesthetics for *them* (which in turn may change after being exposed to computer aesthetics or even the aesthetics of the artwork-in-progress). Therefore, in a sense, this software is "evaluating" what it is doing (as it is doing it), but not in the way that a human does. It is an evaluation done for aesthetic purposes. However, the judgement originates with the programmer, not the program, so it becomes a continuous scale of how much is imbued to each.

A program that can adapt can learn, and hence change its judgement. That we know to be possible (using evolutionary algorithms or machine learning for example), but as Frieder points out, the baby may never get out of its tiny pink shoes. Perhaps we need to wait until machines have their own social evolution.

Frieder also raises the point that aesthetics is tied to the phenomenology of sensual perception—how else could we appreciate work like that of the artist James Turrell for example? It is difficult to imagine a machine experiencing such a work and coming to a similar aesthetic understanding, unless that machine had very similar phenomenological perception to us, or had sufficient knowledge about us (our perception, cognition, experience) and physics, to infer what our understanding would be. The same provisos apply to a machine originating such a work.

But while there may be many areas of human aesthetics, cognition and perception that are currently "off limits" to machines, it does not necessarily preclude machines that may be able to originate something that humans find aesthetically valuable. Indeed, a lot of "computer art" has given us very new aesthetics to contemplate.

Paul Brown (PB): I am very aware that writing too briefly opens up the opportunity for misunderstanding (I suspect Darwin said this?). But, to try:

One of the major themes in human development has been the revealing of structure (logic) through the observation and analysis of phenomena. Let me suggest that this point of view, in it's extreme perhaps, believes that all phenomena can be explained in some rational manner. In the history of art this complements the "classical" roots of art and leads directly to the work of Peirce, Saussure, Cezanne, Seurat, etc., and then into the 20th century experiments in constructivism, rational aesthetics, analytical philosophy, cybernetics, conceptualism, systems art, and so on... We could call this approach Modernist but this term is fraught with misunderstanding, especially as it is so (ab)used within the art world.

Another major theme suggests that understanding comes via entering into a relationship with the phenomena that enables the spontaneous emergence of meaning. We use terms like "intuition" and "inspiration". The extreme of this point of view suggests that critical analysis is unnecessary and may actually be counter-productive (and in theological "controlling" manifestations that it should be suppressed). I know of several artists who, after pursuing PhD "practice-based" research, are now unable to practice since they have lost their spontaneity. Here belief is paramount—the subjective takes precedence over the objective. In the world of art this meme develops in the Romantic tradition. With the same reservations as above we could adopt the term Postmodern to describe this kind of thinking as it developed in the late 20th century.

One important distinctions between these two positions is that the former believes that everything can be explained using rationally/logical methods and the latter does not.

As a member of the former group I believe that the major shortcoming of the latter is that it implicitly invokes the need for a quality of the "unexplainable"—some kind of immaterial "essence" or "soul". However I am also aware that in science we now accept (believe in) dark matter and (even more mysteriously) dark energy—qualities which enable our structural analyses of the universe to make sense but for which we have little or no direct evidence.

Another interesting comment comes from the British biologist/cybernetician Geoff Sommerhoff in his explanation of "freedom of will". He suggests that freedom of will is the response of a simple entity (humans) to an environment (the universe) that seems to be almost infinitely complex. For Sommerhoff freedom of will is no more than a psychological mechanism for helping us maintain our sanity when faced with the actuality of our insignificance and our inability to act independently. Taking this further we can interpret Sommerhoff as suggesting that although everything is knowable, it is not possible for humans to attain all of this knowledge because of our inherent system limitations. This seems to me close to Borges map problem—for a map to be completely accurate it must be—at least—as large (as complex) as the territory it describes. So for us to be able to fully explain the universe we need another universe that is, at least as big, to hold the knowledge.

So for me this objective/subjective question can be expressed:

1. I implicitly believe that everything is rationally explainable (there is no essence or soul);
2. I acknowledge, however, that there are many things that may never be explained;
3. Nevertheless I do not believe that this acknowledgement of limitation should prevent us from seeking explanations—however hard the problems we address may be;
4. I believe that the rational analysis and synthesis of aesthetics (and other perceptual, cognitive, conceptual and creative processes) is one of the key issues for humanity to address in the 21st century—we must now apply our systematic methodologies to our own internal mechanisms (and I'm here using the word "mechanism" deliberately);
5. If we do not then we are in danger of handing our world over to the priests, fascists and other bigots whose only wish is to enslave us.

In response to this on-going discussion, Philip Galanter responds in order to draw out some of the underlying assumptions.

Philip Galanter (PG): In terms of epistemology the (undefended here) subsuming view is that there really are intrinsic unknowns "out there" even though "out there" is a noumenal world that is mechanical, rational and logical. Meaningful, objective and verifiable general explanation is possible. However such explanation is, as a matter of principle, incomplete and statistical. Specific past events may elude explanation, and future events may be unpredictable as a matter of principle even though they are not irrational.

FN: I think I have mentioned before, how much my admired teacher in philosophy, Max Bense, was motivated in all his thinking and writing by his experience as a thinking individual in Nazi Germany.

Nobody should allow him- or herself to let any emotions, anything non-rational creep into their aesthetic (or other) judgement. Rationalism was the weapon in thinking against fascism and other totalitarian movements.

As young students we loved him for these messages. Radically I tried to follow his traces. An exercise that helped me for a long time and occupied my thinking in the most beautiful and satisfying way.

Why then did I later start deviating from this line? And why do I today no longer believe that aesthetic judgement rationalism will get me very far?

It seems to me that, at this moment, I cannot pin down a specific event or insight or influence that caused me to change in the way indicated. In very simple terms, my position is: of course, we try to analyse a painting, a piece of music, a novel, etc. in rationalist concepts and in a rationalist method; such an approach will give us a lot of insight and a way to discuss and criticise without attacking us personally, but only in issues of the subject matter; often, and for many, this is enough and nothing more needs to be done; for others, however, the final judgement remains to be a personal statement based on acquired feelings.

It has happened to me more than once that I enter a gallery room, take a look around, and immediately (and unmediated) react in a positive, excited, interested, attracted way to one of the paintings there. I move closer, study it carefully, think, compare, visit the other paintings in the room, build up a judgement. Often, the immediate impression survives a more careful consideration, and is enforced. Not always though. At times, closer investigation leads to a revision of the first and immediate impression.

I do know that everything I have learned and experienced about Artificial Intelligence, everything I have read from Hubert Dreyfus, Joe Weizenbaum, the Scandinavians, David Noble, from Herbert Simon, Allen Newell, ... all the heroes of AI—all that built up in me, and reinforced again and again, a deep rejection of anything that seems close to the separation of mind and body.

Cartesianism has had a great time, and has led to exciting results. But it has had its time. The belief in "progress" has disappeared from me. Change, yes. Permanent change.

Hannah Arendt refers to Kant as having said that aesthetic judgement relies on examples, not on general concepts. This I believe. I say "believe", not more.

After several weeks of silence, the discussion continues, this time initiated by a report from Harold on his progress with AARON in creating new images for a forthcoming exhibition...

HC: A report from the front. A couple of weeks ago I decided I wanted to see more saturated colour in AARON's output. I gave the program what I thought would be a suitable colour profile (controlling the frequency with which it would choose one of the nine possible combinations of lightness and saturation) and then watched in increasing frustration as the program generated several hundred rotten images.

Yesterday I bowed to what I've always known to be the unyielding dominance of value—lightness—over saturation, and substituted a different colour profile that generated colours from very light to very dark. And this morning I selected forty stunning images: my "aesthetic evaluation"? from more than two hundred mostly excellent images.

What was I looking for when I made the selection?

A sense of place. All the images make use of the same set of form generators; I chose those images that transcended mere arrangement of forms, those that generated the sense that they represented something external to themselves, those that seemed to carry the authenticity of the thing seen.

What contributes to this sense of place?

There are relatively few variables in the program that exercise critical control over the nature and reading of the output. One is the choice of colour profile. Others are the scale of forms relative to the size of the image; the proportions of the image; the background colour (hue, lightness and saturation) relative to what builds in the foreground; the proportion of space allocated to background and foreground; the mode of distribution of the forms.

You'll see that these are all quantifiable. (There are several possible distribution modes, each of which is controlled by quantifiables.)

Is the nature and quality of the output—the sense of place—then quantifiable?

I am aware that there are no intrinsically good or bad values for the variables that control the output. The sense of place—and everything else—results from the combination of all the variable values. That's a multidimensional space with perhaps fifteen or twenty dimensions that I know about; way beyond my own mathematical capabilities if I thought that was a good way to go. But notice that the same set of values generated more than two hundred images, of which I judged only forty to have an adequate sense of place. Evidently there are other elements involved beyond the variable settings; specifically, I suspect, the "clustering" of forms which emerges from distribution and scale and population and all the rest.

Is this emergent property—clustering—quantifiable? I doubt it.

The implication seems to be that a program might be able to pick out the good ones, but couldn't pick out the exceptional ones; which are, of course, the ones I'm interested in. But even this might be going too far, partly because it may not be possible to identify the original variable values from the output, partly because in doing so it would only have identified this particular work as belonging to a particular group and would reject any work that didn't belong to this or another successful group. Clearly, that's not the way to go. The transcendent images that don't belong to any group are precisely the ones I want.

The more important point to make, however, since we appear to be talking about aesthetic evaluation, is that I've not said a word to suggest that beauty is an issue for me. In fact, I don't think I've ever met an artist who did think that beauty was an issue. Beauty is emergent, apparently, from the relentless pursuit of the individual's holy grail, whatever that might be, bearing in mind that my grail and yours are unlikely to have the same shape. That does not necessarily mean that a purely formal evaluation of the work itself, without regard to how it got to be that way—harmony, balance, golden mean and whatnot—are non-starters, but I have yet to see one finish.

And, yes, you certainly do run into cultural issues. Impressionism has been the epitome of "beautiful" painting for a long while now; but the Impressionists were accused of shooting the paint on to the canvas with a pistol. Not good. Though today we'd probably think of that as a great idea; after all, Pollock didn't go in for brushes, either.

FN: I, as one occasional participant in this dialogue, love in particular your comments and deep insight, the insight of a life in art and science, Harold. By necessity our discussion must get closer and closer, as it continues, to the fundamental philosophical question of objective vs. subjective. This discussion would then have to ask what the "thing" would be, what the "work" would be, and much more...

We all know to some extent that these issues cannot be solved (as a mathematical equation may be solved), but that they remain the eternal discourse of philosophy. It produces the question itself in new forms, and therefore also with new answers.

Our question here is, of course, much more pragmatic and mundane. I guess a few statements could be made in this regard. Like, perhaps:

The making of art is subjective. The appreciation of art is subjective. The making of art relies on certain general and specific objective conditions. So does the appreciation.

Humans, as cultural groups or as individuals, like to emphasise how nice it would be to have objectivity. But there is only a little objectivity when humans are involved. There is, however, also little subjectivity if "subjective" is what pertains to this individual, here and now. If the striving for objectivity is taken as an attempt to enter discourses with others (individuals, groups, living or dead), and conduct such discourse with passion and patience, decidedly and forgiving, ready to accept a position, ready to give in and not to win but to convince—if factors like those determine the process then those involved will eventually agree that there is little objectivity, little subjectivity, but lots of historic and societal impact.

Judgement is different from evaluation. The absolute pre-condition for programming (and thus for using computers) is formalisation and computability. This is so even in the most interactive and sensor-prone situation.

The concreteness in your argument, dear Harold, is marvellous, it is telling, it is itself artistic. You know that—if I understand my own thinking well enough—I totally agree with your sentiments. You summarised them beautifully by saying: "at the lowest level of machine evaluation, I can see that the program might be able to tell me which images *not* to print". More, I also think, is not possible. The others say: "we are just at the beginning, give us enough money and time". Birkhoff and all those of the 1930s debate failed. Bense and all those of the 1960s debate (including Nake) failed.

It is perfectly legitimate to use computational methods for some first and preliminary evaluations, as we use the thermometer, the speedometer, the yardstick. When a distance is measured as five meters, some of us say, "oh, I can long-jump this easily". Others will never make it. But all try very hard.

When the temperature in a room is measured as 22 degrees Celsius, some react with "too hot for me", others with "rather cool after a while". Measure, value; evaluation, judgement.

And let us not forget, how you, Harold, continue after your optimistic remark about what the machine might be capable of. You say that you would still take a look before, upon the program's evaluation, you delete the file...

PG: I think that this is the kind of discussion that can always be paused but never ended. For now I'd be happy just to clarify what the differences are.

If it turns out that non-trivial computational aesthetic evaluation is impossible, that in itself would be worth better understanding. It seems to me such a statement might come in two forms. There might be some kind of formal sense, or there might be an engineering analysis leading to absurdly expensive, or quantitatively impossible, practical requirements.

Frieder seems to lean towards the former by saying that aesthetic evaluation would have to be formally computable, but is not. But this leads to (in my mind) an even more interesting question. How is it that the mind is capable of "computing" the uncomputable? Is the mind more than the result of a mechanistic brain?

And if the objection is more practical and in the realm of engineering a similar question is raised. What aspect of the mechanistic brain can we know to be beyond the reach of human engineering? How is it that nature has brought the costs and quantities within reach in a way we will never be able to duplicate?

The strongest objection, to me, would also be the one that claims the least, i.e. that computational evaluation as an engineering challenge is impossible *for the time being*. Maybe it will be within reach in ... 10 years? 50 years? 100 years?

But if the operative objection is this last one it changes the entire conversation. Because then computational aesthetic evaluation is possible in principle and merely contingent. All discussions of creativity should allow for it in principle.

Frieder also mentions that, "Judgement is different from evaluation". In our Dagstuhl discussion Margaret Boden rejected such a notion out of hand. Perhaps they are referring to two different kinds of judgement, or two different kinds of evaluation, or both. In any case this confirms in my mind that the language involved will need more precision than everyday speech, and technical definitions are probably called for. For example, when a human takes a given work of art and merely classifies it to an art movement, can that be called "evaluation" or should some other word be used?

Finally there is a bit of a paradox worth pointing out here. Most attempts to define creativity I heard at the Dagstuhl workshop included a provision that the innovation must not only be new but it must also be of value. Now if computational aesthetic evaluation is more or less impossible does this mean computational creativity is impossible? Or does this mean a computer can be creative without being able to measure the value of its own output?

If so, then turn this back on human creativity. If a creative computer need not "understand" the value of its own creations, does that mean a human can be deemed creative even though they are incapable of knowing whether their creations are valuable?

To me it seems odd to demand that creativity result in value but allow that the creator may not know that it does. It would be similar to crediting someone as being "ethical" even though they cannot discriminate between right and wrong.

My response to these problems is implicit in the chapter I present. I think it will ultimately be more fruitful to disconnect definitions of creativity from questions of value.[8] Just as it's a mistake to connect the definition of art to the definition of good art, I believe it's a mistake to connect the definition of creativity to the definition of valuable creativity.

I see creativity as being more related to issues around complexity and the behaviour of complex systems. For me creativity is simply what complex adaptive systems "do", nothing more and nothing less. From this point of view the value of a given creative act is relative to the (possibly co-evolutionary) situation at hand and the contribution it makes towards adaptation by the creative entity. In this case humans, computers, and all manner of things/processes are capable of some degree of creativity.

PB: Thanks for this good summary of the situation. It seems to me to hit several of the important issues head on. If aesthetic evaluation is uncomputable then how does the mind/brain do it? As you comment, an interesting question in itself. As I briefly mentioned previously, it seems to me that the only way beyond this point is to posit the existence of a metaphysical (super-mechanical) entity which is unacceptable to me. Therefore I assume it has to be computable.

You infer the work of Gödel and Turing and we know that within any finite axiom system there will exist propositions that cannot be resolved. However this doesn't answer the problem since again we must ask: then how does the mind/brain (a finite system) resolve aesthetic evaluation?

I return also to my earlier mention of Sommerhoff's description of freedom of will. He implies that things like creativity and aesthetic evaluation may not be computable until the computing engine is at least as complex (or can reflect the same degree of variety—to use Ross Ashby's term) as the human brain. As suggested in this discussion, this is a long way off.

Nevertheless we have to start somewhere and it seems to me that starting with the assumption that computational aesthetic evaluation is not possible is counter productive—we *must* begin from the belief that it can be achieved.

My glass is half full!

[8] This view is also shared by Dorin and Korb in Chap. 13.

4.4 Conclusion

As you might expect from a topic as complex as computational evaluation of art, there is no real consensus or closure from this discussion, nor could this be realistically expected. Yet it is interesting to examine the different perspectives participants consider to be useful or practical in approaching computational evaluation. As Paul Brown's concluding remarks emphasise, unless you think there is something fundamentally uncomputable and ineffable in what humans do, then computational modelling of human evaluation is at least a possibility. But just because something is possible doesn't make it easy, or even practical. It is tantalising to think that future computational models will shed a different light on evaluation of art (and more generally on human behaviour), complementing and informing other discourses such as critical and cultural theory, or philosophical aesthetics. However, computational models of this kind are still very much in their infancy.

It is also interesting to consider the mirror question to the one that is the main topic of this chapter. Namely, can art made by an individual computer program (or social network of autonomous computer agents) ever be fully understood and evaluated by humans? Such considerations raised in this chapter, and many others running through the entire volume, raise many crucial questions to investigating creativity through computing, a number of which are listed in the final Chap. 16 of this book.

Evaluation remains a difficult and vexed issue for understanding creativity from a computational perspective. No doubt it is something that artists and musicians are involved with at almost every moment of their creative practice, but so far attempts to mimic this process in a machine fall short of what any human being can easily do. Interestingly, the two artists with perhaps the longest experience in this field (Nake and Cohen) see little merit in pursuing the idea of developing creative or aesthetic measures, precisely because they have tried to use them in their own art practices and found them to be creative dead-ends. This should at least give us cause for reflection. While understanding exactly what evaluation is and how it is performed by humans remains an open problem, anyone wanting to make serious inroads into developing machine creativity cannot afford to ignore it.

Acknowledgements We acknowledge the contribution from all the participants, including the original Dagstuhl discussion group on this topic, which consisted of Harold Cohen, Margaret Boden, David Brown, Paul Brown, Oliver Deussen and Philip Galanter. The discussion group notes can be found at http://drops.dagstuhl.de/opus/volltexte/2009/2212/. The interview in this chapter was edited by Jon McCormack.

References

Boden, M. A. (1991). *The creative mind: myths & mechanisms*. New York: Basic Books.

Boden, M., d'Inverno, M., & McCormack, J. (Eds.) (2009). *Computational creativity: an interdisciplinary approach*. *Dagstuhl seminar proceedings: Vol. 09291*. LZI. http://drops.dagstuhl.de/portals/index.php?semnr=09291.

Part II
Music

Chapter 5
Musical Virtuosity and Creativity

François Pachet

Abstract Virtuosos are human beings who exhibit exceptional performance in their field of activity. In particular, virtuosos are interesting for creativity studies because they are exceptional *problem solvers*. However, virtuosity is an under-studied field of human behaviour. Little is known about the processes involved to become a virtuoso, and in how they distinguish themselves from normal performers. Virtuosos exist in virtually all domains of human activities, and we focus in this chapter on the specific case of virtuosity in jazz improvisation. We first introduce some facts about virtuosos coming from physiology, and then focus on the case of jazz. Automatic generation of improvisation has long been a subject of study for computer science, and many techniques have been proposed to generate music improvisation in various genres. The jazz style in particular abounds with programs that create improvisations of a reasonable level. However, no approach so far exhibits virtuoso-level performance. We describe an architecture for the generation of virtuoso bebop phrases which integrates novel music generation mechanisms in a principled way. We argue that modelling such outstanding phenomena can contribute substantially to the understanding of creativity in humans and machines.

5.1 Virtuosos as Exceptional Humans

5.1.1 Virtuosity in Art

There is no precise definition of virtuosity, but only a commonly accepted view that virtuosos are human beings that excel in their practice to the point of exhibiting exceptional performance. Virtuosity exists in virtually all forms of human activity. In painting, several artists use virtuosity as a means to attract the attention of their audience.

Felice Varini paints on urban spaces in such a way that there is a unique viewpoint from which a spectator sees the painting as a *perfect geometrical figure*. The

F. Pachet (✉)
Sony CSL-Paris, 6, rue Amyot, 75005 Paris, France
e-mail: pachet@csl.sony.fr

J. McCormack, M. d'Inverno (eds.), *Computers and Creativity*,
DOI 10.1007/978-3-642-31727-9_5,

Fig. 5.1 The Ryoanji stone garden in Kyoto. It is said that all stones are visible except one, wherever the sitting position

effect is similar to looking at a city photograph on which the figure would have been added with a digital picture editor. Moving away from this precise viewpoint slightly distorts the figure; moving further away breaks it into fragmented shapes, thus breaking the illusion, which reveals the unsuspected virtuosity of these apparently simple creations.

Similarly, artist Liu Bolin paints himself so as to become almost invisible, when he stands exactly at specific locations (near a balustrade, in a cinema with red chairs, etc.). In both cases, what is at stake, from our viewpoint, is the production of simple objects (geometrical figures in the case of Varini, mundane backgrounds in the case of Liu Bolin), together with evidence of the *difficulty* inherent to their realisation.

Another example in the visual domain is the *Ryoanji* stone garden in Kyoto. This garden is well-known for the calm and serene atmosphere it creates and many studies have attempted to uncover the reasons for its attraction (see e.g. Van Tonder et al. 2002). However, one reason stands out: wherever the watcher sits, only 14 out of the 15 stones are visible at a time (Fig. 5.1). Such a property turns an apparently random configuration of stones into a fascinating, singular creation. We argue that a reason for this fascination may also be that the object to see is again both simple and understandably difficult to create.

Virtuosity exists, or rather, occurs, also in time-related performance. People trained in performing fast mental computation compute operations several orders of magnitude faster than normal humans. Alexis Lemaire, world champion of the extraction of the 13th root of very large integers (200 digits), exhibits spectacular performance in all sorts of mental calculations. He calls this activity *hypercalculia* (Lemaire and Rousseaux 2009). What he produces is simple, but almost no one else can do it.

Virtuosity (from the Italian word *virtuoso*) is an essential dimension of music performance. In the Western culture, virtuosity in performance is a controversial notion and is the subject of many debates. On one hand, virtuosity is considered the greatest possible achievement of the art of solo instrumental performance (Valéry 1948, Penesco 1997). On the other hand, virtuosity is often considered in opposition to expressivity (see e.g. O'Dea 2000). But virtuosos are above all outstanding classical musicians (violinists in particular) who perform musical pieces known to be extremely difficult *at the limit of human capacities*.

In the field of poetry, virtuosity manifests itself under the form of 'satisfying difficult constraints'. It was shown for instance that the adaptation of Latin rhetoric to old English poetry created complex constraints for the authors. Satisfying these constraints was the source of great inventiveness and creation (Steen 2008). The association Oulipo (OuLiPo 1988) pushed very far the idea that constraints, in particular difficult ones, could be the source of inventiveness in literature and poetry. Novels by Georges Perec such as 'The void' (a novel without the vowel 'e'), or its counterpart 'Les Revenentes' (a novel with 'e' as the only vowel) are spectacular achievements of this movement.

5.1.2 The Cognitive Science Perspective on Virtuosity

Despite these achievements, virtuosity has hardly been addressed by cognitive science. From the viewpoint of physiology, there are known limits to the motor systems and the sensory-perceptive abilities of humans that are relevant to the study of virtuosity (London 2004; 2010). For instance, Fitt's law (Fitt 1954) states that the time it takes to reach an object is a function of the distance to, and the size of, the target object(s). Consequently, tradeoffs have to be found between speed and accuracy, both ingredients being required for achieving virtuosity, e.g. in music. Another important law governing human interaction abilities is the Hick's law (Hick 1952), which states that the time it takes to make a decision is a function of the number of possible answers:

$T = b \times \log_2(n+1)$ which generalises to: $T = b \times H$, where H is the entropy of the system.

These two rules combined yield the interesting argument that virtuosity is somehow only possible at the cost of not thinking. As Justin London (2010) sharply phrases it: 'Virtuosos can suppress the executive/monitoring functions of their brains when they perform; and thereby avoid the speed traps of their prefrontal cortices'.

The way to achieve this is by intense training. The *10,000 hour rule* (see e.g. Ericsson et al. 1993, Sloboda et al. 1996, Gladwell 2008) states that about 10,000 hours of training are required to become a world expert in any domain. Most biographies of well-known musicians confirm the fact that music virtuosos (in classical music, jazz, and even pop) have spent most of their youth training (Mozart, Charlie Parker, John Coltrane, Biréli Lagrène, The Beatles).

5.1.3 Virtuosity as an Attraction Device

Bird songs are particularly interesting for virtuosity studies as they are a rare case in which the whole production and reception process has been studied in-depth, yielding breakthroughs and fascinating findings.

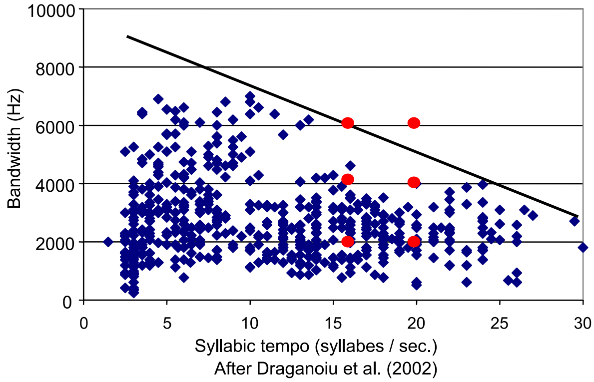


Fig. 5.2 The distribution of canary phrases, in a bandwidth/tempo space, representing the natural tradeoff between bandwidth and syllabic tempo. *Red circles* represent the phrases used for the experiment. The artificial top right phrases optimising the two features in unrealistic ways were the most successful

Researchers in animal behaviour have long been interested in the phenomenon of bird song production and its role in the mating process. In several bird species, male birds produce songs primarily to attract females. The issue of what makes a bird song more attractive than others has received particular attention in recent years. Various results have shown that specific features of songs can account for their popularity. For instance, great reed warbler females (*Acrocephalus arundinaceus*) exhibit a preference for long songs over short ones in the wild (Bensch and Hasselquist 1991).

More interestingly, the study by Draganoiu et al. (2002) focused on the case of the domesticated canary (*Serinus canaria*). Male canary songs have a specific phrase structure. Two features of these phrases were shown to significantly increase liking: frequency bandwidth and trill rate. However, it was also shown that these two features are somehow contradictory: similarly to Fitt's law, a tradeoff is observed in real phrases, due to the specific motor constraints of the birds vocal track.

The breakthrough experiment of Draganoiu et al. (2002) consisted of synthesising artificial phrases optimising these two features in an unrealistic way that is 'beyond the limits of vocal production'. The exposition of these artificial phrases to females birds showed unequivocally that females preferred these phrases to the natural ones (see Fig. 5.2). An interesting interpretation for this preference is that the production of 'difficult' phrases maximising both bandwidth and syllable rate may be a reliable indicator of male physical or behavioural qualities.

This evolutionary argument emphasises the role of virtuosity in music appreciation. In popular music, virtuosity is explicitly present in specific genres (e.g. so-called *shredding* in hard-rock, illustrated by guitarists such as Yngwie Malmsteen or melodic-harmonic virtuosity in bebop), as we show below.

5.1.4 Virtuosos as Creators

In this chapter, we adopt a specific perspective on virtuosity. From the viewpoint of complexity and computer science, we envisage virtuosos as exceptional *problem* *solvers*. Virtuosity can be objectively measured, observed, and as such is, 'as a concept, closer to the ground, than creativity' (Howard 2008).

Indeed, the capacity to effortlessly navigate in large search space in real-time is not only a matter of physiological prowess. By transferring part of the decision processes to the body, a virtuoso naturally compiles his knowledge in a remarkable way that can teach us a lot about innovative problem-solving.

For instance, virtuosos in mental calculation invent and make extensive use of so-called mathematical *tricks*. As an example, squaring any number ending in 5 can be done easily using a simple formula (take the first digits except the last 5, multiply it by itself plus 1, and then concatenate 25 at the end). Some of these tricks are well-known, but many others are not, and probably ignored by their inventors: intense training may result in exceptional performance, not necessarily in clear explanations. In the following sections, we show how jazz virtuosos produce interesting inventions, and how modelling this virtuosity leads to interesting insights about the nature of invention and creativity.

5.2 The Case of Jazz

Much like language, the ability of humans to spontaneously improvise music in real time is considered by many as an extraordinary skill, a sort of magic. Most of this magic, again, comes from hard training. As Levine (1995) states in his introduction: "*A great jazz solo consists of 1 % magic and 99 % stuff that is Explainable, Analyzable, Categorizeable [sic], Doable. This book is mostly about the 99 % stuff.*" This chapter is about putting the *99 %* stuff in a machine, and making the remaining *1 %* explicit. In particular, our aim is to separate clearly what can be reasonably automated—what virtuosos are able to do unconsciously—from what emanates from artistic, conscious decision making.

Invented in the 1940s with Charlie Parkerand Dizzy Gillespie, bebop is an idiom of jazz where a strong emphasis is put on melodic and harmonic dimensions.Virtually all instruments of the classical orchestra have been used by bebop musicians. Nowadays, bebop musicians continue expanding the style. The case of jazz bebop improvisation is particularly interesting because the specific constraints of bebop are shared unambiguously and can be easily expressed using well-defined languages: In some sense, jazz improvisation is a special form of computing.

Scientists have long tried to debunk the magic of jazz improvisation, starting with the psychologist Philip Johnson-Laird. His work is not to be judged by the musical quality of his algorithmic productions, but by the seminal nature of his arguments. One of his main claims is that the ability to produce an improvisation does not require any 'short-term memory' (Johnson-Laird 1991; 2002). He demonstrated this idea by proposing memoryless automata that automatically generate rhythmic and melodic material. Since then, more powerful algorithmic techniques have been used to produce jazz improvisation (see Sect. 5.3), but it can be said that the problem of modelling 'basic' bebop improvisation has been solved, notably by exhibiting improvisation generators satisfying the basic rules of the game (detailed in Sect. 5.2.3).

However, the improvisation problem has been only partially solved. Trained jazz musicians listening to the examples produced by these previous works rarely experience the feeling of hearing a machine outperforming humans.

In fact, professional bebop musicians are like Olympic sportsmen or chess champions, reaching a level of technicality which is far beyond the capacities of a beginner. They are usually sought after not so much because they exhibit a general 'ability to improvise'—children can also improvise—but for their specific display of virtuosity. Contemporary jazz improvisers such as John McLaughlin, Al Di Meola, Biréli Largène (guitar), or Stefano di Battista (saxophone) exhibit a level of virtuosity that seems to reach beyond the limits of what most humans can do (the expression 'not human' appears indeed often in commentaries about these performances on social Web sites). They play intricate phrases at such a speed that even the transcription of their solos from recording is a challenging task. Deciding which notes to play at that speed seems indeed impossible, so the virtuosity question can be rephrased as: How can one perform and execute these musical choices so accurately and so fast?

Of course, performance as well as timbral dimensions are undoubtedly important in music, and can themselves be the subject of virtuosity display (Bresin 2000), but these are outside the scope of our study: Following the argument that 'bebop is more about content than sounds' (Baker 2000), we focus here on the melody generation task. We consider that virtuosity is not only an appealing facet of bebop, but one of its *essential features*. This situation bears some intriguing analogy with bird singing behaviour. Though bebop virtuosity is not only about speed as we will see below, this analogy suggests a primary role of speed in the attraction for specific melodic movements.

5.2.1 The Rules of the Game

In this section we define precisely the musical corpus we target: linear improvisation, which corresponds, roughly speaking, to virtuoso passages of bebop improvisations.

5.2.2 Bebop Phrases

Virtuoso phrases are played fast, typically 1/16th notes at 120 bpm or more, which represent at least 8 notes per second. This speed implies a number of characteristics for these passages that we call 'linear'. The term linear has been used in the jazz theory literature (e.g. Ricker 1997) to describe phrases built from scales rather than from chords (i.e. arpeggios), thereby creating a sensation of melodic or horizontal

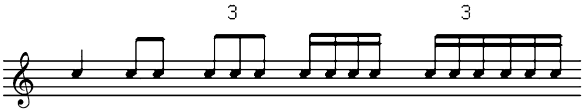


Fig. 5.3 Examples of various rhythms used during linear improvisation

consistency. More precisely we define linear improvisations as phrases which are (1) played fast (eighth-notes or faster), (2) monophonic, (3) without silences, and (4) rhythmically *regular*.

All these criteria are implied by speed: monophony because it is usually impossible to play fast a polyphonic instrument. Regular rhythm means that each beat in a measure is played with notes of the same durations (for the sake of simplicity, 1/4 notes, 1/8 notes, 1/16 notes, or triplets thereof, see Fig. 5.3). Rhythmic regularity is also implied by speed as it is very difficult to change the rhythm when playing fast. Linear improvisation is pure melodic invention, other musical dimensions are secondary.

All virtuoso bebop musicians include linear passages in their choruses. Virtuoso improvisations are of course rarely entirely linear, but they are often at least locally so. As we hypothesise, linear passages correspond to a specific, intentional mode of musical production deliberately chosen (at some risk) by the musician.

5.2.3 The Melodic/Harmonic Interplay

Bebop improvisation is a particular form of tonal music in which harmony plays a central role. Given a chord sequence, usually taken from a shared repository such as the *Real Book* (Real 1981), the game consists of producing a stream of notes that satisfy two criteria simultaneously: *harmonic consistency* and *continuity*. This game, commonly referred to as 'playing or negotiating the changes', can be considered the main technical challenge of bebop improvisation (it is also possibly a key ability in other domains, such as management, see Holbrook 2009). Paradoxically, an extra recipe for producing interesting melodies consists of breaking the first rule through various *escape* mechanisms, as we will see below.

To the ears of a trained musician, virtuoso choruses rarely contain any mistakes with regards to these principles. This is a striking property of virtuosos in general, and jazz improvisers in particular: they produce *perfect melodies*, sounding as if they were proof-read before being delivered. We will now review these principles.

5.2.3.1 Harmonic Consistency

The generated melody must comply with the current chord in the sequence. Strictly speaking, this means that the melody should consist mostly of notes which belong to a scale appropriate for the chord. The identification of the correct scale requires, in principle, an analysis performed on the whole chord sequence. Chord sequence analysis was shown to be a non-trivial task in general (Steedman 1984). In practice, however, simpler forms of analysis are used, which consist in using ready-made associations between chords and scales. Such associations are routinely available in harmonic treatises, e.g., in Aebersold (2000). For instance, on a *minor 7*(*5b*) chord, say *D minor 7* (*5b*), one can use a *harmonic minor* scale one minor third above the root (here, *F harmonic minor*).

5.2.3.2 Continuity

Jazz beginners often improvise by playing arpeggios corresponding to each chord: this simple technique satisfies local harmonic satisfaction by definition, but produces obviously uninteresting, unmelodic phrases. Producing a 'sense of melody' is difficult to define precisely. Continuity is a good approximation and is easier to define. We will see that low-order Markov processes exploiting carefully chosen scales guarantee a form of natural continuity.

Melodic continuity is a difficult challenge for a human when playing fast, as it requires the ability to find quickly short paths between the note currently being played and the next ones, which may be in a different scale. This ability is referred to as chord change negotiation, stressing its inherent problem-solving dimension.

Note that continuity does not necessarily imply *brownness*, in the sense of (Voss and Clarke 1978), i.e. the sole use of small intervals. It rather implies that notes are glued together smoothly, and not made up of isolated elements or patterns, concatenated without care. For instance, the phrase in Fig. 5.7 contains several large intervals but is perfectly continuous.

The One-Step-Max Theorem There is a factor that helps address the continuity challenge: the *one-step-max theorem*. The scales used in jazz (minor, major or diminished, in first approximation) contain an interval of maximum 3 semitones (in the harmonic minor scale). Consequently, any note is always within 1 semitone maximum (up or down) to a note of any possible scale, i.e. a 'good' note. We will see below how this theorem can be used as a rescue mechanism when the basic generator fails to find a solution.

5.2.4 Playing Outside and Side-Slipping

The bebop language is deeply grounded in tonal harmony. However, like all languages, bebop evolves. One important development was caused by a paradoxical force that pushes musicians to escape the constraints of harmonic satisfaction, once they know how to satisfy them perfectly: *playing out* in jazz jargon. Playing out is not to be confused with *free jazz*, a radical way to escape the laws of tonal harmony,

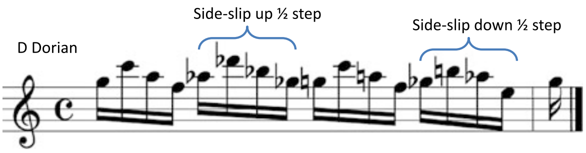


Fig. 5.4 Example of a side-slip, given by (Coker 1997, p. 50). Note how the first side-slip smoothly continues in the 'right key' (here, D minor)

in which there are no more rules whatsoever. Playing out, in bebop, is a precisely defined musical device whose mastery necessitates perfect control of the instrument. The ability to play out 'the right way' can be considered a sign of a complete mastery of the art.

The main way to play out is called *side-slipping* (or *side-stepping*). Shim (2007) dates the origin of side-slipping back to Art Tatum, an acknowledged piano virtuoso, who 'displayed his mastery of chromatic harmony by effortlessly floating in and out of keys' (p. 183). A stepping stone of this evolution is probably the incursion of modal improvisation in the jazz repertoire; with the famous tune 'So What' by Miles Davis, based on a long repetition of a D minor chord. To avoid a 'tide of boredom' due to this harmonic monotony, various techniques for escaping tonality were invented, including side-slipping (Coker 1984, p. 49).

Pedagogical definitions of side-slipping may be found in jazz theory books (Coker 1984; 1997, Levine 1995), with some variations. Side-slipping is a device that produces a short sensation of surprise, in a context deemed too predictable (Levine 1995). The idea is to play out-of-key, with the goal of momentarily creating tension, and then come back to the right key, which can be different from the starting key. Most often, the out-of-key segment uses symmetry. For instance, it can be the same phrase transposed a semi-tone higher. The listening impression is described by Coker (1984) is as follows: 'Like the stretching of a rubber band, the attuned listener seems to know that the player's excursion into another key is very temporary and that he will snap back to the original key when the tension period is over. In the meantime, the listener has been taken on a brief trip that has broken the monotony of modality'. Side-slipping was intensively used by pianists like Lennie Tristano (Shim 2007, p. 183), and many others (John Coltrane, Allan Holdsworth) and is now a classical ingredient of modern improvisation.

Figure 5.4 shows a typical side-slip, given by Coker (1997, p. 50). The mechanical dimension of the side-slip appears clearly: here a simple transposition of a 4-note pattern one semitone up, and then down. Figure 5.6 shows a more complex example of a side-slip produced backward in time, i.e. played before the non-transposed version, creating an even more surprising effect (shocking, then deliciously soothing). Note that such an effect would not work if played at low speed, as the time during which wrong notes are played would be too long, creating a risk for the listener to lose the sensation of tonality. As such side-slipping is not an ornamental device, but a central feature of linear improvisation.

There are many variants of side-slipping, notably concerning the device used to produce the phrase *out of key*, its length, metric structure, etc. (Coker 1997). For

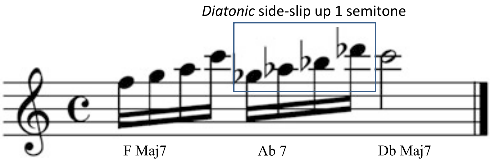


Fig. 5.5 A diatonic side-slip invented by Coltrane. This particular side-slip is such that it actually does not create any harmonic tension, as the transposed motif (up 1 semitone) stays, miraculously, in the tonality (here, one minor third above)

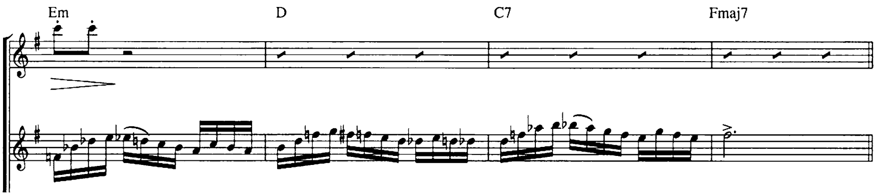


Fig. 5.6 A tricky example of a 'reverse side-slip' by Al Di Meola in a chorus on Guardian Angel (GuitarTrio 1977). The two first phrases are transpositions, (2 then 1) semitones lower, of the last one, which is in the right key, thereby creating a stunning resolution effect, only made possible by speed (here, 184 bpm)

instance, Fig. 5.5 shows a diatonic side-slip invented by John Coltrane (and used extensively e.g. on his improvisations on *Giant Steps*). This is a nice 'trick', or rather a small theorem of tonal music: when a motive in some major key (say, F) is transposed up 1 semitone, it is most of time in the initial key transposed 1 minor third up (here, Ab7).

The difficulty for improvisers is not only to produce the slide-slip, but to re-establish continuity during the *re-entrance* phase. This necessitates tricky planning, as the final notes of the transposed pattern are, usually, precisely out of key, so no natural continuation may be in the musician's hands.

We will see how our framework copes with side-slipping in a general way, by allowing side-slips to be inserted smoothly in the generated chorus while keeping continuity.

5.2.5 Virtuosity Is to Improvisation as Running Is to Walking

Virtuosity is about speed, but not only speed. Beyond speed—innate for computers—virtuosity is the capacity to play *interesting* phrases fast, and make them appear as singular solutions to a difficult problem, much like a magician tirelessly extracts rabbits from a shallow hat. Like running is not walking faster (Cappellini et al. 2006), playing virtuoso phrases calls up cognitive skills and motor mechanisms that differ from the ones used in standard improvisation, which consists basically of paraphrasing the original melody (Baggi 2001). In this view, virtuosity

Fig. 5.7 A virtuoso passage (152 bpm) in a chorus by John McLaughlin on Frevo Rasgado (GuitarTrio 1977). Note the 'smooth' chord transitions

is a specific improvisation *mode*, in which the musician deliberately chooses to enter and exit, during his solo. Often, virtuoso passages constitute the climaxes of the chorus. This is obvious in concert recordings such as the GuitarTrio (1977) in which virtuoso passages (see Fig. 5.7) are followed by enthusiastic rounds of applauses.

An important apparent characteristic of virtuosity in bebop is that the musicians give an impression of precisely *controlling* their production, using some sort of high-level inner commands. Such an impression is obvious when listening to the effortless character of Art Tatumis' improvisations, which allow him to flow in and out of harmonies with a total control on their high level structure. Indeed, the improviser's ultimate fantasy is probably not to *produce* but to *control* such a virtuoso flux (Sudnow 1978), through high-level mental commands. In short, to be the director of one's inner orchestra. How is this possible?

5.2.6 Claims

In this chapter, we make a number of claims. The main one is that we present a system that generates virtuoso phrases of the same musical quality as the ones human virtuosos produce. The validity of this claim is left to the appreciation of a trained jazz listener, who can judge from the outputs (scores and videos) of our system, *Virtuoso*, available on the accompanying web site.

The second claim is that we propose an operational answer to the virtuosity question (how do they do that?), by introducing the notion of *intentional score*: the temporal series of high-level musical decisions taken by the virtuoso to generate a chorus. These decisions are arbitrary, and may be seen as the '*1 % magic*' mentioned by Levine in his introduction (see Sect. 5.2). This intentional score is the backbone for automatically producing virtuoso phrases, and our system may be seen as an *interpreter* of this score, which generates a chorus that satisfies it, i.e. Levine's '*99 % stuff*'. We show through our various examples that this score suffices to generate virtuoso phrases of high quality. All the decisions in the intentional score are done at the *beat level* (and not at the note level), i.e. at a *low frequency*, thereby substantially reducing the cognitive load of rapid note-level decision making. This explains how the bypass of high-level cognitive decision-making may be operated in practice (see Sect. 5.1.2).

Most importantly, we show how human jazz improvisers have contributed, at least in two respects to *inventing the bebop style* (and its extensions) *thanks to* *virtuosity*. The two features we focus on are only possible thanks to extreme virtuosity: (1) side-slips and (2) fine-grained control. We describe and interpret these two major contributions to style invention in the context of Markov-based music modelling.

After a review of the state-of-the art in jazz modelling, we describe a Markov-based model of jazz improvisation and show that it is well adapted to generate melodies that fit with arbitrary chord progressions. We then use this basic model to *state and solve* the two main issues of jazz generation: control and side-slips.

5.3 Modelling Jazz Improvisation Generation

Many studies have addressed music composition and improvisation, so we focus on those specifically addressing jazz. As is often the case in computer science, these studies follow the general algorithmic trends of the moment. We handle separately the case of Markov modelling as this is the core of our proposal.

5.3.1 Non-Markovian Approaches

Ulrich (1977) proposed an all-encompassing system that performs chord sequence analysis and chorus generation using a purely algorithmic approach, reaching a reasonable level of musicality. Walker (1997) and Thom (2000) built interesting systems emphasising the *dialogue dimension* of improvisation rather than the musical quality. A more ambitious case-based reasoning approach was proposed by Ramalho and Ganascia (1994), emphasising the role of motivic components, and following the 'knowledge level' paradigm. This approach proposes to explicitly reconstruct a cognitively plausible model of a working jazz memory, and was applied to the automatic generation of bass lines, yielding some interesting outputs, favourably compared to Ron Carter's samples (Ramalho 1997). It relied on a manually entered set of cases, limiting its scope in practice. Genetic algorithms have been used for music generation by a number of researchers (Weinberg et al. 2008, Bäckman and Dahlstedt 2008, Papadopoulos and Wiggins 1998), yielding real time systems used in concert, and producing interesting improvisation dialogues, like in the *GenJam* system of Biles (1994). These systems, again, apply a general paradigm (here, evolutionary algorithms) to chorus generation in a top-down approach, without concern for harmonic satisfaction and continuity. Their outputs, although sometimes spectacular, are still below the level of professional musicians, and do not display particular virtuosity. Interestingly, the system described by Grachten (2001) was used as a basis for studying jazz *expressivity* in saxophone solos (Ramirez et al. 2008). The studies described in Hodgson (2006), also use a genetic algorithm but focus on detailed characteristics of the Charlie Parker style. Notably, Hodgson shows the importance of dyadic (two-note) patterns in the elaboration of Charlie Parker's melodic repertoire. But the use of random generation, intrinsic to evolutionary algorithms, here also, gives results of varying quality, necessitating manual editing (the author describes the results as 'partially correct'). Note that manual editing echoes the approach of Harold Cohen with his AARON panting program (McCorduck 1991), as well as that of David Cope (1996), who use partial manual editing to finish their compositions. We will see below how we substitute manual intervention by *intentional controls*, and the implication on our models.

Probabilistic grammars were used by Keller and Morrison (2007) to generate jazz improvisation. The outputs of their system 'compare favourably with those played by college-level jazz students of at least an intermediate playing level, if not better'. The grammar rules are manually encoded, and based on an explicit representation of note harmonic status (chord tone, passing tones, etc.). Note-level information is required when teaching improvisation, but we do not think they are necessary for generating improvisation. As we will see, we adopt an approach in which this information is not represented explicitly. However, our generated phrases do contain a natural blend of, e.g. chord tones and passing notes. Note-level characteristics naturally emerge from the generator, rather than being prescribed by the system.

Franklin (2006) showed that recurring neural networks could learn entire songs given a melody and the associated chord sequence, and produce new improvisations on these chord sequences. This system demonstrates that non-symbolic approaches can capture some of the knowledge of jazz musicians, but the results shown are also college-level.

Side-slipping is briefly mentioned as a possible composition operation in the *ImproVizor* system (Keller et al. 2005), but the process and more generally the devices for playing out-of-key 'the right way' have not yet been the subject of modelling in improvisation generation studies. Finally, it can be noted that commercially available software like *Band-in-a-box* (PG Music), or the *Korg Karma*™ family of synthesisers produce reasonable improvisations in real time, over arbitrary harmonic constraints, using algorithmic approaches. These systems may produce musically interesting outputs, but their analysis is difficult because of a lack of published technical information.

5.3.2 Markov Chain Approaches

Other approaches to jazz improvisation based on Markov chains have been explored recently, showing notable success. These systems follow a long tradition in computer music modelling, dating back to the works of Shannon on information theory (Hiller and Isaacson 1958, Brooks et al. 1957). Markov processes are based on the 'Markov hypothesis' which states that the future state of a sequence depends only on the last state, i.e.:

$$p(s_i|s_1,\ldots,s_{i-1}) = p(s_i|s_{i-1}) \tag{5.1}$$

Extensions to higher orders as well as variable-orders (Ron et al. 1996) do not change substantially the basic principle of Markov generation. The Markov hypothesis, in all its forms, can be seen as a concrete implementation of Longuet-Higgins's *memoryless assumption* (see Sect. 5.2).

The Markovian aspects of musical sequences have long been acknowledged, see e.g. Brooks et al. (1957). Many attempts to model musical style have therefore exploited Markov chains in various ways (Nierhaus 2009), notably for sequence generation.

Many experiments in musical Markov models have shown that there is indeed a strong Markovian dimension in musical surface in most genres of tonal music, including jazz (see e.g. Nierhaus 2009 for a survey). The *Continuator system* (Pachet 2003) was the first to propose a real-time improvisation generation system based on Markov chains, producing sequences as continuations of input sequences played by humans. This system was shown to deliver striking results, even passing 'jazz Turing tests' (Van Veenendaal 2004).

Most Markov generators are based on a *random walk* process, exploiting a probabilistic model of the input phrases. The generation is performed step-by-step, in our case, note by note, using a random draw scheme, which takes into account the context, i.e. the phrase generated so far:

Iteration at step i:
$next = Random\_Draw(context_i)$;
$context_{i+1} := Concatenate(context_i, next)$;

In practice, the context is limited to a certain maximal *order*. Random choice is performed as a weighted random draw, using an efficient representation of all encountered suffixes computed from the training set, which yields a probability table. The generated event is then concatenated to the context, and the process is iterated.

It has been shown that this model enables the creation of realistic outputs in many musical styles, with professional musicians (Pachet 2003, Assayag and Dubnov 2004) as well as children (Addessi and Pachet 2005). Like previous approaches, these systems use a general, agnostic algorithm, uniformly applied to music sequences of any style. Consequently, the qualities of its outputs are also independent of the style of its inputs, and uniformly good … or bad. However, it should be noted that these systems perform best in *musically unconstrained* contexts, such as free-form improvisation. No convincing results were obtained when used in a *bebop* setting with the constraints we have introduced in the preceding section.

Random walk approaches have shown limitations when used for generating *complete pieces* as this strategy does not always favour the most probable sequences in the long term (Conklin 2003). This is not an issue in our case, as we will see how the generation can be controlled using higher-level *controls* that determine global characteristics of the generated sequences, taking precedence over the details of the basic generation algorithm. Indefinite memory length is the main claimed advantage of the system proposed by Assayag and Dubnov (2004). In our context, this problem is irrelevant, as our goal is not to reproduce similar pieces, but to use the training samples to generate novel melodies in a highly constrained context. We consider a variable-length generation model, but, following Hodgson (2006), we restrict our maximum length to 2, an intentionally short value, which ensures an optimal compromise between similarity and creativity.

Another problem is related to the case where no solution is found (*NSF* hereafter). This happens when the context has not been encountered in the training phase. This problem, known as the *zero-frequency* problem has been addressed by many researchers in Markov modelling (see e.g. Chordia et al. 2010), with no general solution. Here again, we favour an approach based on the observation of bebop practice, and propose a bebop-specific, simpler solution, described below.

5.4 A Note-Based Jazz Generator

The basic engine in our proposal is a variable-order Markov chain generator, with a maximum order of 2. This generator, described in the preceding section, is able to yield the 'next' note, given the last 2 notes (at most) already played. Our experience has shown that augmenting the memory length does not improve the quality of the generation.

5.4.1 Pitches for Representation, Beats for Generation

All major decisions for generation are taken at the beat level, and constitute *in detail* the *intentional score*, which is a temporal sequence of beat-level decisions. These decisions are the following.

At each beat, a rhythm is chosen (arbitrarily in the first approximation) within the 5 possibilities described in Fig. 5.3 (see Fig. 5.9). This rhythm in turn determines the number of notes to produce for the beat (in our case, 1, 2, 3, 4 or 6). Consequently, there is no need to use durations as training data, as these durations are entirely determined by this rhythm choice. The velocities of each note are the velocities played in the training corpus (see below). No harmonic information is used in the training phase either, as the model used for generation is chosen, for each beat, according to the current chord, as described below. Higher-level attributes such as pitch contour, chromaticity, etc. are handled yet at another level as described in Sect. 5.5.2. Consequently, the representation used for the Markov model is based solely on pitch, reducing this basic mechanism to a simple one.

The justification for this choice is based on a long experience with Markov models for jazz, which convinced us that pitch is the only dimension of music that is well captured. Although other dimensions can technically be represented as such, it does not make much musical sense. There are two main reasons for this: firstly, only *intrinsic* attributes, by definition, are well adapted to Markov modelling. Pitch is an intrinsic attribute, but not rhythm, which emerges from the relation between adjacent notes or events. Second, there is no concrete evidence that modelling higher-level dimensions (harmony, pitch contour, etc.) yields interesting musical material, as these dimensions are correlated to each other in intricate and complex ways, raising the 'viewpoint problem' that inevitably leads to *ad hoc* solutions and compromises. In some sense, the situation is comparable to the *multiple inheritance* problem in object-oriented languages (Stein 1992): it works well when there is no conflict, but all the solutions proposed to solve the problem in the general case failed and were progressively abandoned.

5.4.2 Handling Harmony

There are several ways to consider harmony in a Markovian context. One of them is to consider harmony as a specific musical dimension, and use it as a viewpoint. This approach is followed for instance by Conklin and Witten (1995) or Cont et al. (2007). As discussed above, simultaneously handling several viewpoints creates viewpoint interaction problems that do not have general musically meaningful solutions. Furthermore, it introduces unnecessary level of complexity in generation. In our case, we can observe that chord changes in bebop never occur within a beat (they usually occur at the measure of half-measure level, sometimes at the beat, never within a beat). Hence our solution is simply to use chord-specific training databases, which are selected at each beat according to the underlying chord sequence.

More precisely, we use a simple set of chord/scale association rules. Such rules can easily be found in jazz theory text books, e.g. Aebersold (2000). For each chord type appearing in a chord sequence, we select the Markov model which corresponds to a particular 'scale' (Fig. 5.8). Using various substitution rules, it is easy to reduce the number of needed scales to a much smaller number than the number of chords. A drastic reduction is proposed by Martino (1994) who uses only minor scales throughout all chord sequences, using clever chord substitutions (e.g. *C 7*th chord uses the *G minor* scale, *C altered* uses the *G# minor*, *C maj7* uses *A minor*, etc.). Although the Martino case is easy to implement (and is available in our repertoire of styles) we follow here a more traditional approach, and propose five scales: major, minor, diminished, seventh and whole tone (for augmented chords). As a consequence, we only need training data for these five scales, in a single key (C). The databases for the other keys are simply transposed from the ones in C.

Many variations can be introduced at this level, such as chord substitutions (see e.g. McLaughlin 2004). These can be typically performed at random, or according to any other parameter (e.g. user controls), and belong naturally to the intentional score. An important aspect of this method is that it is independent of all other parameters of the system, and notably does not necessitate an explicit memory of the past.

Here again, our solution is analogous to the way humans improvisers practice and improvise, as illustrated by the huge literature proposing training scales and patterns.

```
selectHarmonicDatabase (chord)
        if chord is major, major 7, major 6 then return MajorDB;
        if chord is minor, minor 7, minor 6 then return MinorDB;
        if chord is half diminished then return HalfDimDB;
        if chord is 7 altered then return SeventhAlteredDB;
        if chord is augmented 5 then return WholeToneDB;
```

Fig. 5.8 The selection of a harmonic database according to the current chord

```
GenerateBeat(context, i) // context = the last generated output
        RP := chooseRhythmPattern;
        H := selectHarmonicDatabase (i chord);
        segment := new empty segment;
        Repeat N times (N = number of notes in RP)
                next(H) = Random_Draw (H, context);
                segment := Concatenate (segment, next) ;
                context := Concatenate (context, next) ;
        return segment with rhythm
```

Fig. 5.9 The basic GenerateBeat function integrates all musical decisions. N is the number of notes per beat, H is the harmonic context, which determines the Markov model to be used. H is constant during the beat

Fig. 5.10 A minor scale played up and down, used as the sole training phrase

Changing Markov databases at each beat also creates a potential problem with regards to continuity: how to ensure that phrases evolve continuously during chord changes? It turns out that there is again a simple solution to chord change negotiation, which does not necessitate modifying the generation algorithm, but consists of *carefully choosing the training corpus*. In cognitive terms, means that all possible chord changes have at least one solution.

Let us consider several cases in turn, to illustrate the Markov process. We start by a training sequence in the key of *A harmonic minor* consisting of a scale played up and down (Fig. 5.10). Using our generator, we can produce phrases in all minor keys like the one illustrated in Fig. 5.11 (still in *A minor*). Other keys are handled simply by transposing the input sequence.

By definition, the generated phrases will all be Brownian, in the sense of Voss and Clarke (1978). This is caused by the fact that each pitch (except for the extremes) has only two possible continuations—one above and one below—in the diatonic scale used for training.

Fig. 5.11 A phrase generated by the Markov generator from the unique training phrase of Fig. 5.10. Phrases generated by diatonic scales are all Brownian by definition

Fig. 5.12 A phrase generated on top of an alternating *A min/A# min* chord sequence, using the single ascending/descending A minor scale as training phrase. Note the two cases where no continuation is found to negotiate the chord changes (indicated by an *arrow*)

5.4.3 Chord Change Negotiation

Let us consider now a chord sequence based on alternating between *A minor* and *A# minor*. We deliberately choose *A# minor* as this key is 'far away' from *A minor*, and therefore harder to 'negotiate', because these two scales share only a few notes. Figure 5.12 shows an example of a phrase generated on this sequence. We notice that there are two NSF cases. They correspond to situations in which the last note of a phrase for a given chord does not exist in the training phrase for the next chord. Here, *C#* does not exist in the training base for *A minor* (first case), and *B* does not exist in the training base of *A# minor*, by definition of the harmonic minor scale.

Contrarily to general approaches to the *zero-frequency problem*, we propose a musically justified solution with the two following arguments:

1. We reduce the number of NSF cases by carefully choosing the training corpus, as detailed in the next section. This step corresponds to *human training*;
2. In the remaining (rare) cases no solution is found, we use a simple heuristic based on the *one-step-max theorem* (see Sect. 5.2.3.2): since there is always a 'good note' at a maximum pitch distance of one semitone, we try both and select the one that works, and we are guaranteed that there is always one.

This double solution turns out to work nicely. It can be seen in Fig. 5.12 that in both NSF cases, the system chooses the right notes a semitone apart to fit with the harmony. The resulting phrase sounds smooth and continuous as if *nothing had happened*: it is virtually impossible to notice that the generated phrase is locally not Markovian. Furthermore, the system can easily produce a report after a series of improvisations, to suggest adding a training phrase containing the NSF cases encountered. In our case it could suggest the musician/system to practise/add phrases

Fig. 5.13 A generation using only the harmonic scale as training base, on a succession of minor chords progressing one semitone up. NSF are indicated at chords #3, #4, and #7

in *A minor* containing a *C#* or an *A#* (i.e. *B* in *A# minor*, once transposed in *A*): an interesting case indeed, which forces the use of chromaticisms or passing notes in a clever fashion. Figure 5.13 shows a more complete example on a succession of chromatically ascending minor chords.

It is interesting to note that this approach to the NSF problem corresponds to the pedagogical strategy proposed by Pat Martino (1994): learn only one scale (minor), but learn how to use it on any chord change to another minor chord. This implies practising over 12 possible changes from one minor scale to another one, in all keys, so a total of 'only' 132 cases from which any chord sequence can be smoothly negotiated (Pat Martino proposes a solution to substitute any chord by a minor chord, see Sect. 5.4.2).

Other strategies could be used, such as simply finishing the phrase. However, this kind of heuristic gets in the way of our modelling goal: the decision to stop or end a sentence is a 'high-level' one that should not rely solely on such low-level technical considerations, but only on the musical intention of the musician.

This mechanism produces phrases which satisfy local harmonic constraints, chord negotiation and continuity. However, the phrases wander up and down according to chance, and there is no direct means of controlling their structure. In some sense, this represents *technical virtuosity* (the ability to play fast), but not *controlled virtuosity* (the ability to play what you want). This most important issue is addressed in Sect. 5.5.2.

5.4.4 An Example Training Set

Obviously the choice of training phrases is crucial to the generation, as only these phrases are used to build the improvisation. Experiments using inputs entered in real time are problematic as errors cannot be corrected once learned by the system. Markov models have not been used, to our knowledge, in a *controlled setting* for jazz improvisation. Here again, the particular context pushes naturally to a careful selection of training patterns, like human improvisers do when they practice. But which phrases are the right phrases?

Fig. 5.14 Phrase #1 in C minor

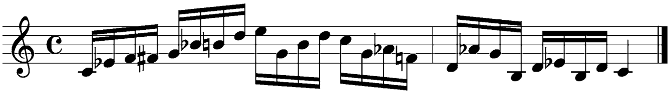


Fig. 5.15 Phrase #2 in C minor

Fig. 5.16 Phrase #3 in C minor

The example given above suggests a constrainton the training phrase: to ensure continuity (and avoid NSF cases), each Markov model should contain all pitches. This is a sufficient condition, by definition, but not a necessary one. Our repair strategy handles graciously the cases where no solution is found. Other more subtle constraints can be deduced from the desired nature of the improvisations to generate, dealing with longer patterns. For instance, the density of the Markov network determines the nature of the generated phrases: the more choice there is for a single note, the more possibilities there are for controlling the phrase. If a single note has only one possible continuation, there is a risk of producing repeated patterns when reaching this particular note. Note that this is a current situation with human players, who sometimes learn only a small number of escape solutions, when reaching particular notes or passages (on guitar, this is often true for notes played in the top of the neck). A static analysis of the Markov model can reveal such bottlenecks, and be used to suggest new phrases to learn to create new branching points.

To illustrate the generation of phrases from the training phrases, we describe a part of a Markov model, specifically designed to represent a 'classical' bebop player, with no particular stylistic influence. We give here the complete set of phrases used in the *minor* scale. These phrases are played in the key of C, and then transposed in the 11 other keys. The interested reader can find the corresponding database for the other scales in C (major, diminished, seventh and whole tone) on the accompanying web site (http://www.csl.sony.fr/Virtuosity). These other databases are similarly transposed in the 12 keys.

The following six phrases (Figs. 5.14–5.19) were designed (by the author) to contain basic ingredients needed to produce interesting jazz melodies in C minor. Of course, they do not contain all the patterns of this style, as this would be an impossible task, but they can be enriched at will. As can be seen, not all pitches are present in the database (at least for all octaves). This is intentional to show how the mechanisms we present here interact with each other.

Figure 5.20 shows a phrase generated on a chord sequence consisting only of a C minor chord. The various segments of the training phrases are indicated, showing

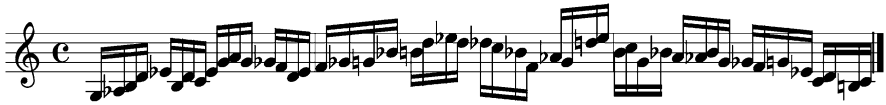


Fig. 5.17 Phrase #4 in C minor

Fig. 5.18 Phrase #5 in C minor

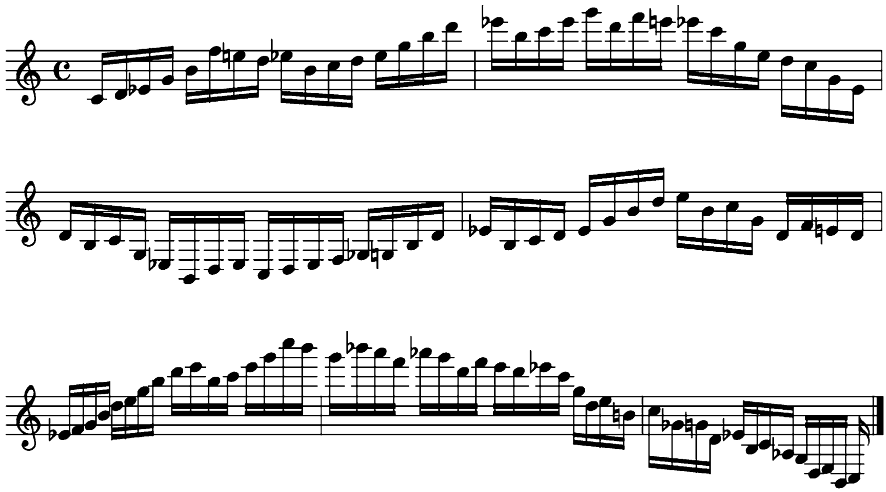


Fig. 5.19 Phrase #6 in C minor

how Markov generation creates a new phrase by concatenating bits and segments taken from the training set.

Figure 5.21 shows a phrase generated on the chord sequence *C / B 7| E min / F# 7 | B maj7*, using the training phrases in *minor*, *major* and *seventh* in several keys. The NSF cases and the segments reused are indicated. The phrase produces a perfect sensation of continuity, and the harmony is locally satisfied. Other examples can be found on the accompanying web site.[1]

[1] http://www.csl.sony.fr/Virtuosity.

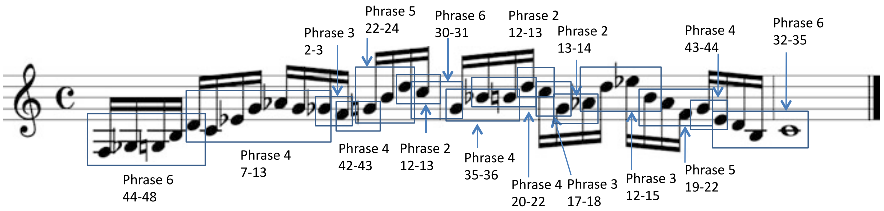


Fig. 5.20 A phrase generated on a C minor chord sequence. The compositions of the segments of the training phrases are indicated by overlapping boxes. Segments of length 2 to 7 are being used in this case. Training phrases #2 to #6 have been used. No NSF case is encountered

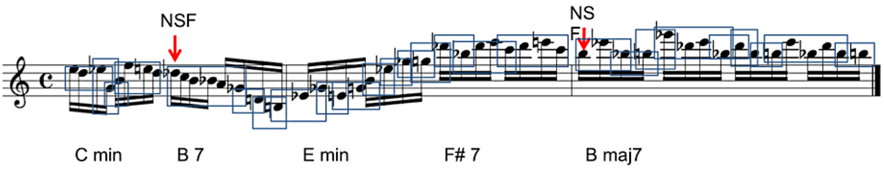


Fig. 5.21 A phrase generated on the sequence *C min / B 7 | E min / F# 7 | B maj7*. Two NSF cases were encountered, indicated by an *arrow* (and non-overlapping boxes): for the *C min* → *B7* transition, and for the *F#7* → *B maj7* one. They were addressed using the one-step-max theorem. The discontinuity is not musically perceptible by the author

5.5 Escaping Markovian Boredom

Once we have established the basis for generating melodies that comply with the rules of the game, we can now describe how to model the two important innovations that bebop has introduced in jazz. These two innovations relate to boredom: producing phrases that satisfy the criteria we have described is pleasing, but may lead to boredom after repeated exposure. From a computer science perspective, we call 'Markovian Boredom' the sensation that the phrases generated all come from the same set of rules, grammar, and that, eventually, there is no hope of hearing something new, striking, outstanding. From there on, boredom follows irrevocably.

As described in the introduction, two devices have been invented by jazz musicians to escape boredom. These devices have in turn contributed to changing the style itself. We describe here how these two devices can be modelled in our Markov framework, the issues they raise technically, and how they can be addressed.

5.5.1 *Side-Slips and Formal Transforms*

The model we have introduced so far generates notes streams on arbitrary chord sequences, which are continuous and satisfy local harmonic constraints. In the examples shown here, we use a limited training material (about six phrases for major, minor and seventh, three phrases for diminished and whole-tone, used in particular for augmented chords). More scales can be added easily (to deal with rarer chords like altered, or minor diminished 5th), but adding more scales or training sequence does not improve substantially the quality of the generation.

It turns out that *playing out* can be easily integrated in our framework. As we have seen, playing out or side-slips may be considered as an excursion from the tonality to another one, followed by a smooth landing to the right tonality. More generally, we can consider side-slips as specific *formal transforms*, operating on, e.g., the last generated phrase. Formally, side-slips can be introduced as yet another case in the *GenerateBeat()* method introduced in Sect. 5.4.2:

```
GenerateBeatAsTransform(context, H, i):
        // context represents the last generated output
        return Transform(context, N)

where Transform is defined for each possible transform, e.g.:

Transform (phrase, N)
        return Transpose (phrase, N, 1) ;
```

The particular side-slip consisting in transposing the last phrase one semitone up, can simply be represented by a transform operation, taking a phrase as input and producing its transposition. Other reasonable bebop transforms include:

- Transposing a semitone, a minor third, a tritone or an octave up or down;
- Reversing then transposing a semitone up or down, as illustrated in Fig. 5.22 (4th case).

Transforms can also be used to implement many *tricks* invented by bebop improvisers, such as transposing diatonically the phrase, by taking into account the harmony of the next beat (see the Coltrane or Di Meola examples in Sect. 5.2.4).

A most important aspect of formal transforms is the landing strategy: How to come back seamlessly to the original tonality? Our Markov framework provides the landing strategy for free: transforms may produce notes which are out-of-key, but the various strategies we proposed for negotiating chord changes can be used readily to ensure a smooth return to the key. As an example, Fig. 5.22 shows a phrase generated on chord sequence composed only of *A minor* chords.

The decision to use a formal transform, again, belongs to the intentional score, i.e. is taken at the beat level. In the case of a purely automatic improvisation system, this decision can be determined by musical heuristics, such as the following:

- When a chord is used for a long time, e.g. more than 1 measure (the original reason for the introduction of side-slips in the first place);
- When a NSF case is encountered (thereby avoiding the use of a repair mechanism);
- When a *direction* is imposed (e.g., go up pitch wise) but no easy way to satisfy it is found (see Sect. 5.5.2 on control below).

It is important to stress out that transforms are grammatical constructs, and as such cannot be learned effectively using any Markov mechanism. Using phrases such

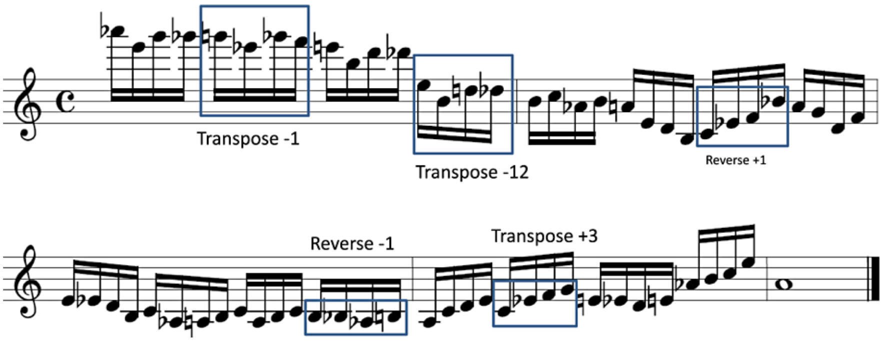


Fig. 5.22 A chorus generated on an *A minor* sequence. Formal transforms are triggered randomly during the chorus, as indicated. Note, and hear, the smooth landings on the A minor key following the transforms, in particular the striking effect produced by the third transform (reverse-1)

as the licks in 4 as training phrases for the corresponding scale (D minor) would blur the very notion of harmony, as notes from the side-slips would be considered as equivalent to notes from the original key. Furthermore, such an approach would require a tremendous amount of training data (for each possible pattern, and each possible side-slip). More importantly, it would then be impossible to trigger *intentionally* decisions to produce, or not, these side-slips.

5.5.2 The Control Issue

Above all, virtuosos can be seen as exceptional humans in the sense that they seem to exert *full control* on their production. The control issue is both the most difficult and the most interesting one to handle.

We can state the control problem as follows: how to generate a sequence that fits an arbitrary criteria, defined by target properties of the next generated phrase? In our context, such properties can be defined in terms of phrase features such as: pitch (go 'higher' or 'lower'), harmonic constraints ('more tonal notes'), intervals (chromatic), direction (ascending, descending), *arpeggiosity*, etc. Allowing such a level of control in our system is key to producing phrases with musically meaningful intentions.

The difficulty in our case comes from the fact that the generator operates on a note-by-note basis, whereas most criteria useful in practice operate on complete phrases. Let us suppose, for instance, that we want the next generated phrase to be 'higher' in pitch than the current one. It does not suffice to simply choose, at the note level, the next note with the higher pitch. Such a policy has been proposed in (Pachet 2003), and works well for simple harmonisation tasks, but not here, as we want the pitch criteria to hold on an entire next phrase. For instance, a good strategy could consist in first choosing a lower pitch and then playing an ascending arpeggio. So *longer-term planning* is needed to find satisfying, let alone optimal, phrases.

```
GenerateBeatMoreThan (context, H, I, BiasCriteria, startValue):
// context represents the last generated output
        State the generation problem as a CSP
        Compute one solution according to
        the time left that optimises the criteria
```

Fig. 5.23 The method for generating a beat according to a bias. We use the approach described in Pachet and Roy (2011). The criteria is optimised depending on the time available (using an anytime approach)

In a Markovian context, control raises a fundamentally difficult problem, because control goes in the way of the basic Markov assumption (see Sect. 5.3.2). Indeed, control consists precisely in establishing properties to be satisfied not on 'the next item' to play, but on the next sequence of items. Unfortunately, the Markovian view of sequence generation is that the future in only determined by the current state (or the N last current states, depending on the chosen order).

We have addressed this problem from a fundamental perspective, and proposed in Pachet and Roy (2011) a general solution to generate Markov sequences satisfying arbitrary properties. This solution consists in reformulating Markov generation not as a greedy algorithm, but as a *constraint satisfaction problem* (Freuder and Mackworth 1994). Constraint satisfaction is a powerful technique that enables the fast exploration of large search spaces. In our case, controlling a Markov sequence amounts to exploring the space of all possible sequences of length N (in our case, N is the number of notes per beat). This space can be huge as soon as the training set is large, or the length of the sequence to generate is high.

In this section we illustrate how controlled Markov generation can be used to influence the generation in real-time using meaningful musical criteria. Any criteria can be defined to control sequences, as long as they are computable. We present here a case in which the criteria are scalar values computed from a given sequence with simple *features*, but this scheme can be extended to more complex algorithms, classifiers in particular, as discussed below.

In a first phase, a set of melodic features are defined, such as:

- Mean pitch of a sequence;
- Mean interval of a sequence;
- Tonalness of a sequence.

Tonalness is a scalar value $\in [0, 1]$ and gives an indication of how tonal is a melody with regards to the corresponding chord in the sequence. It can be computed using, e.g. a *pitch profile algorithm* (Krumhansl 1990).

The next step is to substitute the generation of a beat by the corresponding constraint satisfaction problem, as described in (Pachet and Roy 2011).

We illustrate the mechanism as follows. We start by a phrase played on an *A minor* chord (24). We then generate three beat continuations to fill up a four beat measure (still in *A minor* in our case; changing the harmony has no impact on the control issue). We select the ones maximising criteria we consider useful as *controls*:

Fig. 5.24 The starting 4-note phrase is in the box. Here, a continuation with 'higher pitch' (*mean pitch* = 78.5 > 59.5)

Fig. 5.25 A 'lower pitch' continuation (*mean pitch* = 52.41 < 59.5) (here in the bass clef)

Fig. 5.26 A 'more chromatic' continuation (*mean interval* = 0.666 < 2.33)

Fig. 5.27 A 'less chromatic' continuation (*mean interval* = 2.33). Note the large intervals

Fig. 5.28 A 'less tonal' continuation (*tonalness* = 0.66 < 1.0). Note the Gb and Eb

higher/lower pitch, *more/less chromatic*, and *less tonal*. Figures 5.24–5.28 show continuations which optimise these five criteria, as confirmed by the values of the features. These values are compared to the initial 4-note phrase values, i.e.:

- Mean pitch: 59.5;
- Mean interval: 2.33;
- Tonalness (in the key of *A minor*): 1.0 (all notes in *A* minor).

It is important to note that this control mechanism is independent from the other mechanisms introduced here, notably formal transforms (see Fig. 5.23). Figures 5.29 and 5.30 show a combined use of transforms and controls on the chord sequence of Fig. 5.7. It can be checked that indeed the generated phrase do satisfy all the criteria.

5.5.3 Reusing Intentional Scores

The intentional score is the collection of all decisions taken for each beat. As we have seen above, these decisions concern (1) the choice of the rhythm pattern, (2) the choice of the scale (and hence, of the Markov database), (3) the decision to use and the selection of a formal transform, (4) the decision to control the generation with a specific criteria, and (5) the decision to start or stop the phrase. This score is a time

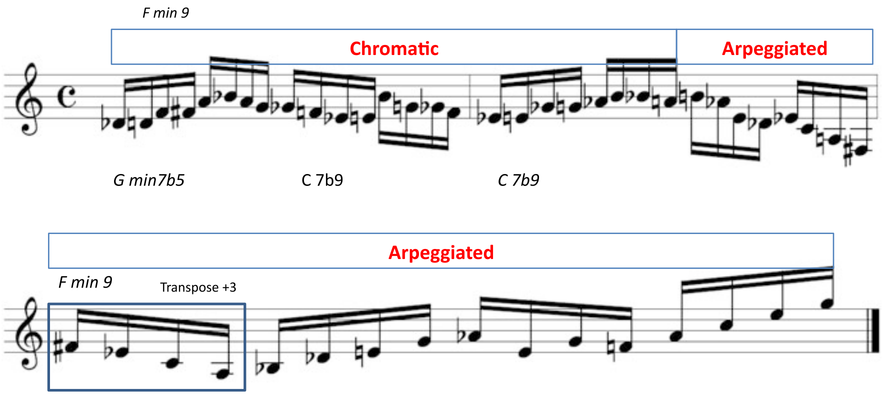


Fig. 5.29 A phrase generated with an intentional score consisting of 'chromatic' for the first six beats, and 'arpeggiated' for the next six on the same chord sequence, and one random transform. The melody generated fits almost perfectly the constraints

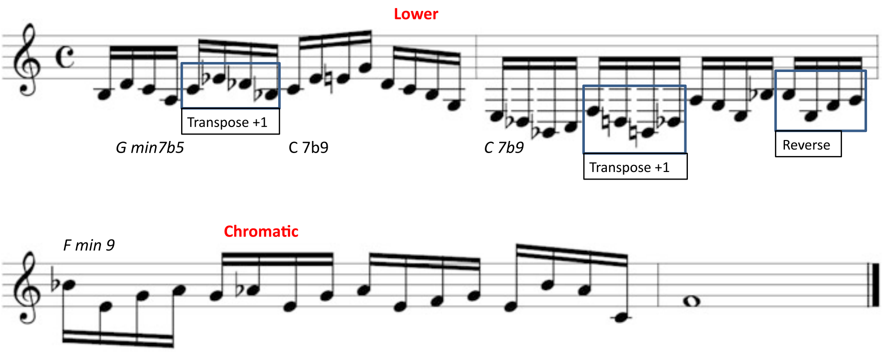


Fig. 5.30 A phrase generated on the chord sequence as Fig. 5.7, with three intentionally chosen side-slips and three subjective biases

line with commands at every beat. An improvisation can be seen as an *interpretation* of this score.

The intentional score represents the 'arbitrary' portion of chorus generation, so it cannot be generated automatically. In practice, it can be set randomly, or using an interface, e.g. gestural to produce the various commands in real time, as described in the next section.

An interesting application of concept of intentional score is to induce such an intentional score from an *existing chorus*, to generate a new improvisation with the same *structure*. We illustrate this idea using the chorus shown in Fig. 5.7. Of course, there is not a single way to infer the intentional score used by John McLaughlin. The score we consider uses solely 'target pitch' subjective biases, extracted from the actual mean pitches of the various beats in John McLaughlin's phrase. It looks as Fig. 5.31.

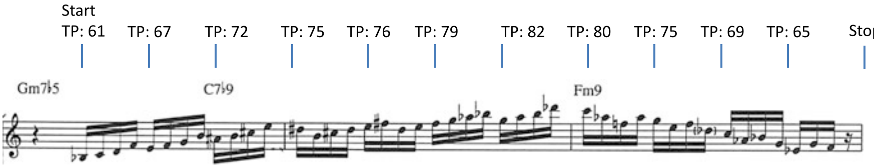


Fig. 5.31 A possible intentional score inferred from the phrase of Fig. 5.7. TP denotes the mean MIDI pitch for each beat

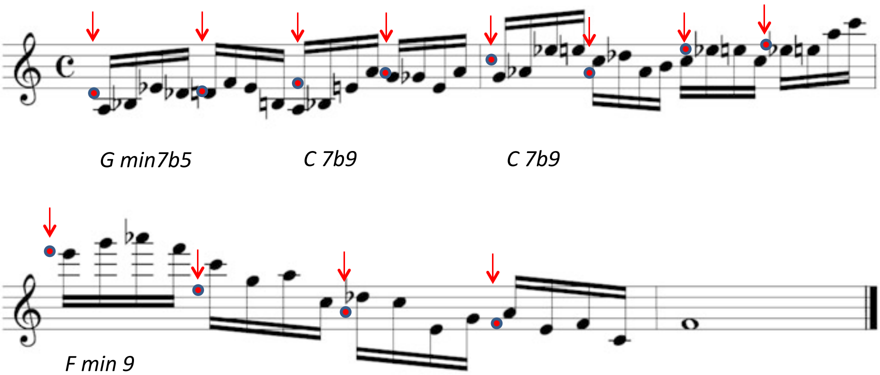


Fig. 5.32 A phrase generated on the same chord sequence as Fig. 5.7, with the intentional score induced from John McLaughlin's chorus in Fig. 5.31, consisting only of target pitches at every beat, as indicated

This score can be used to generate the phrase illustrated in Fig. 5.32. It can be seen that the resulting phrase follows approximately the same pitch contour. The phrase is not the same, as it uses only the note patterns of our training set, but it gives an indication of how to exploit intentional scores in a creative way.

5.6 Virtuoso: A Virtuoso Enabling Interactive System

Virtuoso is an interactive musical system that implements the ideas presented here so that a user can experience the sensation of being a virtuoso, without having to be one himself. *Virtuoso* is a jazz chorus generator that is controllable in real-time using arbitrary input controllers. The main features we have introduced that account, from our point of view, for a substantial part for the virtuoso aspects of jazz (side-slips and high-level control) are mapped to various gestural controls, including start, stop, speed (number of notes per beat), side-slips, as well as several criteria to control the generation as described in Sect. 5.5.2.

Several videos (Virtuoso 2011) show the author using the system, as well as intentional scores deployed during the improvisation. A number of experiments were conducted with jazz pianist Mark d'Inverno. An *a posteriori* analysis of the session by the two musicians playing is provided. Although subjective, these analysis show that a sense of *playing together* was achieved, and the music generated by the system, controlled by a human, was of professional-level quality.

5.7 Discussion

The major claim of this study is that all important decisions concerning virtuoso performance in jazz can be taken at the beat level instead of note-level. This explains how virtuosos improvise melodies satisfying so many difficult and contradictory constraints at high speed. By delegating the choice of individual notes within a beat to a non-conscious, sensory-motor level, they have enough time to focus on high-level decisions, such as influencing pitch contour, chromaticity, tonality, etc. Concerning the *memoryless assumption* hypothesised by Longuet-Higgins (see Sect. 5.2.1), we invalidate it because of side-slips, which require the memory of the last phrase played. However, the cognitive requirements remain minimal. In some sense, most of the work is done by the fingers.

Conceptually, we do not consider Markov models as representations of musical *ideas*, but as a *texture* that can be controlled to produce meaningful streams of notes. The mechanisms we propose (transforms and controls) turn this texture into realistic, sometimes spectacular, virtuoso improvisations.

Concerning the relation of virtuosity studies to creativity studies, we have stressed the importance of two important dimensions of jazz improvisation (side-slips and fine-control) that are made possible only by *extreme virtuosity*. We have shown how to model these two aspects in a Markovian context. The first one (formal transforms) does not raise any difficult modelling issues. The second one (control) does, and induces a very difficult combinatorial problem. How human virtuosos solve this problem in real-time remains a mystery. It forms important future work for virtuosity studies.

Running is not the only locomotion mode of animals. Likewise, virtuosity is not the only mode of jazz improvisation. Our system is in fact a brittle virtuoso: it knows how to run, but not so well how to walk. Such brittleness was pointed out by Lenat and Feigenbaum (1991) in the context of expert-systems and attributed to a lack of *common sense* knowledge. A *musical* common sense is indeed lacking in most automatic systems, and much remains to be done to build a completely autonomous jazz improviser exhibiting the same level of *flexibility* as humans: a competence in virtuoso mode as well as in other modes, and the ability to intentionally switch between them. *Slow improvisation*, in particular, is a most challenging mode for cognitive science and musicology, as it involves dimensions other than melody and harmony, such as timbre and expressivity which are notoriously harder to model.

However, considering melodic virtuosity as a specific mode, we claim that these automatically generated choruses are the first ones to be produced at a professional level, i.e. that only a limited set of humans, if any, can produce. A claim we leave to the appreciation of the trained listener.

More generally, this chapter is an invitation to elevate virtuosity to a *field of study* for cognitive science and computer science. Its links to creativity have only been sketched here, but they are undoubtedly deeper and yet, unexplored. Understanding virtuosity is a key to understanding creativity, in humans and with machines.

References

Addessi, A., & Pachet, F. (2005). Experiments with a musical machine: musical style replication in 3/5 year old children. *British Journal of Music Education*, *22*(1), 21–46.

Aebersold, J. (2000). *The jazz handbook*. Aebersold Jazz Inc. http://www.violistaz.com/wp-content/uploads/2009/01/ebook-guitar-the-jazz-handbook.pdf.

Assayag, G., & Dubnov, S. (2004). Using factor oracles for machine improvisation. *Soft Computing*, *8*(9).

Bäckman, K., & Dahlstedt, P. (2008). A generative representation for the evolution of jazz solos. In *EvoWorkshops 2008* (Vol. 4974, pp. 371–380). Napoli: Springer.

Baggi, D. (2001). *Capire il jazz, le strutture dello swing*. Istituto CIM della Svizzera Italiana.

Baker, D. (2000). *Bebop characteristics*. Aebersold Jazz Inc.

Bensch, S., & Hasselquist, D. (1991). Evidence for active female choice in a polygynous warbler. *Animal Behavior*, *44*, 301–311.

Biles, J. (1994). Genjam: a genetic algorithm for generating jazz solos. In *Proc. of ICMC*, Aarhus, Denmark, ICMA.

Bresin, R. (2000). *Virtual virtuosity, studies in automatic music performance*. PhD thesis, KTH, Stockholm, Sweden.

Brooks, F. P. Jr., Hopkins, A. L. Jr., Neumann, P. G., & Wright, W. V. (1957). An experiment in musical composition. *IRE Transactions on Electronic Computers*, *6*(1).

Cappellini, G., Ivanenko, Y. P., Poppele, R. E., & Lacquaniti, F. (2006). Motor patterns in human walking and running. *Journal of Neurophysiology*, *95*, 3426–3437.

Chordia, P., Sastry, A., Mallikarjuna, T., & Albin, A. (2010). Multiple viewpoints modeling of tabla sequences. In *Proc. of int. symp. on music information retrieval*, Utrecht (pp. 381–386).

Coker, J. (1984). *Jazz keyboard for pianists and non-pianists*. Van Nuys: Alfred Publishing.

Coker, J. (1997). *Complete method for improvisation* (revised ed.). Van Nuys: Alfred Publishing.

Conklin, D. (2003). Music generation from statistical models. In *Proceedings of symposium on AI and creativity in the arts and sciences* (pp. 30–35).

Conklin, D., & Witten, I. (1995). Multiple viewpoint systems for music prediction. *Journal of New Music Research*, *24*, 51–73.

Cont, A., Dubnov, S., & Assayag, G. (2007). *Anticipatory model of musical style imitation using collaborative and competitive reinforcement learning*. *LNCS* (Vol. 4520, pp. 285–306). Berlin: Springer.

Cope, D. (1996). *Experiments in musical intelligence*. Madison: A-R Editions.

Draganoiu, T. I., Nagle, L., & Kreutzer, M. (2002). Directional female preference for an exaggerated male trait in canary (serinus canaria) song. *Proceedings of the Royal Society of London B*, *269*, 2525–2531.

Ericsson, K., Krampe, R., & Tesch-Römer, C. (1993). The role of deliberate practice in the acquisition of expert performance. *Psychological Review*, *100*, 363–406.

Fitt, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, *47*(6), 381–391.

Franklin, J. A. (2006). Recurrent neural networks for music computation. *INFORMS Journal on Computing*, *18*(3), 321–338.

Freuder, E. & Mackworth, A. (Eds.) (1994). *Constraint-based reasoning*. Cambridge: MIT Press.

Gladwell, M. (2008). *Outliers, the story of success*. London: Allen Lane.

Grachten, M. (2001). Jig: jazz improvisation generator. In *Workshop on current research directions in computer music*, Audiovisual Institute-UPF (pp. 1–6).
GuitarTrio (1977). *Friday night in San Francisco, choruses by Al Di Meola, John McLaughlin and Paco De Lucia*. *Artist transcriptions series*. Milwaukee: Hal Leonard.
Hick, W. E. (1952). On the rate of gain of information. *Quarterly Journal of Experimental Psychology*, *4*, 11–26.
Hiller, L., & Isaacson, L. (1958). Musical composition with a high-speed digital computer. *Journal of the Audio Engineering Society*, *6*(3), 154–160.
Hodgson, P. W. (2006). Learning and the evolution of melodic complexity in virtuoso jazz improvisation. In *Proc. of the cognitive science society conference*, Vancouver.
Holbrook, M. B. (2009). Playing the changes on the jazz metaphor: an expanded conceptualization of music, management and marketing related themes. *Foundations and Trends in Marketing*, *2*(3–4), 185–442.
Howard, V. A. (2008). *Charm and speed virtuosity in the performing arts*. New York: Peter Lang.
Johnson-Laird, P. N. (1991). Jazz improvisation: a theory at the computational level. In P. Howell, R. West & I. Cross (Eds.), *Representing musical structure*. San Diego: Academic Press.
Johnson-Laird, P. N. (2002). How jazz musicians improvise. *Music Perception*, *19*(3), 415–442.
Keller, B., Jones, S., Thom, B., & Wolin, A. (2005). *An interactive tool for learning improvisation through composition* (Technical Report HMC-CS-2005-02). Harvey Mudd College.
Keller, R. M., & Morrison, D. R. (2007). A grammatical approach to automatic improvisation. In *Proc. SMC 07*, Lefkada, Greece.
Krumhansl, C. (1990). *Cognitive foundations of musical pitch*. New York: Oxford University Press.
Lemaire, A., & Rousseaux, F. (2009). Hypercalculia for the mind emulation. *AI & Society*, *24*(2), 191–196.
Lenat, D. B., & Feigenbaum, E. A. (1991). On the thresholds of knowledge. *Artificial Intelligence*, *47*(1–3), 185–250.
Levine, M. (1995). *The jazz theory book*. Petaluma: Sher Music Company.
London, J. (2004). *Hearing in time*. New York: Oxford University Press.
London, J. (2010). The rules of the game: cognitive constraints on musical virtuosity and musical humor. In *Course at interdisciplinary, college (IK)*, Lake Möhne, Germany.
Martino, P. (1994). *Creative force, Part II*. Miami: CPP Media/Belwin.
McCorduck, P. (1991). *AARON's code*. New York: Freeman.
McLaughlin, J. (2004). This is the way I do it. In *The ultimate guitar workshop on improvisation*, Mediastarz, Monaco. 3 DVD set.
Nierhaus, G. (2009). *Algorithmic composition, paradigms of automated music generation*. Berlin: Springer.
O'Dea, J. (2000). *Virtue or virtuosity?* Wesport: Greenwood Press.
OuLiPo (1988). *Atlas de littérature potentielle*. Gallimard: Folio/Essais.
Pachet, F. (2003). The continuator: musical interaction with style. *Journal of New Music Research*, *32*(3), 333–341.
Pachet, F., & Roy, P. (2011). Markov constraints: steerable generation of Markov sequences. *Constraints*, *16*(2).
Papadopoulos, G., & Wiggins, G. (1998). A genetic algorithm for the generation of jazz melodies. In *Proceedings of STeP'98*, Jyvaskyla, Finland.
Penesco, A. (1997). *Défense et illustration de la virtuosité*. Lyon: Presses Universitaires de Lyon.
Ramalho, G. (1997). *Un agent rationnel jouant du jazz*. PhD thesis, University of Paris 6. http://www.di.ufpe.br/~glr/Thesis/thesis-final.pdf.
Ramalho, G., & Ganascia, J.-G. (1994). Simulating creativity in jazz performance. In *Proc. of the 12th national conference on artificial intelligence, AAAI'94* (pp. 108–113). Seattle: AAAI Press.
Ramirez, R., Hazan, A., Maestre, E., & Serra, X. (2008). A genetic rule-based model of expressive performance for jazz saxophone. *Computer Music Journal*, *32*(1), 38–50.
Real (1981). *The real book*. The Real Book Press.

Ricker, R. (1997). *New concepts in linear improvisation*. Miami: Warner Bros Publications.
Ron, D., Singer, Y, & Tishby, N. (1996). The power of amnesia: learning probabilistic automata with variable memory length. *Machine Learning*, *25*(2–3), 117–149.
Shim, E. (2007). *Lennie tristano, his life in music* (p. 183). Ann Arbor: University of Michigan Press.
Sloboda, J., Davidson, J., Howe, M., & Moore, D. (1996). The role of practice in the development of performing musicians. *British Journal of Psychology*, *87*, 287–309.
Steedman, M. J. (1984). A generative grammar for jazz chord sequences. *Music Perception*, *2*(1), 52–77.
Steen, J. (2008). *Verse and virtuosity, the adaptation of Latin rhetoric in old English poetry*. Toronto: University of Toronto Press.
Stein, L. A. (1992). Resolving ambiguity in nonmonotonic inheritance hierarchies. *Artificial Intelligence*, *55*, 259–310.
Sudnow, D. (1978). *Ways of the hand*. London: Routledge & Kegan Paul.
Thom, B. (2000). Bob: an interactive improvisational music companion. In *Proc. of the fourth international conference on autonomous agents*, Barcelona, Catalonia, Spain (pp. 309–316). New York: ACM Press.
Ulrich, J. W. (1977). The analysis and synthesis of jazz by computer. In *Proc. of IJCAI* (pp. 865–872).
Valéry, P. (1948). Esquisse d'un éloge de la virtuosité. In *La table ronde* (pp. 387–392).
Van Tonder, G. J., Lyons, M. J., & Ejima, Y. (2002). Perception psychology: visual structure of a Japanese zen garden. *Nature*, *419*(6905), 359–360.
Van Veenendaal, A. (2004). Continuator plays the improvisation Turing test. http://www.csl.sony.fr/~pachet/video_vaeenendalcontinuator.html.
Virtuoso (2011). Accompanying website. www.csl.sony.fr/Virtuoso.
Voss, R. F., & Clarke, J. (1978). 1/f noise in music: music from 1/f noise. *The Journal of the Acoustical Society of America*, *63*(1), 258–261.
Walker, W. F. (1997). A computer participant in musical improvisation. In *Proc. of ACM international conference on human factors in computing systems*, Atlanta, Georgia (pp. 123–130).
Weinberg, G., Godfrey, M., Rae, A., & Rhoads, J. (2008). A real-time genetic algorithm in human-robot musical improvisation. In R. Kronland-Martinet et al. (Eds.), *LNCS: Vol. 4969*. *Proc. of 2007 computer music modeling and retrieval* (pp. 351–359).

Chapter 6
Live Algorithms: Towards Autonomous Computer Improvisers

Tim Blackwell, Oliver Bown, and Michael Young

Abstract A Live Algorithm is an autonomous machine that interacts with musicians in an improvised setting. This chapter outlines perspectives on Live Algorithm research, offering a high level view for the general reader, as well as more detailed and specialist analysis. The study of Live Algorithms is multi-disciplinary in nature, requiring insights from (at least) Music Technology, Artificial Intelligence, Cognitive Science, Musicology and Performance Studies. Some of the most important issues from these fields are considered. A modular decomposition and an associated set of wiring diagrams is offered as a practical and conceptual tool. Technical, behavioural, social and cultural contexts are considered, and some signposts for future Live Algorithm research are suggested.

6.1 Introduction

A Live Algorithm is an autonomous music system capable of human-compatible performance (Blackwell 2007, Blackwell and Young 2005). The context is improvised music; the Live Algorithm listens, reflects, selects, imagines and articulates its musical thoughts as sound in a continuous process. Or at least that is the dream of researchers working in this field. In practice, of course, the algorithm merely computes; an incoming stream of sampled sound acts as real-time input to a fixed and

T. Blackwell (✉)
Department of Computing, Goldsmiths, University of London, New Cross, London SE14 6NW, UK
e-mail: tim.blackwell@gold.ac.uk

O. Bown
Design Lab, Faculty of Architecture, Design and Planning, University of Sydney, Sydney, NSW 2006, Australia
e-mail: ollie@icarus.nu

M. Young
Department of Music, Goldsmiths, University of London, New Cross, London SE14 6NW, UK
e-mail: m.young@gold.ac.uk

J. McCormack, M. d'Inverno (eds.), *Computers and Creativity*,
DOI 10.1007/978-3-642-31727-9_6,

mechanical process that ultimately delivers an outgoing audio stream, rendered by a digital-to-analogue audio convertor.

The objective of a mechanical thinking brain is very far from realisation. The pending issue, therefore, is how a machine could emulate human performance convincingly enough that companion improvisers, and listeners, would accept the Live Algorithm as a contributing and creative group member with the same musical status as any other performer. The overarching aim is to extend the possibilities of music performance, achievable by the challenges of interacting creatively with a mechanical process, and by the exploitation of the pristine world of algorithmic patterning.

To achieve this, Live Algorithms should be able act responsively, proactively and appropriately without direct intervention, and contemporary methods in Artificial Intelligence, applied in real-time, offer suitable opportunities for rising to this challenge. The resulting systems, those that truly achieve this goal or at least achieve steps towards it, differ substantially to the norms of computer-as-instrument (with a fundamental reliance on human agency) and computer-as-score, in which a designers intentions are encoded as a set of rules or instructions, comparable to a musical score.

This chapter will examine several aspects of Live Algorithms and improvised music. In order to set the scene, a description of improvised music is given, specifying four attributes that we ascribe to human performers, and by implication, to a Live Algorithm: autonomy, novelty, participation and leadership (Sect. 6.2).

A formal specification is given that helps situate Live Algorithms in a wider taxonomy of computer music systems, based on a modular decomposition into analysis, patterning/reasoning and synthesis components (Sect. 6.3). A Live Algorithm is defined as a system in which these three elements are present, interconnected, absent of a human controller, and such that the above four attributes are satisfied. Several possible configurations of these modular elements (with or without a human controller) are considered, relating to a number of existing computer music practices. The core of the Live Algorithm, the patterning element, f, is considered in greater detail and a dynamical systems approach to their design is outlined.

The chapter continues with a discussion concerning the experience of performing with a Live Algorithm (or of hearing a Live Algorithm performance) in terms of the nature of performance interaction, and the possibilities of human-machine interaction in the near future (Sect. 6.4). Examples of simple behaviours that might be expected to occur are provided. Human performers operate in an extended context that lies beyond any single performance; they are subject to social and cultural forces that shape and inform their own approach to music. These social aspects and consequent implications for Live Algorithms are discussed.

Cumulatively, these ideas are offered as theoretical tools for the field of Live Algorithm research. Having presented them in very abstract terms, we turn to discussing a number of prototypes that we believe conform to the structure of a Live Algorithm (Sect. 6.5).

The chapter closes with an account of some possible directions for Live Algorithm research.

6.2 The Field: Creative Group Improvisation

Improvisation covers a spectrum of possibilities, from the spontaneous selection of prepared materials, which may be edited and mutated on the fly, but are made within an agreed macro-structure (for example Indian classical music and jazz), to the spontaneous creation of new micro-structures within a performance with (ideally) no previous agreements (within a genre known as "free improvisation", Bailey 1993). In the latter case, any discernible macro-structures can only emerge as the performance unwinds, since there is no single pre-defined plan or score. Other improvisational practices sit between these extremes. For example a rudimentary score might serve as a partially-defined macro-structure with opportunity for personal expression on finer levels.

6.2.1 Collective Improvisation

Free improvisation is a self-referential music. Practitioners seek to avoid any overt references to other musical genres, organisational methods or expressive devices. We propose that collective free improvisation is the ideal context for machine improvisation because the system can be considered formally, and at a first approximation, as an exchange of symbols (sonic events) between data sources (people and machines).

It has been suggested, in analogy with self-organisation in Nature, that macro-structures may emerge as a consequence of the micro-interactions between performers (Blackwell 2001), and they can do so without the agents being aware of the developing structure, or even of needing to be aware. This is exemplified by animal collectives such as vast starling flocks and herring shoals which are leaderless, decentralised and self-organising (Bonabeau et al. 1999). When group improvisation is viewed as self-organising (and this may not cover all aspects of human performance), we perceive a possibility for machine participation if the Live Algorithm can simulate inter-performer interactions.

6.2.2 The Individual (Human or Machine) in Interaction

A human performer in a collaborative improvisational context requires a basic set of capacities in order to participate appropriately. These capacities are as much to do with general aspects of the human experience and social interaction as with being a good musician. However, in improvised performance they can be specifically manifest or recognisably lacking in more or less musical terms.

6.2.2.1 Autonomy

An autonomous system, in contradistinction to an automatic system, is able to act and respond to unknowable and unforeseen inputs, and in ways that have not been completely prescribed. Autonomy is one quality that might enable a machine improviser to become accepted as an equal partner in a group setting.

The term *agent*, as used in Artificial Intelligence, refers to a device that perceives its environment through sensors, and takes action on the environment by means of actuators (Russel and Norvig 2003). An *autonomous agent* in robotics is any embodied system that satisfies its own internal and external goals by its own actions while in continuous interaction with the environment (Beer 1995). Autonomy therefore expresses the ability of an agent to take action based on its own precepts, rather than following an inbuilt plan.

An *autonomous musical agent* would therefore base its action (musical output) in part on what it perceives (musical input). The extent to which action is based on preloaded musical responses determines the degree of automation. A system that has no input is purely generative (rather than interactive). It is similar to a closed dynamical system where the initial conditions, including any hardwired knowledge, are sufficient to determine all future states. Such a system is automatic. Any further ability an automatic system might have to make a decision or take an appropriate action in the eventuality of an unplanned input would render the system autonomous, with more autonomy resulting from a greater capacity to make such decisions or actions.

6.2.2.2 Novelty

Whether supporting, leading or subverting, a musician's contribution should endeavour to avoid the clichéd and the obvious. The ability to find a novel, but also appropriate, way of playing within a very familiar musical environment would challenge any musician, but the inability to never find novelty would mark the musician down as dull, and would inhibit the ability of the group as a whole to develop communally creative structures.

Fundamental and distinct types of creativity are described by Boden (2004). A basic creative practice is to produce new combinations of established ideas or objects, just as, for example, the amateur jazz improviser combines learned melodic patterns in response to whatever harmonic structure is presented. Algorithmic emulation of such behaviour is possible as the soloist feature on the commercial *Band In A Box*[1] demonstrates; however the results are rudimentary.

6.2.2.3 Participation

An improviser has to be able to support ongoing musical activity by making contributions that do not detract from but rather enhance the current musical direction. This would be a very hard characteristic to pre-program in a top-down manner:

[1] http://www.pgmusic.com/.

how to ascertain and characterise a musical direction, and which of many possible contributions might enhance the current musical mood? However an algorithm specification does not necessarily require a top-down structure. Participatory activity should be recognisable both to human performers and listeners. The extent and character of the participation might be evident in apparent types of behaviour (some examples are discussed in Sect. 6.4.1); musical processes that allude to social modes of interaction. The wider challenges in achieving true human-machine participation are explored in later in this chapter, from musical, social and cultural perspectives.

6.2.2.4 Leadership

Another attribute of an improvising musician is the capacity to undertake a direct leadership role. In other words to attempt to change the musical direction, to invoke a new musical centre. In improvised music such roles may be fuzzy and interchangeable, and never explicitly agreed upon, but at any given time there is a balance to be struck between responsiveness and proactive intervention.

6.2.3 Relationship of the Four Attributes to Creativity

Perhaps the most familiar model of a creative process is the "exploration of a conceptual space", i.e. explorative behaviour within constraints, whether explicitly defined or not. In freely improvised group performance, it is characteristic for timbral, textural and gestural events, however individually novel, to be consistent with a shared, emerging aesthetic. This could be viewed as a participatory exploration of a musical space. In algorithmic terms, an iteration through a set of parameters or the navigation of system state to distant areas of state space can be viewed as an exploration of the potentialities of the formal computer code.

Boden's most demanding level of creativity is the notion of a transformation of conceptual space itself (Boden 2004). It is very challenging to think of any algorithmic process that could produce brand new interpretations of its own output. However, the ability to intervene proactively seems a necessary pre-condition of transformational creativity. We believe that live algorithmic music in which leadership from any party is possible offers such a prospect, i.e. to change our expectations about how humans and machines can engage in collective performance, and the consequent nature of the creative outcomes. Collective improvisation already offers a powerful approach to transformational creativity, in that the group does not possess a single shared conceptual space but a set of distinct individual conceptualisations interacting in a shared creative activity: different participants can influence these collaborators. Individual understanding of the music is a continually evolving interaction.

6.3 Theoretical Considerations

This section takes a formal approach to Live Algorithms. First, a design methodology is outlined, then it is shown how this methodology can categorise computer music systems and be applied as a conceptual tool for describing Live Algorithm behaviour. The impact of Artificial Intelligence on Live Algorithm design is considered and a dynamical systems approach is described in detail.

6.3.1 P, Q and f

Our PQf architecture, originally presented in Blackwell (2001) and described in this section, identifies three modules which may be combined in various ways to implement computer music systems. The modules are: P (listening/analysis), Q (performing/synthesis) and f (patterning, reasoning or even intuiting). The purpose is two-fold. The modules represent basic functionalities (the actual software might not be cleanly divided but the functions of conceptual parts of the system remain well defined) and their wiring diagram (Fig. 6.1) helps us to explore what systems are possible, a possible architecture for the development of any particular system and a taxonomy of established practice. Secondly, the modules represent actual software components. The development and distribution of separate modules and a language for inter-module communication would enable rapid evaluation of computer music systems, saving much effort, and encouraging novel combinations.

6.3.2 Definition of a Live Algorithm

A Live Algorithm is defined as a system in which these three modules are present, interconnected, absent of a human controller, and such that the above four characteristics (autonomy, novelty, participation and leadership) are ascribable attributes of the system.

6.3.3 Architecture

Naively speaking, P is to ears as f is to brain as Q is to voice, but in humans these compartments are themselves conceptually ambiguous. The boundaries between the modules can be reconsidered according to different ideas about perception, cognition and production (including the conceptual status of cochleas, hands and instruments). The same re-evaluation can occur in novel computer music systems. For example, in an extreme analysis $P = adc$, $Q = dac$, $f = internal\ dsp$ where adc and dac are converters between analogue (a) and digital (d) representations and dsp stands for any digital signal processing module.

There are several fundamental wirings of the three modules, with or without a human controller (Fig. 6.1), that can be used to form a taxonomy of computer music systems. The figure shows the controller placed to the left of the system (parameter domain) and the audio environment, Ψ, to the right of the system. Musicians, operating in the sonic domain (to the right of the system in the figure) contribute directly to Ψ.

P and Q are established subcomponents of music systems. The novel aspect of a Live Algorithm derives from the inclusion of a patterning/reasoning module, f, which has neither audio input or output, but is a more general purpose algorithm which could be applied equally in non-computer music contexts. In general f embodies a computational process with input and output parameter streams. In Live Algorithm terms, f is a generative unit, the machine equivalent of ideas and imagination. This function is key to enabling the system to demonstrate the capabilities of autonomy and novelty.

Each wiring is already in common use in various computer music scenarios. These are described in the following in each case, and their potential for Live Algorithms research is discussed.

P performs analysis of incoming audio (Fig. 6.1A). Its human-equivalent function is *listening*. In the figure, Ψ is the musical environment; Ψ_{in} (Ψ_{out}) are incoming (outgoing) audio streams. (Alternatively, an incoming sound wave could be digitised by an analogue-to-digital converter. Such a converter would be regarded as part of P itself.) P processes incoming samples, producing analysis *parameters*. These parameters seek to describe the audio, in music theoretic terms (events, pitch, duration), as spectral data, in timbral terms such as smoothness and roughness, or in other high level descriptors. P therefore emits a stream of parameters at a slower rate than the signal rate. In *Music Information Retrieval*, the data is used for the automatic scoring of performance. Figure 6.1A could represent a possible performance scenario in which a musician can inspect analysis parameters in real-time, most likely via a graphic display. This set-up may be used to supplement the sonic information the musician already has access to. Figuratively, the musician (functioning as a controller) is placed to the left of the P module to emphasise that system interaction is via parameters, and not by audio (in which case the musician would be placed to the right of Ψ). Reliable algorithms for machine listening are of considerable importance but the problem is very challenging when the audio source is a combination of several instruments. Machine listening is the subject of a large research effort within the DSP community.

P itself does not perform any function other than analysis. If some further purpose is intended, the analysis parameters are fed into an algorithmic module, f, as depicted in Fig. 6.1B. For example, if the information is to be used to access similar excerpts from a music database, f would perform the similarity measure and the look-up.

Note that links between modules are not directional to indicate that parameters might be passed in either direction. For example, a subcomponent of f might require a finer level of analysis from P, and could therefore send an instruction to P to that effect. The bi-directionality of system components means that the division into P,

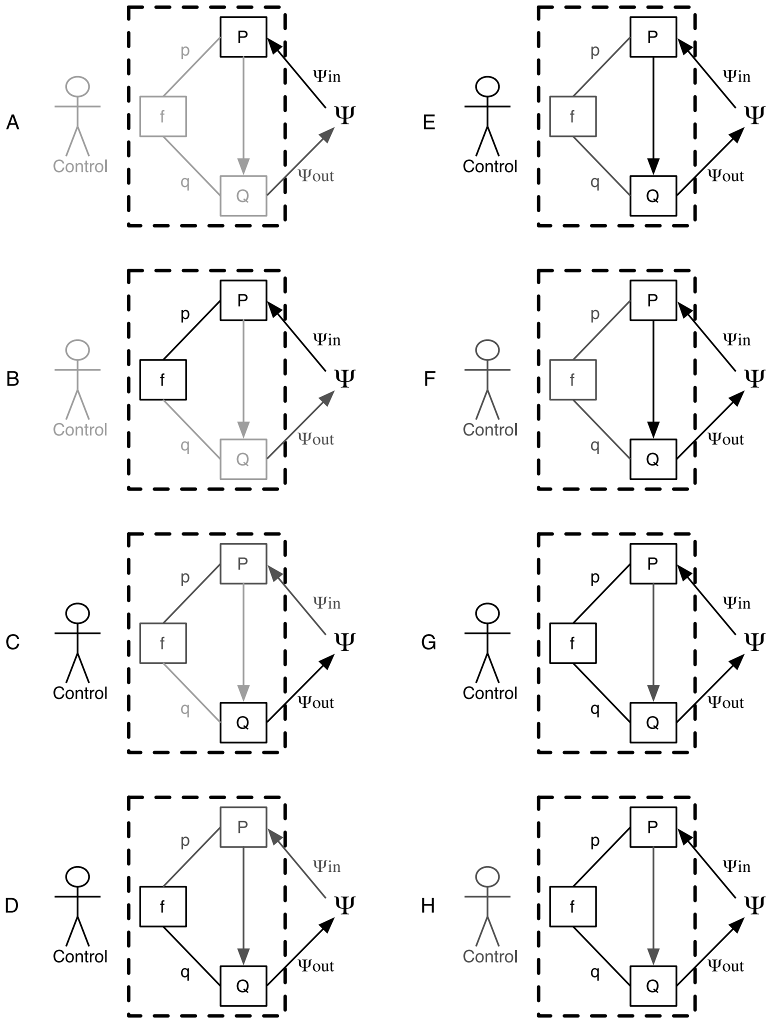


Fig. 6.1 PfQ "wiring diagrams" for different computer music applications, a non-exhaustive set of possibilities. An optional human software controller is depicted to the left of the modular decomposition; the shared audio environment, denoted Ψ, and placed to the right of the system, represents all utterances from instrument musicians and other computer music systems. The diagram shows eight common computer music systems: **A**—Audio analysis; **B**—Audio analysis with added functionality as provided by f (e.g. real-time score generation); **C**—Audio synthesis; **D**—Generative (Algorithmic) music; **E**—Live computer music involving a musician-controller who is able to monitor and adjust the functioning of a largely automatic and reactive system (such systems have become the accepted practice in live computer music settings); **F**—Reactive system as used, for example, in Sound Installations; **G**, **H**—prototype and actual Live Algorithmic systems. The ideal, wiring **H**, runs autonomously without the presence of any human control. In practice, some human attention is often required, as depicted in **G**

f and Q is to some degree arbitrary; in practice the separation is distinct since each module serves a different fundamental purpose.

Figure 6.1C shows a synthesis unit, Q, outputting audio. *Audio synthesis* is a well studied area of computer music and there are many available techniques, ranging from the rendering of sampled sound and the emulation of actual instruments (e.g. by physical modelling), to the design of new synthetic sounds using purely algorithmic techniques with no obvious analogue in the domain of physical instruments (for example, granular synthesis). The figure shows the possibility of synthesis control by an operator in real-time (but not sonically, since that would involve a P module). This is the archetypal computer music application, the computer-as-instrument; real-time control of an *electronic instrument* by parameter manipulation (the instrument might be controlled via a mouse and keyboard, or by a hardware device such as a MIDI[2] keyboard, a pedal or some other controller). Figure 6.1C might also represent live sound diffusion, in which a performer can alter parameters such as spatial position and volume in the playback of a pre-recorded composition.

Figure 6.1D shows the attachment of a module f to Q. f provides a stream of synthesis control parameters. This is the typical realisation of *generative/algorithmic music*, depicted here with a human controller, but equally possible without. This is the computer-as-proxy. f automatically generates musical information occupying a compositional functionality. There are many possibilities for f, and this represents a major branch of musical activity. It might be understood to represent any and all forms of rational process underlying pre-compositional musical methods, but contemporary, computational applications are most apposite. Two modern examples are Xenakis's development of computer-based stochastic techniques (Xenakis 2001), and the work of Cope (1992) who has developed generative systems that produce music by reference to a database of extant compositions.

One important source of potential f's are algorithms from complex and dynamical systems science. There is a wide and inspiring palate of patterning algorithms. Example include: cellular automata, generative grammars, L-systems, evolutionary algorithms, flock and swarm algorithms, iterated function systems (fractal generators), chaotic dynamics, real time recurrent neural networks and particle systems (see, e.g. Flake 1998, McCormack et al. 2009).

Clearly not interactive, fQ offers some potential for variation. For example, if some of the algorithm parameters (i.e. in f) are pseudo-random, or influenced by some other data stream, it could function as a device with variable and less predictable sonic output. An fQ compositional system could be used in performance in which musicians play alongside the system. In this case the interaction is severely limited, as the players role is to only to follow. This scenario was established in the now largely defunct genre of "live performer and pre-recorded tape" that developed through the 1950s to 1970s, although such practices are still evident in many commercial musical contexts. An intriguing contemporary possibility is the real-time manipulation of an algorithm (and synthesis parameters). Such a system is used by

[2]Musical Instrument Digital Interface, an interface and protocol standard for connecting electronic musical instruments and devices.

live laptop performers (e.g. *live coders*) who manipulate or even create small algorithmic units—f's—in real-time.

Figures 6.1E and 6.1F show a direct parameter mapping between analysed source and synthesiser control. If the functions P and Q are mutual inverses, a mapping between them is trivial, $Q = P^{-1}$, and the system is a (possibly distorted) musical mirror. Alternatively, the relationship between P and Q and the mapping between them may be so complex that a participating musician is challenged to find any obvious correlation. These systems can only be a vague prototype of a Live Algorithm as they are only automatic, in the sense defined earlier; the initial conditions that establish the mapping remain unaffected by context and new data. If there is direct intervention by a user, as in 6.1E, the system as certainly not autonomous. These systems cannot achieve autonomy because they are overwhelmingly reactive (for example it may not play in the absence of sound or pause in the presence of sound). Any attempt to move beyond this basic feedthrough device requires algorithms that do more than provide a huge switchboard of parameter connections, i.e. an f module (see below).

Systems E and F may be used in certain *sound installation* contexts, including situations where Ψ represents sonic and non-sonic environments (including movement, brightness, etc.). Although the system is primarily a parameter map, a musician could monitor synthesis data that may inform his/her choice of synthesis parameter settings. This scenario is the accepted practice in live computer music, and accounts for the vast majority of music produced in a live setting with computers. Typically a software development system such as Max/MSP is used to implement P and Q functionality (although they are infrequently broken down into actual software modules), with a visual display in the form of a "patch" and soft controls such as sliders and dials to allow real-time (non-sonic) interaction with the system. E might be considered as an enhanced instrument.

Systems 6.1G and 6.1H are syntheses of the basic A–F types. The most significant elements are that they incorporate both analysing and performing within the sonic domain, and establish a loop across the sonic and computational domains by incorporating a functional module f. The ideal Live Algorithm operates autonomously (system H); in practice, and until a true Live Algorithm is realised, some degree of intervention is desirable (system G). Further exploration of these challenges is presented in the sections below. Before this, we need to consider the fact that all these processes occur in real-time, and also musical time, in which sonic events have a past, present and future.

6.3.4 The Live Algorithm from the Outside

The Live Algorithm is, from the point of view of fellow performers, a black box. We consider the functionality of the system as a whole in terms of the most primitive description, the flow of data into and out from the device. Such a study points at possible performance measures that can be utilised in Live Algorithm design.

In Fig. 6.1, the analysis parameters p and q are marked alongside the appropriate links. In principle parameters could be passed in either direction, but at a simple level we may consider that f receives p as input, and emits q in a continuous process. Moments of silence between events are then represented by streams of constant parameters. The task of P is to deliver a parameter stream $\{p\}$ to f, and that of Q is the sonification of an output stream $\{q\}$. P and Q act as transducers that enable f to interact with the external environment. The process can be formally represented as

$$\begin{aligned}\Psi_{out} &= Q\big(f\big(x, P(\Psi_{in})\big)\big)\\ &\equiv F(\Psi_{in})\end{aligned}$$

where x is any internal state of f and it is assumed that $\{\Psi_{in}\}$ does not include any of the system's own outputs. F is the *observable* function of the Live Algorithm. f itself is hidden. Performers only have access to each other's F; inferences about private desires, goals, etc. in other words, performers f's, are made on the basis of these observations.

The Live Algorithm must therefore decide when and what to transmit with regard to the history of inputs and outputs, $\{\Psi_{in}\}$ and $\{\Psi_{out}\}$, the internal dynamic state x and any static parameterisation of f, P or Q (which may include data on previous performances).

The task of finding an f that might satisfy the requirements of a Live Algorithm is formidable. One way forward is to explore possibilities on an ad hoc basis (as is common) and in the lack of any formalised objective this is the only available means of development. The development of a performance measure for a Live Algorithm, however, would suggest a more systematic methodology. The search for a performance quantifier could commence by investigating human practice. If such a metric could be established, it would guide development of fs; there is even the possibility that a Live Algorithm could become its own critic and learn from experience.

Although the input streams are continuous, we may suppose for the purpose of this discussion that inputs and outputs can be coarse-grained. For example, audio samples can be grouped into a phrase of notes. The continuous streams could then be considered as sequences of discrete events; the precise definition of what might be a meaningful event, and the duration of the sequences is not important for the following analysis. The streams can then be split into past, current and future and comparisons can be made between them:

Ψ_{out} contains elements

1. of past Ψ_{in}'s (referential)
2. of current Ψ_{in} (reactive)
3. of future Ψ_{in}'s (pre-emptive)
4. of past Ψ_{out}'s (consistent)
5. of future Ψ_{out}'s (planning)
6. not found in past or current Ψ_{in}'s (independence)
7. not found in past Ψ_{out}'s (exploratory).

In order to participate in a group improvisation, the Live Algorithm has to convince the other participants that it is listening to and engaging with their contributions. As with conversation, this is achieved by the assimilation of inputs into outputs. Participation can therefore be measured by the degree of reference and reaction.

A novel contribution has to be surprising in some way to those engaged in the musical conversation. Novelty can be measured by the extent to which an output is independent (of inputs) and is self-exploratory.

The concept of leadership is very subtle: what can the system do so that the environment is persuaded into participation? In order to lead, a player has to embark on a musical direction that is picked up by the other participants. Copy cat scenarios such as $\Psi_{out} = \{A, B, A \ldots\}$; $\Psi_{in} = \{B, A, B \ldots\}$ should not be considered to involve a leader. Leadership therefore requires both pre-emption and novelty.

We would naturally regard any musical human partner as autonomous. A central problem of Live Algorithm research is to persuade human musicians that a machine partner is making valid contributions, contributions that should be respected, and not ignored or regarded as a malfunction. This is a problem of perceived autonomy. Operationally we suppose that a non-heteronomous and non-automatic system is *autonomous*, where a system with no referential or reactive elements is defined as *automatic* and a system that is fully determined by its environment, i.e. has no independent elements, is *heteronomous* (i.e. subject to external control). Autonomy is a relative term, with degrees of autonomy ranging in extent form marginal to very tight coupling with the environment. In this definition of autonomy, Ψ_{out} is not entirely determined by either the history $\{\Psi_{in}\}$ *or* the internal state x alone. An autonomous system sits between heteronomy and automation.

Bertschinger et al. (2008) point out that in order to avoid heteronomy, different actions must be possible given the same environment. In our formulation, future actions are contingent on both histories, $\{\Psi_{in}\}$ and $\{\Psi_{out}\}$, so that a specific Ψ_{in} occurring at t_1 and again at $t_2 > t_1$ would, in general, be followed by a different response Ψ_{out} since the histories $\{\Psi_{in}\}_{t_1}, \{\Psi_{out}\}_{t_1}$ and $\{\Psi_{in}\}_{t_2}, \{\Psi_{out}\}_{t_2}$ will generally differ.

A Live Algorithm that deploys stochastic methods (or at least some degree of statistical uncertainty) could also supply different actions given the same environment: completely random behaviour should not be considered autonomous (Bertschinger et al. 2008). Randomness is avoided if the Live Algorithm provides structured output, which may or may not incorporate past inputs or outputs. Although a Live Algorithm might produce randomness over one epoch $[t_1, t_2]$, an exploratory system would eventually produce structured output in order to avoid repeating epochs of randomness.[3] Exploratory behaviour, as noted above, prohibits prolonged repetitions of periods of randomness that might otherwise be counted as trivially novel.

Improvisation is not just about acting spontaneously, although this plays an important part. Output that lacks coherence and darts from idea to idea is a feature of

[3]We sidestep issues concerning the randomness, or not, of sequences of finite length.

inexperienced improvisors; more experienced improvisors exhibit some degree of coherence in their approach. Regularity, as captured by consistency and the ability to plan ahead, is measurable by the relationship of the current output to previous and future outputs.

The musical elements in question have not been defined, and a prescription of how to make the measurements has not been detailed. The process is dependent on the level of description, and it may be that several levels are needed. At the level of note events, for example, the elements are phrases, and comparisons can be made using a similarity measure based on a definition of the distance between two phrases (how closely the phrases shapes match each other, or the number of changes needed to bring the phrases into agreement, for example). Comparisons between streams are also possible using information-theoretic techniques such as mutual entropy; such methods might be important where an appropriate level cannot be defined.

6.3.5 Artificial Intelligence

Artificial Intelligence (AI) offers various schemes that may prove fertile for Live Algorithm research and strategies for developing functions f, as represented in general in the wiring diagrams above.

Reasoning can be based on a programmed rule set, or derived through training. In the former, the Live Algorithm designer has access to the vast experience of symbolic AI research that includes knowledge representation, problem solving, planning and expert systems. Within this AI framework, the problem domain is represented symbolically. The representation contains knowledge about the problem domain, and the symbols are syntactically manipulated until a solution is found. The main focus of the symbolic framework is on a suitable formal representation of the problem domain, the inclusion of domain specific knowledge and efficient algorithms.

Machine learning is another major AI framework. The learning algorithm can be based, for example, on a neural architecture or on Bayesian structures (e.g. Hidden Markov Modelling). Responses are learnt over a sequence of test cases. The focus is on the learning algorithm, the training set and the network architecture.

We can suppose that a human improviser can potentially refer to his/her own theoretic knowledge as well as her/his experiential knowledge of music making and of group improvisation in particular. It would be inhibitive to deny similar advantages to a Live Algorithm. Domain knowledge can be hard-wired into the Live Algorithm and trial performances offer ideal test cases for learning algorithms. Since the definition of the Live Algorithm only makes reference to inferred behaviour and not to any supposed mental states, the debate as to whether cognition is symbol manipulation (computationalism) or dependent on a neural architecture (connectionism), or indeed some other paradigm, is not relevant; rather, any technique can be requisitioned in order to further the overall goals of Live Algorithm research.

As an alternative approach to reasoning or learning, we mention here the *dynamical systems* framework which has already proven to be a rich source of ideas in Live Algorithm research.

Symbolic and connectionist approaches to mobile robotics have not been an unqualified success (Brooks 2009). The computational problem is the navigation of a dynamic, uncertain environment. An incredible amount of initial data would be needed in the closed system approach in order to account for all the possible inputs the robot might receive. In contrast, the open, dynamic framework has proven much more fruitful; the robot program is open, and modelled more closely on, say, how an ant might move through a forest. The similarity between the improvisational environment which will be very dynamic and uncertain, and the passage of an ant, or mobile robot, through an uncharted environment leads us to expect that the dynamical framework will be advantageous to Live Algorithm research too.

In a dynamical system, a state x evolves according to the application of a rule, $x_{t+1} = f(x_t, \alpha)$ where α stands for any rule parameterisation. The sequence $x_t, x_{t-1}, \ldots$ defines a trajectory in the space H of possible states. A closed dynamical system is one whose evolution depends only on a fixed parameter rule and on the initial state. These dynamical systems are non-interactive because any parameters are constant. The dynamical systems framework is quite comprehensive, encompassing ordinary differential equations, iterated maps, finite state machines, cellular automata and recurrent time neural networks.

Fully specified dynamical systems have a rich and well studied set of behaviours (Kaplan and Glass 1995 is an introductory text; Beer 2000 provides a very concise summary). In the long term state trajectories end on a limit set, which might be a single point or a limit cycle in which the state follows a closed loop. Stable limit sets, or attractors, have the property that nearby trajectories are drawn towards the limit set; states that are perturbed from the limit set will return. The set of all converging points is known as the basin of attraction of the attractor. In contrast, trajectories near to unstable limit sets will diverge away from the set. Infinite attracting sets with fractal structure are termed strange; trajectories that are drawn to a strange attractor will exhibit chaos. The global structure of a dynamical system consists of all limit sets and their basins of attraction and is known as a phase portrait. Phase portraits of families of dynamical systems differing only in the values of their parameters α, will not in general be identical.

An open dynamical system has time dependent parameters and therefore many phase portraits. Since smooth variation of parameters can yield topological change at bifurcation points (a stable equilibrium point can bifurcate into two or more limit points, or even into a limit cycle), the global properties of open dynamical systems are highly context dependent and system behaviour can be very rich. In a live algorithmic setting, the open dynamical system parameters derive from the analysis parameters. If H is chosen to map directly to the control space of Q, system state can be directly interpreted as a set of synthesiser parameters. Inputs p could be mapped to attractors, with the advantage that trajectories moving close to p will resemble Ψ_{in} (participation). However x may not lie in the basin of attraction of p and the trajectory might diverge form p, potentially giving rise to novelty and leadership. Small changes in input might lead to bifurcations in the phase portrait, sending a trajectory into a distant region of H, giving unexpected outputs. The ability of an open dynamical system to adapt to an unknown input marks it out as a candidate for autonomy.

The flow of p into f is the virtual counterpart of the sonic interactions that are taking place between performers. In a dynamical system, p could become a state alongside x, with the difference that the dynamics of p are driven by the outside world, whereas the dynamics of x are enacted by a map. Interaction fits naturally within the dynamical systems approach, unlike reasoning and machine learning algorithms which are normally constructed as closed systems. The variety of possible inputs would have to be specified by the designer in a rule-based system, and in a learning system, response is dependent on the comprehensiveness of the test set. The dynamical systems approach offers a more robust alternative. Finally we note that an extremely large number of alternative outputs (the size of H) can be easily implemented in a dynamical system.

6.4 Live Algorithms in Context

This section considers aspects of Live Algorithms that cannot be directly programmed. Improvisers are characterised by individual behaviours which are the result of learning and playing music in a social and cultural context. We speculate that a Live Algorithm might also participate in these contexts. The section looks at some behaviours, and then discuses the social and cultural dimensions of improvisation.

6.4.1 Live Algorithm Behaviour

Young and Bown (2010) identify four distinct behaviours that might be exhibited by a Live Algorithm: shadowing, mirroring, coupling and negotiation. These behaviours give some indication of the capacities systems in Fig. 6.1E–G would need to demonstrate.

The behaviours are expected to be emergent, rather than directly programmed. In general it is better to set overall goals and let a system develop its own behaviours in order to that accomplish these goals. A top-down approach is rigid and relies on a complete analysis of the problem; a bottom-up approach is more robust. The performance goals for a Live Algorithm are not well understood; Sect. 6.3.4 advocates the study and codification of the observed function F of human improvisors.

6.4.1.1 Shadowing

Shadowing involves a synchronous following of what the performer is doing, mapped into a different domain. Systems E–H in Fig. 6.1 could produce this, although only E or F are necessary. A pitch shifter in both the audio or MIDI domain, or any synchronous audio or MIDI effect, are very simple examples. In such cases, shadowing reduces to the configuration shown in Fig. 6.1E, with or without a human controller. The strength of shadowing lies in the fact that performer and Live Algorithm express a strong coherence, with tightly unified temporal patterning. In its simplest form, shadowing achieves strong but trivial participation, and little or no leadership, autonomy or novelty. However, even in this simple form, the appearance of coherence can have a strong effect for both performer and audience, and can contribute to the sense of autonomy of the system, and the generation of novelty through its interactive affordances. More complex forms of shadowing might involve more sophisticated musical responses such as counterpoint and harmony. A system based on rhythmic entrainment and temporal anticipation rather than an instantaneous response could achieve shadowing in a way that exhibited creativity, and the possibility to switch to a leadership role.

6.4.1.2 Mirroring

Mirroring involves some extraction of more abstract stylistic information or musical content from the performer, which is "reflected" back to the performer in novel ways. Pachet's *Continuator* system provides a highly sophisticated example of mirroring (Pachet 2004). System E in Fig. 6.1 would be the most apposite context.

In human performance mirroring is used as an explicit device or appears as a more implicit principle of building on a shared mood or theme. As with shadowing, the system predominantly takes the lead from the performer. This clearly demonstrates participation, and can contribute to a form of collaborative creativity through the opening up of new possibilities. As with shadowing, an appearance of autonomy comes with the sense that the musical output is coherent. By successfully achieving the local goal of mirroring with an unknown performer the system demonstrates a basic autonomous capacity, perhaps even implying that it "understands".

The mirroring approach is more immediately capable of leadership, but like shadowing must be enhanced by other behaviours to achieve this ends. Choices about how the mirroring is managed can lead to greater autonomy. In order to achieve leadership the mirroring must be capable of appropriate alteration to the style being reflected. Shadowing and mirroring preferably require an interaction scheme where the performer's output can be clearly distinguished from the environment, rather than where the state of the environment, consisting of the mixed output of both performer and Live Algorithm, is given as input. Naturally this can be achieved if the system is capable of distinguishing its own output from the mixed input, but this is challenging in practice.

Mirroring fits the fully fledged Live Algorithm scheme of Fig. 6.1H, where f involves the storage, analysis and retrieval of incoming feature data p.

6.4.1.3 Coupling

Coupling refers to a system's behaviour that is largely driven by its own internal generative routines, which are perturbed in various ways by information coming from the performer. This is a particular application of system G and H in Fig. 6.1. Designers can place the greatest emphasis on the design and behaviour of f, exploring the musical behaviour of diverse computational systems, possible with a flexible approach to the form of P and Q. Through such mutual influence, the performer and Live Algorithm can be seen as a coupled dynamical system, where both participants are capable of acting independently. Coupling does not prescribe a specific behaviour, and may involve aspects of mirroring and shadowing (in the latter case the coupling would be tighter), but tends to refer to a situation in which the system can clearly be left to lead (by acting more independently of the performer), possibly to the detriment of the sense of participation (in which case we can think of the algorithm as "stubborn" or unresponsive). However, a sense of participation depends on the attitude of the performer. A performer may deride shadowing and mirroring as a failure to truly participate, that is, to bring something original to the collective performance. A successful coupling-based system would demonstrate autonomy and creativity, and in doing so achieve participation.

Coupling is a practical behaviour because it is essentially trivial to achieve; it evades strict requirements about the kind of interactive behaviour the system exhibits, as long as the performer is demonstrably exerting some kind of influence over the system. This offers creatively fertile ground for what McLean and Wiggins (2010) refer to as the *bricoleur programmer*, building a Live Algorithm by musical experimentation and tinkering. It also relates to an aesthetic approach to computer music performance that favours autonomy, leadership and the potential for surprising variation (novelty and thus creativity) over participation. It allows for the introduction of various generative art behaviours into a performance context.

6.4.1.4 Negotiation

Negotiation is defined as a more sophisticated behaviour that is related to coupling but is based on aspects of human cognition. Only system H in Fig. 6.1 allows for this behaviour. A system that negotiates constructs an expectation of the collective musical output and attempts to achieve this global target by modifying its output. Since the collective musical output depends on the performer as well, negotiation, as the name suggests, may involve attempts to manipulate the behaviour of the performer, or equally, to adjust one's expectations in light of the direction of the music. As with coupling, with negotiation the Live Algorithm is understood as interacting directly with a piece of music, rather than with other individuals. More sophisticated Live Algorithms could perform acoustic source separation and use a "theory of mind" to infer individual behaviour from the environment.

Negotiation can be seen as a framework for the design of a Live Algorithm (see Young and Bown 2010), involving the interaction between an expectation and a behaviour (which contributes to the musical environment), either of which can be modified to create a better fit between them. This is harder to achieve than the other behaviours, so is less pragmatic. The challenge is to obtain a system that achieves strong collaboration through negotiation: that has distinctive expectations (leadership) and finds a way to satisfy those expectations whilst also accommodating the behaviour and expectations of the performer (participation). Achieving this balance can be seen as a well-formed creative challenge, one that requires autonomy.

Systems that are capable of successful negotiation are a goal for Live Algorithms research. A relevant question is how minimal a system can be whilst achieving a sense of negotiation in performance. Coupling can be seen as a weaker form of negotiation. Negotiation can be seen as fulfilling the traits autonomy, participation and leadership most fully. Novelty (leading to creativity) can be introduced into the expectation of the system.

Shadowing and mirroring can be seen as behaviours that attempt to offer the semblance of participation through acknowledgement of the performer's output, demonstrating the ability of the system to produce meaningful responses. Coupling and negotiation, on the other hand, can be seen as behaviours that attempt to create a sense of mutualism between performer and Live Algorithm (thus autonomy and leadership on behalf of the Live Algorithm), by imposing the reciprocal demand on the performer to satisfy some expectation in the Live Algorithm itself.

6.4.2 Agency and Live Algorithms

Music involves temporal dynamics on a number of time scales, from the waves and particles of microsound, through the patterning of rhythm, meter and melody, through movements, concerts, compilations and mixtapes, and out towards larger periods of time, those of genres, subcultures, individual lives and eras (see Chap. 7 for a similar discussion). Musical agency, the influence someone or something has on a body of music, which can be thought of in terms of the four categories presented in Sect. 6.2.2, actually applies at all of these time scales.

For sensible reasons, Live Algorithms focus on the kind of agency that is concentrated in a single performance, defined by Bown et al. (2009) as *performative agency*. But a great deal is lost about the musical process if we strictly narrow our focus to this time scale: in short, what a performer brings to a performance. For a free improvisor, we can think of the information stored in their bodily memory. For a DJ, we can include the material carried in their record box. In all cases, performing individuals bring with them a tradition, embodied in the development of a style, whether through practice, through social interaction or through the construction and configuration of their musical tools and resources (including instruments and bits of data).

It is hard to categorise exactly what is going on in terms of performative agency when you hear the remote influence of performer A in the behaviour of performer B, but it is necessary to consider this process in its broadest sense in order to correctly approach the agency of Live Algorithms. There are many channels through which the work of one individual becomes involved in the work of another, through the imitation of playing or singing styles, cover versions and remixes, the copying of instruments, effects, orchestration, and more recently through the shared use of software and audio samples.

As well as offering immediate presence on stage, then, Live Algorithms can also involve forms of musical presence occurring at this cultural time scale. The *OMax* system of Assayag et al. (2006), for example, can load style data, and can be used to generate such data through analysis. Here is a powerful new form of culturally transmissible data—style, encoded for use by a generative system—which can spread, evolve, and potentially accumulate complexity through distributed cultural interaction. In this way a system such as *OMax* offers a potential mechanism for bringing a less immediate kind of agency to Live Algorithm performance, reducing the burden of proof through mirroring, although not necessarily achieving the cognitive sophistication of human musical negotiation. In general, a medium term goal for Live Algorithms research may be to find formats in which behaviour can be abstracted and encapsulated in transferrable and modifiable forms, such as file formats that encode styles and behaviours.

Bown et al. (2009) categorise this interaction as memetic agency, an agency that applies outside of an individual performance, which complements performative agency and makes sense of it by accounting for the musical innovation that did not happen there and then on stage. Memetic agency adds an additional temporal layer to the taxonomy of systems presented in Sect. 6.3, which are focused on the timescale of performative agency, by requiring terms for the dynamical change of the elements P, Q and f, the initial conditions of each system, and the configuration of interacting elements, from one system to the next.

The term "memetic" refers loosely to numerous forms of cultural transmission. By strictly focusing on the performative agency of Live Algorithms, all aspects of memetic agency would appear to be left to the algorithm's designer or user: a human. And yet this agency can be critical to understanding a performance. At the extreme, pop singers who mime are almost completely unengaged from the act of musical performance, and yet memetic agency allows us to make sense of such performances. In musical styles such as jazz, much structure is already mapped out and can easily be hard-wired into a Live Algorithm's behaviour, and yet the musical style itself is emergent, not coming from a single human originator, but through repeated listening, copying and mutation. Software is rapidly become a part of this emergent social process. Today, hard-wiring is inevitable at some level in Live Algorithm design, and Live Algorithms designers, as creative practitioners themselves, can gauge the relevance of such factors in specific musical styles and performance contexts. There is nothing wrong with hard-wiring style into a system, and expecting it still to be creative.

However, as the origin of the term implies, memetic agency encompasses a notion of cultural change in which individual humans are not the only agents. Dawkins' original use of the term *meme* referred to a fundamental unit of cultural reproduction, comparable to the biological unit of the gene (Dawkins 1976). As contemporary evolutionary theory emphasises, human agency is predicated on the service of genetic success, and is not an intentionality in and of itself. Memes are just an equivalent hypothesised cause in service of which human behaviour can be manipulated. Individuals may aspire to achieve a more idealised intentionality in the tradition of the enlightenment, and this is a common way to view musical performance, but whether an individual had achieved such a feat would be hard to prove and to explain.

Between this memetic view of objects as passive participants, secondary agents in the terminology of Gell (1998), and their potential, through AI, to act as creative, human-like, active participants (primary agents). Live Algorithm design seeks a spectrum of agency and influence, rather than a distinct split between the human-like and the artefact-like. We expect to see this agency emerging not just on the stage but off it as well.

6.4.3 Live Algorithms as Musicians

Given that musicians are accustomed to negotiation as a form of improvised musical practice, a Live Algorithm ought to allow musicians to be themselves. There is no necessity to make direct contact through a control interface with the machine, as represented by the unattended systems F and H in Fig. 6.1; such contact might undermine the relationship, both in the eyes of observers, and in fact in any claim to machine autonomy. So, just as in the human world of performance practice, the use of additional tools, novel instruments and experimental interfaces is a matter of aesthetic choice, not practical necessity. Contact ought to be of a more profound, conceptual, nature.

Live Algorithms allow human-machine interactions that preserve the "basic agency relationships" (Godlovitch 1998) we expect in performance. These relationships are developed and expressed in the musical sound itself. Linkage between agent and result need not be absent or vestigial as can easily be the case in the complex, technical world of computer performance. Rather, sonic events operate at different semantic levels, generating both musical affect and effecting communication between players.

Performers are customarily valued for their capacity to demonstrate skill under constraints. This is true even of free improvisation, where the chief constraints relate to the higher-level aspects of group interaction already noted. Whatever is true of human performers must be also be true of Live Algorithms, at least in the imagination of other participants and observers. Arguably, a concert audience attributes value to a performance empathetically, in accordance with Husserl's transcendental concept of "intersubjectivity" (Husserl 1999). Performers are recognised as subjects who are themselves experiencing the world, so intentions and abilities can be attributed to them by the observer, and a critical experience of musicianship and technical accomplishment is experienced in proxy, that is, through empathy with them. Even if the observer cannot play a violin they can develop an empathetic reaction in observing it done, and this is arguably the foundation of the live music experience.

Collective, participatory performance should be considered as a social medium for participants and their audience alike, along with how we can expect Live Algorithms to be regarded as social beings is a matter for imaginative speculation.

In group performance we may see evidence of social "intimacy" (Reis and Shaver 2008) in the extent of evident mutual engagement, i.e. the close—albeit staged—interpersonal relations that occur between players. Intimacy in social psychology is characterised as a reciprocal, "interactional process" that develops between individuals; this is as true of music-making as any imaginable praxis. Intimacy develops when revelatory self-disclosure from one subject in turn finds validation through another's response. This is subsequently interpreted by the subject as evidence of an emergent and binding understanding with the other participant. Intimacies are evidence of psychological proximity, cohesiveness and trust (Prager 1995); trust that a partner can offer what is wanted (or if not, that they can offer what will provide benefit rather than harm). The development of trust occurs in situations that require interdependence, as when experience is shared, and activity and aims co-ordinated ('agentic' cohesiveness), or when there is an apparent need for a reciprocal exchange of information, for mutual control and a state of *quid pro quo* in order to achieve something desirable. All these are significant facets of participatory music performance.

If intimacy is learned over time, through a series of transactions and negotiations, it cannot be designed for in advance. Freely improvised music rests upon this premise as well. To situate a computer in this setting could be a grossly simplistic and anthropomorphising endeavour. But there are instances in which trust is fostered without direct social contact. On-line or computer-mediated intimacy has been studied by Parks and Floyd (1996) showing how trust develops free of non-verbal cues or immediate trust situations. Human-computer musical intimacy might occur in a similarly shared but restricted environment; i.e. the music itself, even though the respective understandings of that environment would differ entirely (Young 2010).

6.5 Prototypes

Many systems exist that aim to satisfy the goal of achieving some or all of the features listed in Sect. 6.2.2 (in general expressing "performative agency" as discussed in Sect. 6.4.2), validating their performative efficacy there and then in a performance context. The fellow performers and audience must be convinced of the autonomy, creativity, participation and leadership of the system through what it does on the stage. For this reason, a successful behaviour for a Live Algorithm is mirroring, performing in deference to the human improvising partner by deriving performance information from it.

A clear example of a mirroring algorithm is François Pachet's *Continuator* system (Pachet 2004). The *Continuator*, from which the term mirroring is borrowed, is explicitly designed to develop improvised responses to a solo performer in the style of that performer, using a Markovian analysis of the performer's input (see also Chap. 5 in this volume). The continuator works in a MIDI domain and performs on a MIDI instrument such as a synthesised piano. Pachet describes this as a tool to achieve creative flow, in which the performer has aspects of their playing style revealed to them in novel ways, as with a mirror. It is clearly participatory, can lead to novelty through interaction, and is autonomous in its capability to independently infer and reproduce style. The *OMax* system of Assayag et al. (2006) uses a similar framework of behavioural modelling, but is more geared towards the construction of improvising behaviours beyond that gathered by a performer in real-time. As such it can also exhibit leadership.

In terms of our PQf wiring diagrams, such systems are complete Live Algorithms (Fig. 6.1H) typically operating in a MIDI or other music symbolic domain: the f system operates directly on such symbolic data, in tandem with some kind of stored representation of a responsive behavioural strategy, such as a Markov model. Note that here as in other cases, the symbolic form of data flows p and q mean that f can easily be simulated in simpler virtual environments. This can be practical for training purposes.

A number of related systems provide frameworks that straddle the range of behaviours from shadowing to negotiation. Research into granular audio analysis and resynthesis offers a lower-level alternative to MIDI and introduces timbral information to an agent's perceptual world. Casey (2005) proposes a method for dissecting sequences of audio into acoustic lexemes, strings of short timbral/tonal categories. Based on this principle, Casey's *Soundspotter* system (Casey 2009) can be used to match incoming audio from one source with pre-analysed audio from another, offering rich creative potential. Schwarz's *CataRT* system uses a similar mechanism, providing a scatter plot interface to a corpus of pre-analysed audio data (Schwarz et al. 2006).

In its raw form, *Soundspotter* offers a powerful new kind of shadowing (more powerful than the MIDI domain given the kinds of timbral transformations and within-note control it allows), and can be considered more as a novel timbral effect or a creative tool than a Live Algorithm. This fits with the scheme of Fig. 6.1E. The *Soundspotter* framework, however, provides a firm foundation for more generative and interactive use, as demonstrated in *Frank* developed by Plans Casal and Morelli (2007), which introduces a generative process based on a coevolutionary algorithm, effectively introducing a novel f operating on feature data. As with MIDI data, here the data flows p and q take the form of (lower level) symbolic data (lexical, in Casey's terms, Casey 2005), meaning that there is a convenient model for embedding different f's in a stable musical context. Although *Frank* does not directly map input to output, it is able to take advantage of the shadowing nature of the *Soundspotter* system, for example by giving the impression of echoes of musical activity from the audio input. Britton's experiments with chains of feedback in *CataRT* have likewise explored the generative capabilities inherent in Schwarz's concatenative synthesis framework (Schwarz et al. 2006).

Thus whilst MIDI is a well established domain based on musical notation in the Western music tradition, timbral analysis and acoustic lexemes indicate new ways for music to be transformed into a conceptual space and then retransformed into sound. These principles of transformation are key to the formulation of a Live Algorithm, central to which is the identification and isolation of an abstract nested behavioural module, f, which enjoys some degree of transferability between contexts.

In these cases (and elsewhere, as in Fourier transform-based analysis and resynthesis) a compelling feature is that the channels of analysis (p) and synthesis (q) are equivalent, such that if f were simply bypassed $f(x, p) = p$ then the input data would be recovered in some shadowed form. In the extreme case, $Q = P^{-1}$, the input sound comes back unaltered.

The *Swarm Music* and *Swarm Granulator* systems of Blackwell (2001) and Blackwell and Young (2004) explore both MIDI and timbral domains using swarm dynamics. Incoming analysis parameters are mapped onto attractors in an internal space. A swarm of musical events explores this space, on occasion becoming drawn to attractors. The positions of swarm members are mapped directly onto synthesis parameters so that if the individuals were sitting directly on the attractors, the output would precisely mirror the input. Novelty arises from the exploration of phase space, and participation is coded in the tendency to move towards any discovered attractor. *Swarm Music/Granulator* is a direct realisation of the dynamical systems programme advocated in Sect. 6.3.5.

The behaviour of the above prototypes is not always human-like, and indicates how machine improvisers are already super-human in certain respects. Humans have an evolved capacity for vocal imitation, extended to imitation on musical instruments, but the mechanisms for perception and action are far from equivalent, the latter using learnt motor movement to make sound. An individual learns the relationship between action and perception through practice. The capacity for shadowing cannot be taken as given for humans, therefore, not only because our response times are too slow, but because we do not have inherent mechanisms to generate sound in equivalent ways to how we perceive sound. That said, the way we hear sound is deeply influenced by the salience of spoken language to us, a fact which matters in modelling human music perception, and the cognitive methods used by human to achieve this mapping may also turn out be useful to further Live Algorithms research.

A number of Live Algorithm systems also demonstrate a lack of equivalence between input and output interfaces by combining standard hard-wired audio analysis tools (P), domain-general AI techniques or dynamical systems such as neural networks, particle swarms or generative grammars (f), and bespoke hand-programmed generative systems (Q). This modular approach provides the opportunity to integrate human and machine decision-making processes by breaking down the behaviour of Live Algorithms into a set of problems that can be solved in different ways.

Lewis' celebrated *Voyager* system is an example of a system that is hand-coded with complex rule-based behaviour (Lewis 2000). The system uses standard audio analysis tools in order to render incoming audio in a MIDI-based form. *Voyager* is designed in a modular way according to Lewis' introspective detailing of improvisation behaviours. It encodes musical knowledge, acting as a proxy for Lewis' creative agency, and achieves each of our four goals through the success of this knowledge encapsulation. It achieves novelty using simple combinatoric processes.

Young's *Piano_prosthesis* and related prosthesis projects (Young 2008) demonstrate a more hybrid approach using standard analysis tools, such as IRCAM's *yin* ~

object for Max/MSP, to establish an internal representation of the musical environment, a composed generative music system, and a process of machine-learning establishing a connection between the two. A generative music system designed by Young acts as a flexible, parametrically controllable improvisation system with composed elements. A feedforward neural network is then used to learn a set of mappings from the musical environment to the parameters of the generative system in real-time as the performance is taking place. Young's systems exhibit elements of mirroring and shadowing in their generative rules, and come close to a notion of negotiation, as the system continually updates an internal model of what is happening in the music (without predetermined rules governing what is learned or when), which it can try to manipulate.

Similarly, Bown and Lexer (2006) have explored the use of recurrent neural networks that exhibit simple low-level dynamical behaviours such as repetition with variation and coordinated activity with an input. These networks can be embedded in a Live Algorithm by hand-coding connections between standard audio analysis tools and the recurrent neural network at one end, and between the recurrent neural network and a stochastic generative music system at the other end. In an extreme case, the recurrent neural network updates at sample rate, receiving the input audio signal directly, and generating the output audio signal directly from the activation of a single output node.

6.6 Further Considerations

This concluding section offers some directions for future Live Algorithm research. The list cannot be comprehensive since it is impossible to predict which route(s) will further the ultimate aims of the Live Algorithm agenda, but it is expected that these topics will play some part in the process.

6.6.1 Embodiment

Brooks (2009) and other researchers in embodied robotics have argued against the symbolic, representational AI approach to cognition, favouring instead a physically grounded framework in which robots are situated in the world (they deal with the world by perception and immediate behaviour, rather than by abstract representations and symbolic manipulation), and are embodied in the world in the sense that their actions have immediate feedback on their own sensations. The complexity of the environment is a key issue; rather than building fragile approximate models of the world, the embodied approach utilises the world itself in order to pursue a goal. The complexity of the world is (hopefully) tamed by working in and with the world, rather than by attempting to imagine and represent the world. A consequence of this is that embodied and situated systems can themselves have complex, emergent behaviour.

One concomitant implication for a Live Algorithm would be an embodiment that involves a means to play a physical instrument by movement rather than a synthetic production of sound by electronics. The task of learning to play a physical device would involve the development of a better potential to listen to the type of sounds the instrument could make. Hence sensor and actuator components would develop together in a feedback loop. Ultimately the expectation would be that the Live Algorithm would have a greater ability to make and hear complex sounds, an important aspect of human improvisation. Robots move in real space, and have some goal, even if only not to fall over. The analogy for our purposes would be movement in a sonic field. It remains to be seen what goals would be pertinent; perhaps to navigate between two timbres, or to find an interpolative rhythm.

6.6.2 Learning

It is without doubt a feature of improvisation that practitioners improve with time. It would be unreasonable to deny our Live Algorithm the chance to reflect on its own performance and find ways to improve. Consequently, the algorithm must be able to make mistakes. The definition of what a mistake might be for a Live Algorithm raises many fundamental issues.

There are many machine learning techniques that can be imported into the field, but they would all require the existence of some kind of performance metric, as discussed in Sect. 6.3.4. Some kind of objective evaluation of an improvised performance is needed. Such a measure could be developed by the analysis of human group performance. Unfortunately results in this area are lacking. An information-theoretic approach might be fruitful: organisation (and dis-organisation) can be computed over various timescales using entropy and complexity measures. The analysis would have to be then checked against human evaluation.

Ultimately we would require that the Live Algorithm becomes its own critic; should the algorithm feel shame as well as satisfaction?

6.6.3 Anticipated Criticisms

In human-machine dialogue, the human input to the machine is already a source of considerable organisation and information. Many algorithm designers exploit this information either intentionally or by accident. The algorithm ultimately feeds on the inherent musical organisation of the input stream.

In order to guard against this, tests could be set up involving groups of Live Algorithms (i.e. without human performers). If such a group could spontaneously generate new structures there would be more confidence in the ability of the algorithm to create its own patterns within the context of a machine-human dialogue. Interestingly, Miranda (2008) has demonstrated the emergence of songs from interacting robots; giving Live Algorithms the chance to interact with other artificial musicians might provoke growth in unexpected directions.

6.6.4 Cultural Embeddedness

Given the great importance of memetic agency to performance, a grand challenge for Live Algorithms is to expand into this realm of activity, through the implementation of long-term social-musical behaviour: the development of a style over numerous performances and practice, the appropriate absorption of influence, including appropriate copying of resources such as software instruments, raw musical data and audio samples. Social interaction can also be achieved through the development of a reputation and authority amongst a niche of interested individuals, an aspect of memetic agency not considered above since it involves long-term interaction with individuals that are not themselves musical producers. Only by addressing this time scale can we encapsulate the gamut of traits that we associate with human creativity. The emergence of software-mediated social networks makes this possibility more tractable, as the medium through which social musical interaction takes place becomes increasingly amenable to software-sourced agency. Software agents that trawl the pages of on-line musical networks such as MySpace (www.myspace.com) or Last.fm (www.last.fm), and distribute new musical output through their own pages, are already at work.[4]

6.6.5 A Final Note

The field is very active and many approaches are currently being followed. It is hard to guess which direction (whether on our list or not) will ultimately provide the biggest insights. Perhaps progress will be made with large hybrid systems that incorporate self-organising dynamical systems, machine learning, physicality and machine culture.

It should be stressed that the overall objective is not to imitate the practice of human improvisation. We do not need surrogate human performers. The aim is very different from an artificial accompaniment machine, a replacement bass player for example (although such devices may spin-off from Live Algorithm research), since such systems would not be capable of leadership, a fundamental property of Live Algorithms. Rather we seek artificial improvisers that can play alongside humans in a way that enhances our musical experience. We expect that Live Algorithms will give us access to an alien world of computational precision and algorithmic patterning, made accessible through the interface of real-time interaction. We also hope that the study of artificial improvisation will provide insights on the human activity.

Live Algorithms already enjoy an active musical life. The Live Algorithms for Music network[5] provides a nexus for musicians, engineers and cognitive scientists.

---

[4] For example, the *Cybraphon*: http://www.wired.com/gadgetlab/2009/07/cybraphon/.

[5] See http://www.doc.gold.ac.uk/~mas01tb/LiveAlgorithms/livealgorithms.html.

A workshop in 2009 at Goldsmiths, University of London, with a concert at the alternative London venue Café OTO, attracted a spectrum of systems which took part in a series of duets with internationally renowned improvisers.

The future of computer music is surely an exploitation of the creative potential that intelligent machines may offer, rather than the mundane speeding up of routine tasks or in menu-driven tools. Ideas that lie broadly under the umbrella of Artificial Intelligence and Artificial Life will become increasingly adopted by computer musicians and engineers.

Live Algorithms—performing near you, soon.

Acknowledgements Our thanks to all those who contributed to the Live Algorithms for Music concerts and symposia, the UK Engineering and Physical Sciences Research Council for initial network funding (grant GR/T21479/0) and the Goldsmiths Electronic Music Studios for hosting LAM concerts. Oliver Bown's research was funded by the Australian Research Council under Discovery Project grant DP0877320. We dedicate this chapter to the memory of the late Andrew Gartland-Jones, in acknowledgement of his encouragement and vision during the early days of the LAM network.

References

Assayag, G., Block, G., & Chemillier, M. (2006). Omax-ofon. In *Proceedings of sound and music computing (SMC) 2006*.

Bailey, D. (1993). *Improvisation*. Da Capo Press.

Beer, R. (2000). Dynamical approaches to cognitive science. *Trends in Cognitive Sciences*, *4*(3), 91–99.

Beer, R. D. (1995). On the dynamics of small continuous recurrent neural networks. *Adaptive Behavior*, *3*(4), 469–509.

Bertschinger, N., Olbrich, E., Ay, N., & Jost, J. (2008). Autonomy: an information theoretic perspective. *Biosystems*, *91*(2), 331–345. Modelling Autonomy.

Blackwell, T. M. (2001). *Making music with swarms*. Master's thesis, University College London.

Blackwell, T. M. (2007). Swarming and music. In E. Miranda & A. Biles (Eds.), *Evolutionary computer music*. Berlin: Springer.

Blackwell, T. M., & Young, M. (2005). Live algorithms. *Society for the Study of Artificial Intelligence and Simulation of Behaviour Quarterly*, *122*, 7.

Blackwell, T., & Young, M. (2004). Self-organised music. *Organised Sound*, *9*(2), 137–150.

Boden, M. (2004). *The creative mind: myths and mechanisms* (2nd ed.). London: Routledge.

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence*. London: Oxford University Press.

Bown, O., Eldridge, A., & McCormack, J. (2009). Understanding interaction in contemporary digital music: from instruments to behavioural objects. *Organised Sound*, *14*(02), 188–196.

Bown, O., & Lexer, S. (2006). Continuous-time recurrent neural networks for generative and interactive musical performance. In F. Rothlauf & J. Branke (Eds.), *Applications of evolutionary computing, EvoWorkshops 2006 proceedings*.

Brooks, R. A. (2009). New approaches to robotics. *Science*, *253*, 1227–1232.

Casey, M. (2005). Acoustic lexemes for organizing internet audio. *Contemporary Music Review*, *24*(6), 489–508.

Casey, M. (2009). Soundspotting: a new kind of process? In R. T. Dean (Ed.), *The Oxford handbook of computer music and digital sound culture*. London: Oxford University Press.

Cope, D. (1992). Computer modelling of musical intelligence in EMI. *Computer Music Journal*, *16*(2), 69–83.

Dawkins, R. (1976). *The selfish gene*. London: Oxford University Press.
Flake, G. W. (1998). *The computational beauty of nature: computer explorations of fractals, chaos, complex systems, and adaptation*. Cambridge: MIT Press.
Gell, A. (1998). *Art and agency: an anthropological theory*. Oxford: Clarendon Press.
Godlovitch, S. (1998). *Musical performance: a philoshophical study*. London: Routledge.
Husserl, E. (1999). *Cartesian meditations: an introduction to phenomenology*. Norwell: Kluwer Academic.
Kaplan, D., & Glass, L. (1995). *Understanding non-linear dynamics*. New York: Springer.
Lewis, G. E. (2000). Too many notes: computers, complexity and culture in voyager. *Leonardo Music Journal*, *10*, 33–39.
McCormack, J., Eldridge, A. C., Dorin, A., & McIlwain, P. (2009). Generative algorithms for making music: emergence, evolution, and ecosystems. In R. T. Dean (Ed.), *The Oxford handbook of computer music* (pp. 354–379). New York: Oxford University Press.
McLean, A., & Wiggins, G. A. (2010). Bricolage programming in the creative arts. In *22nd annual psychology of programming interest group*.
Miranda, E. R. (2008). Emergent songs by social robots. *Journal of Experimental and Theoretical Artificial Intelligence*, *20*(4), 319–334.
Pachet, F. (2004). Beyond the cybernetic jam fantasy: the continuator. *IEEE Computer Graphics and Applications*, *24*(1), 31–35.
Parks, M., & Floyd, K. (1996). Making friends in cyberspace. *Journal of Communication*, *46*(1), 80–97.
Plans Casal, D., & Morelli, D. (2007). Remembering the future: an overview of co-evolution in musical improvisation. In *Proceedings of the 2007 international computer music conference*, Copenhagen, Denmark.
Prager, K. (1995). *The psychology of intimacy*. New York: Guildford.
Reis, H., & Shaver, P. (2008). Intimacy as an interpersonal process. In S. Duck et al. (Eds.), *Handbook of personal relationships: theory, research and interventions* (pp. 367–389). Oxford: Wiley.
Russel, S., & Norvig, P. (2003). *Artificial intelligence: a modern approach*. New York: Prentice Hall.
Schwarz, D., Beller, G., Verbrug, B., & Britton, S. (2006). Real-time corpus-based concatenative synthesis with CataRT. In *Proceedings of 9th international conference on digital audio effects*.
Xenakis, I. (2001). *Formalized music: thought and mathematics in music*. Hillsdale: Pendragon Press.
Young, M. (2008). NN music: improvising with a 'living' computer'. In R. Kronland-Martinet et al. (Eds.), *Computer music modelling and retrieval: sense of sounds*. Berlin: Springer.
Young, M. (2010). Identity and intimacy in human-computer improvisation. *Leonardo Music Journal*, *20*, 97.
Young, M., & Bown, O. (2010). Clap-along: a negotiation strategy for creative musical interaction with computational systems. In *Proceedings of the first international conference on computational creativity*.

Chapter 7
The Extended Composer

Creative Reflection and Extension with Generative Tools

Daniel Jones, Andrew R. Brown, and Mark d'Inverno

Abstract This chapter focuses on interactive tools for musical composition which, through computational means, have some degree of autonomy in the creative process. This can engender two distinct benefits: *extending* our practice through new capabilities or trajectories, and *reflecting* our existing behaviour, thereby disrupting habits or tropes that are acquired over time. We examine these human-computer partnerships from a number of perspectives, providing a series of taxonomies based on a systems behavioural properties, and discuss the benefits and risks that such creative interactions can provoke.

7.1 Introduction

One of the distinguishing features of human society is our usage of tools to augment our natural capabilities. By incorporating external devices into our activities, we can render ourselves more quick, powerful, and dexterous, both mentally and physically. We are effectively extending ourselves and our practices, temporarily taking on the capabilities of our tools in a transient hybrid form (McLuhan 1964, Clark and Chalmers 1998, Latour 1994).

Recent advances in computational technology have resulted in software tools whose flexibility and autonomy goes beyond anything previous possible, to the extent that the tools themselves might be viewed as creative agents. This class of tool suggests an entirely new type of relationship, more akin to a partnership than to the causally unidirectional usage of a traditional tool.

In this chapter, we direct particular attention to how the computer can be used as a partner to augment the practice of musical composition. By "composition", we are

D. Jones (✉) · M. d'Inverno
Goldsmiths, University of London, London, UK
e-mail: daniel@jones.org.uk

M. d'Inverno
e-mail: dinverno@gold.ac.uk

A.R. Brown
Queensland Conservatorium of Music, Griffith University, Brisbane, Australia
e-mail: andrew.r.brown@griffith.edu.au

J. McCormack, M. d'Inverno (eds.), *Computers and Creativity*,
DOI 10.1007/978-3-642-31727-9_7,

talking in the traditional sense: the creation of a *static, determinate musical work*, whose value is in virtue of its musical content rather than its means of production. Though we will touch on ideas of improvisation, we wish to set aside performance, interactive artworks, and group creativity, and focus on the common situation of an individual artist, developing a body of work through computational means. We will explore the partnership with generative computational systems from a number of distinct perspectives, and outline some of the opportunities and hazards of such partnerships.

In considering the practice of composing with semi-autonomous music software systems, we wish to highlight two particular outcomes. Firstly, an interaction with such systems can serve to actively *extend* and reshape our creative behaviours in response to its own creative acts, encouraging unusual creative directions, or enabling actions which are otherwise unlikely. Secondly, by mirroring our own creative behaviours—either as a whole, in part, or through transformations—such a tool can help us *reflect* on our own stylistic habits and tropes.

Though the capacity to alter innate human practices is not exclusive to digital tools, we argue that computational methods enable more comprehensive and precise support of an artist's behaviour. The analytical, generative and adaptive features often found in these tools can offer new creative routes based on dynamic awareness of context and past history, harnessing the powerful probabilistic capabilities of the microprocessor.

These tendencies can change our relationships with tools and may reshape our creative processes. This influence is possible if we accept that creativity is influenced by experiences and opportunities, including those driven by our internal drives as well as by the network of instruments, methods and stimuli that we adopt. Taking the thesis that the *means* by which we produce an art object impacts upon its nature, it follows that amplifying the autonomy possessed by these means serves to broaden the range of objects that we can produce. By observing the successes and failures of this hybrid human-technology system, we can learn new ways of working which may otherwise not have arisen.

In the mainstay of this chapter, we examine human-agent partnerships from several perspectives, identifying a number of characteristic properties which distinguish them from their predecessors in the non-digital world. Along the way, we formulate a series of taxonomies which can be used to as a starting point to categorise different forms of creative technological partnership.

Before doing so, we will take a step back and consider some theoretical building blocks relating to tool use. We will later draw on these ideas in our discussion of digital tools and interactive music systems.

7.1.1 Thinking Through Tools

> People need new tools to work with rather than tools that 'work' for them. (Illich 1973, p. 10)

In daily life, the use of tools is second nature to us. We seamlessly conduct our goal-orientated activities via physical objects without the slightest awareness that we are doing so. So accustomed are we to the use of knife and fork, computer keyboard, can-opener and door-key, that the only times we become aware of their presence is when they malfunction and interrupt our activity (Heidegger 1977).

Through the complex mechanical and chemical mediation of biro on paper, we are able to convey structures of our thought to unseen recipients. Consider the example of a drawn diagram. Relationships between spatial and temporal elements can be relayed clearly and concisely, with reasonable expectation that the message will be successfully received. Moreover, by working through the details of the diagram on paper—through sketching, drafting, and observing the formalised realisation of our ideas—we can use the process of diagramming as a means to develop our own thoughts (Goel 1995). Yet, the role of the pen is completely invisible throughout. If we were continually distracted by the task of gripping the biro and steadily applying its nib to the paper, the task of relaying our ideas would be insurmountable.

In a well-known encounter, the physicist and Nobel laureate Richard Feynman discusses the archive of his own pen-and-paper notes and sketches. When asked about these "records", Feynman retorts:

> … it's not a *record*, not really. It's *working*. You have to work on paper and this is the paper. (Clark 2008, pp. xxv, original emphasis)

The implication here is clear. This physical transduction of ideas—through arm, hand, pen, paper, and back to the mind via our optical apparatus—is not simply a trace of what is going on in our mental hardware, but an integral part of the thinking process. The application of pen on paper cannot be considered a passive artifact but as a fundamental machinery responsible for "the shape of the flow of thoughts and ideas" (Clark 2008).

The above case is cited as an exemplar of what Andy Clark terms the "extended mind" hypothesis (Clark and Chalmers 1998). In brief, Clark argues that the adoption of pen and paper and other such "cognitive scaffolds" serves to shift the actual processes of thought outside of our brains and bodies, and that our sensorimotor interactions with can-openers and door-keys are embodied forms of thinking. We can consider ourselves as "open-ended systems—systems fully capable of including non-biological props and aids as quite literally parts of [ourselves]" (Clark 2003). Just as our mental conditioning serves to subtly affect our reactions to tasks, so too do the nuanced differences in the form and function of the physical tools through which we act.

Feynman was in good company when observing that writing could be a form of active thinking, rather than simply passive transcription. A century earlier, Nietzsche's adoption of the typewriter had impelled him to observe that "[our] writing tools are also working on our thoughts" (Kittler 1999). Something new emerges from this formulation that we will return to shortly: that the causal relationship between tool and user is fundamentally reciprocal. We live through our tools, and our tools shape our experiences. Actions with tools involve a *feedback loop*. It is our belief that feedback loops are key to the creative process (McGraw and Hofstadter 1993); the reader will observe them cropping up repeatedly throughout this chapter.

Needless to say, many eras of industrial development have provided us with a menagerie of tools far more exotic than the typewriter, biro or can-opener. We will focus on one specific example, albeit the most general-purpose example that we can currently imagine: the digital computer.

7.1.2 The Computer as Meta-tool

The traditional conception of a tool is an implement which provides us with mechanical means to carry out some task that exceeds our natural capabilities; consider unscrewing a nut, or levering open a crate. *Epistemic* tools, such as the abacus, can perform the same role in the domain of cognition (Norman 1991, Magnusson 2009). The information age has heralded a qualitatively new kind of cognitive extension, in the form of digital computing devices. Equipped with a programmable computer, and given an appropriate physical interface, we can produce a wide array of epistemic tools. The computer, therefore, is a meta-tool, a platform upon which we can build and use new forms of cognitive scaffolding.

The tools that we construct upon this platform do not themselves have to be static and single-purpose. Their functionality can adapt to new contexts—even those which have not been anticipated ahead of time. Software components can be modularised and aggregated, resulting in complex assemblages which incorporate the features of multiple sub-tools.

Moreover, we can confer upon our computational tools a degree of unpredictability—a most useful property when seeking to catalyse innovation and one less common in mechanical tools. With digital pseudo-random number generators, we can harness the power of chance processes, deploying them in targeted contexts to stimulate and provoke by providing new options and uncertainty.

With more sophisticated software, applications can respond with extended, nonlinear outputs, opening up vistas of possibility in comparison to the predictable one-to-one response of a traditional tool. Sufficiently complex computational systems can operate with autonomy, produce novelty, and make assessments about fitness to purpose, all characteristics associated with creativity (Boden 2004).

Throughout this chapter we will assume that our agents are "black boxes" (Latour 1994), closed to functional modification and analysis; for all intents and purposes these programmed devices, though capable of semi-autonomous action, can be considered as a tool.

7.1.3 Digital Partners in Creative Practice

Looking towards the sphere of modern musicianship, we have seen technology emerge at countless new loci, bringing about new functional relationships and modes of engagement (Brown 2000). Almost all of the disparate tasks involved in music-making can now be performed using a single digital device: from recording, arrangement and production, through to networked collaboration and distribution to listeners.

Lubart (2005) proposes a loose taxonomy of the roles that we can metaphorically consider a computer as playing within a creative context: as a *nanny*, taking care of routine tasks and freeing up our cognitive faculties for the real creative grist; as a *penpal*, aiding the process of communication and collaboration; as a *coach*, providing exercises and collections of related knowledge through a targeted database system; and as a *colleague*, working as a "synergistic hybrid human-computer system" to explore a conceptual space in tandem. Though some of the associative elements of the "coach" role are relevant to this discussion, we are here mostly concerned with the latter case, in which the computer is embedded within the same creative framework, co-operating to create a work through a succession of interactions, to form a *partnership* between creator and computational system (Brown 2001).

The capacity for autonomy in computational systems can allow them to operate with distinct agency in the creative process, a property described by Galanter as *generativity* (Galanter 2003). When using generative processes, the artist sets up a system with a given set of rules. These rules are then carried out by computer, human, or some other enacting process.[1] A purely generative work involves no subsequent intervention after it has been set in motion; a work with no generative elements has no capacity for autonomous action, and so requires continual intervention to operate.

The class of systems that we are interested in lies somewhere between those which are purely generative and those which must be manually performed. Such a system is *interactive*; it does not produce output which is completely predictable from an artist's input, nor does it simply follow only its internal logic. The output of such a system follows somehow from the previous marks of the artist (and, in some cases, the computational system itself), but its output is mediated through some predetermined structure or ruleset. A prominent example is François Pachet's *Continuator* (Pachet 2003), which captures the performance of a user and subsequently plays it back under some statistical transformations.

Systems capable of such creative interactions can be described as having *agency*. Philosophically, agency is typically aligned with intent, goal-based planning, and even consciousness. It is not this strong type of agency that we are attributing to generative art systems. We have in mind a broader, anthropological description of agency, closer to that provided by Gell (1998) in relation to art objects. Here, agency is attributed to anything seen to have distinct causal powers.

> Whenever an event is believed to happen because of an 'intention' lodged in the person or thing which initiates the causal sequence, that is an instance of 'agency'. (Gell 1998, p. 17)

[1] For examples, see the crystal growth of Roman Kirschner's installations, Hans Haacke's *Condensation Cube* (*1963–65*), or Céleste Boursier-Mougenot's *Untitled* (*2010*), in which zebra finches are given free reign over a gallery of amplified electric guitars.

Such a liberal definition allows agency to be attributed even to fixed, inert objects such as coins, clarinets, and cups (d'Inverno and Luck 2004)—in fact, many objects which are more inert than the class that we are interested in.

We will restrict our discussion of agency to those entities which demonstrate behaviour that can be classified as generative; that is, with the productive capacity to autonomously produce musical output. By partnering with an interactive, generative system, we enter into a form of distributed agency, incorporating multiple distinct productive drives. Yet having agency alone does not ensure aesthetic interest; for that, we need creativity. In the human-computer partnerships we are concerned with in this chapter, creativity inheres within the distributed system as a whole.

7.2 Computational Aides for Algorithmic Inspiration

There is an extensive ancestry around strategies to provoke and direct creative action. A commonplace example is the varied pursuit of *inspiration*. A dressmaker, bereft of creative direction, might browse the shelves of the haberdashery for ideas in the form of patterns, fabrics and accessories. A web designer may surf through collections of layouts or graphic images; indeed, at the time of writing, popular social bookmarking site Delicious[2] lists over 4,500,000 web pages tagged with the keyword "inspiration". Such creative foraging is so ubiquitous across the creative industries that countless published collections are available—within design, fashion, architecture and advertising—whose sole purpose is the provision of creative nourishment.

In making the switch to outside sources of inspiration such as these, we are augmenting our internal cognitive search and delegating our ideational activity to the external world. This can be considered as another case of the extended mind (Clark and Chalmers 1998)—or, rather, the extended imagination.

Many approaches, of course, demonstrate a more explicit intentionality than simply disengaged browsing. Csikszentmihalyi (1992), for example, recounts an ethnographical report of the Shushwap Native American practice of uprooting and relocating its village every 25–30 years. In doing so, they introduced novel, chaotic challenges to their living practice, ensuring a continual enrichment of cultural cycles.

More recently, the Surrealist writers sought to subvert the conscious mechanisms of decision-making by encouraging "automatic" drawing: the accumulation of pen strokes produced without rational control, whose result was claimed to express the subconscious or paranormal.

The chance operations of the Black Mountain College and the indeterminate works of the Fluxus group formally introduced aleatoric processes as a means of creative inspiration and delegation. The forefather of both schools is composer John

[2] http://www.delicious.com/.

Cage (1968), whose comprehensive engagement with chance, randomness and indeterminacy informed the work of countless members of the avant-garde (Pritchett 1993).

La Monte Young, a student of Cage's, was a key part of the early Fluxus movement. "An Anthology of Chance Operations" (Young 1963) is perhaps the paradigmatic text, collecting numerous instructional scores and "open form" pieces: those which leave significant constitutive elements open to choices made by the performer. In doing so, certain formal structures are imposed—some very loose, some very precise—which can act as catalysts or frameworks for artistic innovation.

The improvised painting of the Cobra group drew up a manifesto describing the process of "finding" a painting through its production, seeking an art which is "spontaneously directed by its own intuition" (Smith and Dean 1997, p. 108). Later, the American abstract expressionists adopted practices such as action painting, aleatoric and combinatorial techniques, thereby surrendering unmediated authorship of their works (Smith and Dean 1997, p. 109).

A broader approach is taken by Eno and Schmidt's *Oblique Strategies* cards (Eno and Schmidt 1975), which indirectly suggest escape routes from creative deadlock via koan-like prompts. Similarly, sets of lateral, discipline-agnostic "heuristics" are collected in the works of Pólya (1971) and de Bono (1992). A heuristic can be thought of as a problem-solving rule of thumb; its literal translation, as Pólya notes, means "serving to discover" (Pólya 1971, p. 113). Rather than offering a concrete, logically rigorous method, heuristics provide imprecise but plausible ways to tackle a problem. In this case, they suggest formal approaches, in the form of rhetorical questions such as "Have you seen it before?" (p. 110).

A markedly different tack was taken by the Oulipo movement, whose exercises in constraint offer new creative routes to writers—paradoxically, through *restricting* the parameters of their production (Matthews and Brotchie 2005). Similar constraints were present in the theatre of ancient Japan, whose ritualistic practices subscribed to a well-defined set of norms (Ortolani 1990). Submitting to external demands can be seen as another form of delegating artistic decisions, trading the openness of a blank slate for a more focused problem domain.

7.2.1 Computational Strategies and Algorithmic Aides

Historically, the potential for deploying computational technology in a creative context did not escape even the earliest computer scientists. Alan Turing's fascination with such ideas lead to the establishment of the field of artificial intelligence (Hodges 1985). Partly due to the limited success of artificial intelligence in developing fully autonomous computational systems, and partly because of the increased access to computing tools by artists and designers, experiments with creative partnerships between artists and computing systems began to flourish.

Early experiments in computer-aided composition are successively described by Hiller (1968), Chadabe (1984) and Ames (1987), with early experiments building on statistical methods and generate-and-test techniques using models of musical procedures. Koenig (Laske 1981) and Xenakis (2001) incorporated more thoroughgoing stochastic constituents in their composition, with scores and synthesis determined by multi-level algorithmic processes. So too did Cage in a handful of later multimedia works, including HPSCHD, a collaboration with Lejaren Hiller (Pritchett 1993, p. 159). Cornock and Edmonds (1973) describe the transformations that interactive tools were already effecting on the roles of both artist and audience, written in the terminology of "art systems" and multi-agency processes.

In the last quarter of the 20th century, increased computational power has enabled the wider use of real-time interactive systems (Rowe 1993, Winkler 1998) and generative simulation systems based on physical and biological processes (Berry and Dahlstedt 2003, Nierhaus 2009). Other major touchstones of algorithmic composition include Karlheinz Essl's *Lexikon Sonate* (1992), David Cope's *Experiments in Musical Intelligence* (1996), and George Lewis's *Voyager* (2007).[3]

Interactive tools for musical creativity have begun to make their way into popular culture in a number of forms. Brian Eno (1996) has historically championed the cause of generative music through his significant media profile, recently creating algorithmic soundtracks for games such as Electronic Arts' *Spore*.[4] The translations of his ideas to the popular iPhone and iPad formats, in interactive ambient sound apps such as *Bloom*,[5] have attracted popular attention to generative music systems and this and similar apps underscore a move toward music making with semi-autonomous music systems.

7.3 The Human-Computer Partnership: Characteristics and Categories

Interaction with a semi-autonomous music system inhabits an unfamiliar midpoint on the spectrum of creative relationships. It resides somewhere between tool usage and human collaboration, inheriting some characteristics of each and adding some of its own.

In this section, we will explore creative partnerships with generative computational systems from a number of distinct but related perspectives, with a view to a fuller appreciation of the potential opportunities and hazards that such partnerships can yield. These perspectives do not follow a strict progression, but are ordered based on an attempt to guide the reader intuitively, beginning with abstract principles and ending with issues of assessment and evaluation. To provide an overview, we briefly summarise each below, before expanding further in the following sections.

[3] For a more complete history of algorithmic composition, we refer the reader to Collins (2009).

[4] http://www.spore.com/ftl.

[5] http://www.generativemusic.com/.

- **Feedback** (7.3.1)
  In which we examine the multi-level feedback loops which characterise creativity, particularly the iterated cycle of *generation* and *evaluation*.
- **Exploration** (7.3.2)
  In which we discuss different ways that novelty and serendipity can be introduced by algorithmic means.
- **Intimacy** (7.3.3)
  In which we argue towards the need for trust and intimacy with a generative partner, and the surrounding issues of embodiment and predictability.
- **Interactivity** (7.3.4)
  In which we introduce five classes of productive dialogue that can be entered into with a computational partner: *directed, reactive, procedural, interactive* and *adaptive*.
- **Introspection** (7.3.5)
  In which we consider computational partners as a conduit for introspection, allowing us to reflect on our existing creative habits.
- **Time** (7.3.6)
  In which we review different timescales of the creative feedback loop, ranging from seconds to centuries.
- **Authorship** (7.3.7)
  In which we reflect upon issues of authorship and non-human agency, and the surrounding moral objections.
- **Value** (7.3.8)
  In which we discuss the differences and difficulties in assessing the aesthetic value of an art object produced with computational partners, and the proper evaluation of autonomous creativity tools.

Throughout this coverage, we will continue to draw on key examples from the field of algorithmic composition and interactive performance.

7.3.1 Feedback

> Already at the very beginning of the productive act, shortly after the initial motion to create, occurs the first counter motion, the initial movement of receptivity. This means: the creator controls whether what he has produced so far is good.
>
> – Paul Klee, *Pedagogical Sketchbook* (1972, p. 33)

Feedback is at the very heart of creativity, from Klee's "initial motion" to the point at which we stand back and decide that a work has reached its finished state. We oscillate back and forth between creative acts and reflection upon those acts, with each new mark, note, or theorem offering subtle potential to alter the direction of a work. This is a *feedback loop*, in which data about the past informs the events of the future. After each new brushstroke, what *was* just about to happen is *now* in the past, and will affect whatever we do next. It is this short cycle of repetition (depicted in Fig. 7.1), in which the output of one act becomes the input for the next, that constitutes feedback.

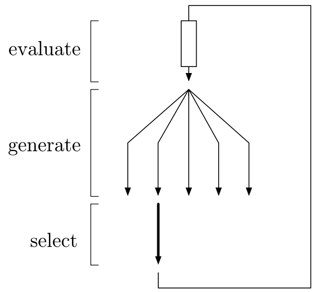


Fig. 7.1 The central feedback loop of the creative process. We iteratively *generate* creative acts, and *evaluate* how they fit into the work in its entirety

McGraw and Hofstadter (1993) describe this very cycle as the "central feedback loop of the creative process":

> Guesses must be made and their results are evaluated, then refined and evaluated again, and so on, until something satisfactory emerges in the end. (McGraw and Hofstadter 1993, p. 16)

Reducing this to its most abstract form we are left with two elements which repeat until we are satisfied with the outcome. These two elements are:

- **generation** (of the guesses that are made), and
- **evaluation** (of their results)

During the creative process composers switch from one to the other, alternating between the generation of new elements and the evaluation of the piece in its entirety.

The underlying goal of many of the computer-aided compositional strategies described above (Sect. 7.2) is to tinker with the makeup of these generate/evaluate activities, artificially expanding or warping the typical creative trajectory (Fig. 7.2). As we amplify the pool of material available for generation, we increase our creative scope. If we constrain the pool, we free up our decision-making faculties in favour of a deeper exploration of some particular conceptual subspace. Likewise, imposing a particular creative event enforces a radically new situation which demands an appropriate response, potentially introducing unanticipated new possibilities.

Generation by the computational system needs to be externalised, typically as sound or score, for our response. However, much of the human "generation" is internalised, a product of the free play of our imaginative faculties. By considering a collection of stimuli in the context of a given project, we can assess their potential to be incorporated. Disengaged browsing and creative foraging throw new (material) elements into our perception, enriching the pool of generative source material.

Imaginative stimulation is often assisted by reflective questioning. The likes of Oblique Strategies (Eno and Schmidt 1975) and Pólya's heuristics (1971) perform these types of operations as a way to provide lateral cognitive stimulus. Examples drawn from the Strategies include *Change ambiguities to specifics*; *Don't avoid what is easy*; *Remove a restriction*; and *Is it finished*?

These directives advocate a change to the parameters that we have tacitly adopted for our generation/evaluation routines. Some serve to highlight hidden, potentially

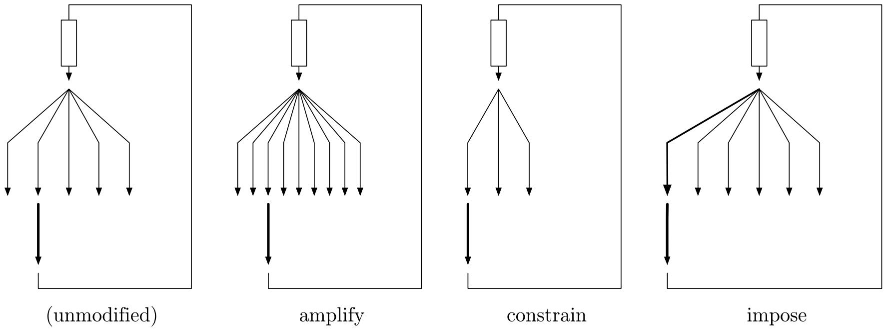


Fig. 7.2 Transforming the feedback loop using artificial methods. With generative (and even traditional) tools, we can *amplify* or *restrict* the pool of potential creative material, or *impose* a radically new direction

artificial constraints; others suggest explicitly imposing such constraints, to see what pops out.

In contrast with simple browsing, which expands the pool of creative *content*, these strategies amplify the diversity of *formal* ideas to utilise in a project. They feature analogy-based approaches, which can suggest metaphorical linkages with other domains, working on the presupposition that certain systemic structures can bear fruit when applied in a new field.

7.3.2 Exploration

Your mistake was a hidden intention

– Eno and Schmidt, *Oblique Strategies* (1975)

Let us return to the analogy of creativity as a search within a conceptual space, probing the dimly-lit peripheries of a problem for undiscovered terrain.

To take a purely logical stance, we can imagine such a search as the sequential application of deductive rules: *if* I have just played the dominant seventh, *then* I should next play the major tonic, *except* in the case that I wish to avoid immediate resolution, in which case I will play the major fourth. Many theoretical models of cognitive processes, prominently Markov chains (Wiggins et al. 2009), follow conditional rules, moving between states with probabilities that can be inferred from existing patterns of behaviour.

If we possessed such a cognitively-encoded ruleset for a given domain, and imaginative acts were simply the derivation of new consequences of these rules, it seems at first glance that the creative field would remain static and invariant. On a small scale, a fundamental source of diversity lies in relaxing or bluntly flouting these rules. Citing Thelonius Monk and his endeavours for his music "to find other places", Prévost (2004) suggests that the aspects of chance in an improvised musical performances are opportunities to make unforeseen *errors* which can subsequently

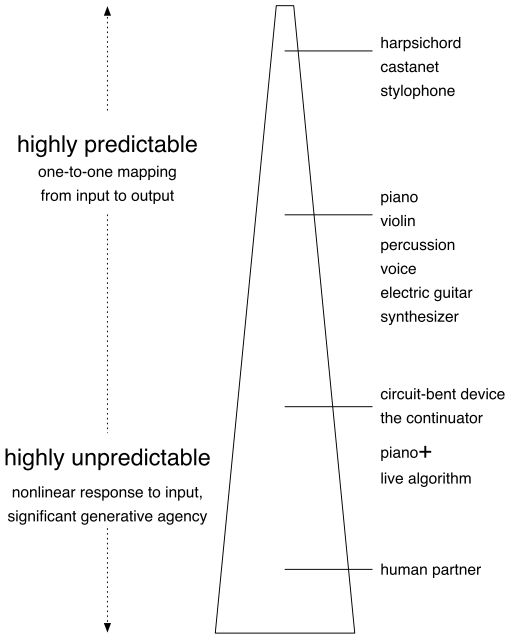


Fig. 7.3 A rough scale of predictability. Traditional acoustic instruments, giving tacit, embodied knowledge, have a one-to-one mapping from physical actions to acoustic events. On the *top* of the scale, instruments with only one degree of freedom (velocity or pitch) have limited variance. On the *bottom* of the scale, generative systems can produce an unlimited autonomous response to action, resulting in a more opaque experience. Human collaborators, though sometimes predictable, are capable of effecting a radical transformation of the interaction rules

be followed and investigated. He recounts a tale in which Monk, frustrated with an improvised performance, complained that he had "made all the wrong mistakes" (Prévost 2004)—indicating the existence and appeal of *correct mistakes*, which may aid us in this creative search.

To follow this path intentionally, then, we are effectively designing for *serendipity* (André et al. 2009, van Andel 1994): tacitly encouraging or inducing "correct mistakes" as a route to unforeseen discoveries and new creative terrain; introducing "disorder", as John Cage ordained (Cage 1968). One piece of generative music software that exploits these characteristics is Intermorphic's *Noatikl*.[6] This system relies heavily on constrained stochastic choice in selecting musical values and is advertised as a tool to "generate new ideas" and "break composer's block" (Cole and Cole 2008), providing an explicit use of aleatoric processes as a way of developing unexpected alternatives and jolting composers out of familiar habits and patterns.

A given creative act can generate a class of output along a scale of predictability as illustrated in Fig. 7.3. We may have complete, trained control over our actions, or we may surrender some control to chance. This surrender may be accidental (we slip and stumble) or intentional (we may use automatic writing, heavy air notes, or chance processes). Both these kinds of accident—intentional accidents and accidental accidents—can be retrospectively incorporated into the work.

[6] http://intermorphic.com/tools/noatikl/.

To clarify further, we shall take a look at some examples. Native Instruments' *Absynth*[7] is a virtual synthesiser, with scores of user-adjustable parameters to control a range of synthesis techniques. Alongside these determinate controls, Absynth has a feature called 'mutate'. When triggered, this nudges its parameters in random directions. Given the complex web of relationships between parameters, the output can thus be wildly unpredictable, whilst retaining a link to the previous settings. This may prompt the user to make further adjustments or suggest new sonic directions, purely through chance discoveries.

The tabletop *reacTable* (Jordà et al. 2007) device likewise has a *reacTogon* instrument which uses chance processes in hands-on interaction. Sequences of events are generated by nodes on a hexagonal grid, which collide and intersect to create unpredictable chain reactions, generating note sequences which could not be anticipated ahead of time. Effectively, we are exploring the space of *interactions* with a partner system, making use of its inherent scope for serendipity.

The fundamental benefit of these systems is that they can push us into new forms of creative adventure, by augmenting both the generative and evaluative aspects of the central creative loop. By introducing processes from outside the canon of traditional musical practice we are injecting innovation which may not have occurred through incremental, exploratory development. Such processes can generate new fragments of material that can be assimilated and modified by the artist.

In an interview, Björk Gudmundsdottir recounts an anecdote regarding composer Karlheinz Stockhausen and his everyday pursuit of the unfamiliar.

> Stockhausen told me about the house he built himself in the forest and lived in for ten years. It's made from hexagonal pieces of glass and no two rooms are the same, so they are all irregular. It's built out of angles that are reflective and it's full of spotlights. The forest becomes mirrored inside the house. He was explaining to me how, even after ten years, there would still be moments when he didn't know where he was, and he said it with wonder in his eyes. And I said, "That's brilliant: you can be innocent even in your own home", and he replied, "Not only innocent, but curious." (Gudmundsdottir 1996)

We experience a similar effect when we switch to a non-standard interface for composition. From experience, the first interactions with a system such as the *reacTogon* or McCormack's *Nodal* (McCormack et al. 2008) give rise to a creative play which pushes the user towards unfamiliar terrain. By overcoming the habits formed when repeatedly using a given interface or mode of creative operation, our curiosity and openness are restored.

In all of these cases, the "central loop of the creative process" (McGraw and Hofstadter 1993) is being widened to incorporate agencies which are not present in what may be considered "normative" creativity. The Romantic conception of an isolated painter, toiling for weeks over a canvas in visual engagement with his subject, makes way for a hybrid, collective creative intelligence, whose output is the result of an internal tussle between heterogeneous and nonaligned forces.

[7] http://www.native-instruments.com/en/products/producer/absynth-5/.

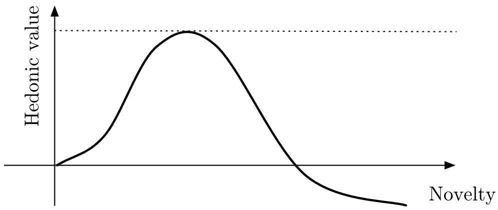


Fig. 7.4 The Wundt curve. As novelty increases, gratification rises to a peak, falling again as we move towards more extreme unfamiliarity

7.3.3 Intimacy

To enter into a meaningful and enduring relationship with a tool or creative partner, we must secure a degree of trust in it: trust that its responses will have some relevant correlation with our own, rather than it disregarding our inputs and behaving completely autonomously; trust that we can gain an increasing understanding of its behaviour over time, in order to learn and improve our interaction with it, either through embodied (tacit, physical) or hermeneutic (explicit, neural) knowledge; and, in the case of computational or human partners, trust that its activity will continue to generate interest through autonomous creative exploration. In other words, the output of such a system should be novel, but not too novel; as represented by the Wundt curve shown in Fig. 7.4.

Creative interaction with generative systems is often premised on a duality, wherein the computational system generates material and the human acts as a fitness function, selecting or rejecting materials and arranging them into a final product. This would be a tiresome process if the generated material varied widely from what was required. Consistency of operation also improves the confidence of an artist in the output of a generative system. Confidence and predictability in the system contribute to the development of a partnership and, ultimately, to the productivity and quality of the work.

Predictability aside, it is clear that all designed artifacts, including generative systems, are biased by decisions made by their developers and by the materials and processes they use. We must align our thinking with the patterns and prescribed methods that underlie the design thinking of the system (Brown 2001). Understanding these patterns is necessary to get the best out of the system.

For an effective partnership with a computational tool, we suggest that it is necessary to accept such biases as characteristics, rather than errors to be fighting against. Again, taking the analogy of a traditional musical instrument, good musicians learn to work within the range of pitch, dynamics and polyphony of their instrument as they develop their expressive capability with it.

A quite different difficulty lies in the material status of our tools. Magnusson (2009) argues that acoustic and digital instruments should be treated with categorical difference, with implications for our ontological view of their interfaces. The core of an acoustic instrument, he argues, lies in our embodied interaction with it, realised through tacit non-conceptual knowledge built up through physical experience. A digital instrument, conversely, should be understood hermeneutically, with its core lying in its inner symbolic architecture. Tangible user interfaces are “but arbitrary peripherals of the instruments’ core” (Magnusson 2009, p. 1).

This implies that our interactive habits are developed quite differently with a digital tool. When playing an acoustic instrument, we can typically offload a large amount of cognitive work into muscle memory, which, with practice, can handle common tasks such as locating consonant notes and moving between timbres. An alternative to this development of embodied habituation for computational systems is the use of automation and macros that can capture repeated processes and actions.

This type of process encapsulation is inherent to many generative computer composition systems including *Max/MSP*,[8] *Supercollider*,[9] *Impromptu*[10] and so on. The hierarchical arrangement of motifs or sections that this type of encapsulation allows is well suited to music compositional practices. These come together in an interesting way in the software program *Nodal*,[11] in which generative note sequences and cycles can be depicted as graphs of musical events (nodes). Nodal allows for the creation of any number of musical graphs and for the user to interact with them dynamically. The behaviour of individual nodes can be highly specific, providing confidence in the exact detail of music generated, while musical fragments and riffs can be set up as independent graphs that "capture" a musical idea. However, despite this level of control and encapsulation, the interactions between nodes and graphs can give rise to surprisingly complex and engaging outcomes.

7.3.4 Interactivity

One of the affordances of computational systems is the shift from the traditional interactive paradigm, in which one action results in one musical response, to "hyperinstruments", which can respond to actions with multiple, structured events. This can be seen as meta-level composition or performance, described by Dean as "hyperimprovisation" (Dean 2003), where a computational improvisatory partner does more than react to human responses.

McCullough (1996) advises that dynamic control over high level operations rather than low level details yields a sense of control over a complete process in tool usage generally. This kind of meta-control is typical of manipulating generative processes. Beilhartz and Ferguson (2007) argue that the experience of connection and control for generative music systems is critical; "The significance of generative processes in an interactive music system are their capability for producing both a responsive, strict relationship between gesture and its auditory mapping while developing an evolving artifact that is neither repetitive nor predictable, harnessing the creative potential of emergent structures" (Beilhartz and Ferguson 2007, p. 214).

As a consequence of the more structured possibilities for tool-use relationships, many different kinds of control flow exist within computational creative tools

[8] http://cycling74.com/products/maxmspjitter/.

[9] http://supercollider.sourceforge.net/.

[10] http://impromptu.moso.com.au/.

[11] http://www.csse.monash.edu.au/cema/nodal/.

Fig. 7.5 Example of a drawing produced with Ze Frank's reactive v_draw system (zefrank.com/v_draw_beta). The volume level of sounds produced by the user is translated into lines on screen: quiet noises turn the line anticlockwise, loud noises turn the line clockwise

(Fig. 7.6). Awareness of these and how they might be combined within or across a generative system is an important step toward a better understanding of the range of creative relationships that are possible.

A *directed* tool is the classical form of computational application: controlled through a typical HCI device (mouse, keyboard, touchscreen), these are used to mediate creative acts onto a screen or printing device. The user exercises control over the outcome of their actions, which is produced (effectively) immediately. Typical examples are desktop applications for graphics, musical composition or word processing, such as Adobe Photoshop and Sibelius. Such a tool should operate predictably and readily learnable.

A *reactive* tool senses a user's creative acts, through a microphone, camera or other sensor, and responds proportionately—often in another sensory domain. A commonplace example is the real-time visualisation of music, as exemplified by the likes of Apple's iTunes media player. No expectation is produced for further development within the aesthetic narrative, though the user may be able to learn and master the mode of interaction.

Other examples of reactive tools include Ze Frank's *v_draw*[12] web application, which maps sound volume levels into drawn lines (see Fig. 7.5). Amit Pitaru's *Sonic Wire Sculptor*[13] performs the same operation in the other direction, transforming drawn 3-D structures into looping sound.

A *procedural* system involves a fixed process, typically designed by the user, which unfolds gradually when triggered. Examples include the phasing techniques used by Steve Reich, Iannis Xenakis' physical simulations of particle clouds, and the plant growth models of Lindenmayer systems (McCormack 1996). Though some indeterminate elements may be present, and a seed configuration may be input by the user (as in the case of L-systems), no subsequent intervention is required or expected.

[12]http://www.zefrank.com/v_draw_beta/.

[13]http://pitaru.com/sws/.

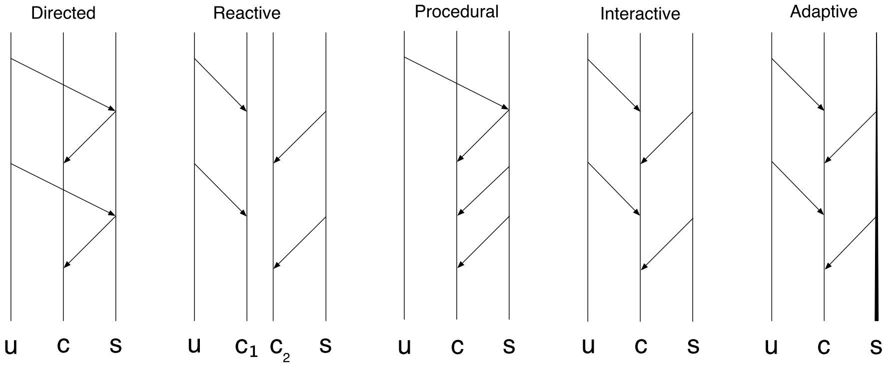


Fig. 7.6 Types of interactive dialogue. *u* is the user or artist; *c* is the "canvas" or constructive space; *s* is the computational system, which when adaptive changes its behaviour over time

An *interactive* system, conversely, tracks its users actions and responds to them within the same "canvas", creating the potential for further development upon the system's actions. This canvas may be an acoustic space, the virtual page of a word processor, or even a physical sheet of paper. It then becomes possible to respond to the system's output, potentially reshaping *its* direction as well as our own. The outcome contains elements of both the system and user and attribution to each becomes blurred. An example is the MetaScore system (Hedemann et al. 2008) for semi-automatic generation of film music via control of a generative music system's parametric envelopes.

An *adaptive* system extends beyond the interactive by developing its behaviour over a time period. These systems change the dynamics of their responses according to the history of observations or behaviours. This introduces a behavioural plasticity which allows its activity to remain relevant and novel to its user. Tools falling into this class often make use of dynamical systems such as neural nets (Miranda and Matthias 2005, Bown and Lexer 2006, Jones et al. 2009), evolutionary algorithms (Brown 2002) and ecosystems (McCormack 2003, Jones 2008; see also Chap. 2 in this volume).

7.3.5 Introspection

Early theorists of computer music—partly, no doubt, as a consequence of the technological limitations of the era—placed emphasis on the purification of the compositional process as a way of better understanding our own behaviours, either personal or cultural (Supper 2001, Hiller 1968, Ames 1987). To model the processes that tacitly underlie a existing musical system, we must first formalise them in an effectively computable form; i.e. transform them into a set of algorithms, with which we can generate new pieces that fall into the same class. By creating a computer program which executes these algorithms, we are therefore exploring the range of works within this class, which can enhance our understanding of their properties.

Besides the formal benefits offered by describing a style in an algorithmic form, this also serves to reveal selective bias within the application of these procedures. It is distinctly possible that artists fail to follow one pathway in some creative terrain due to their tendency to automatically follow a more normative path, as trodden by previous artists or by themselves on previous occasions. Like many tools, algorithmic descriptions of music are likely to emphasise existing tendencies, some of which the composer may previously been unaware of; conversely, there are many examples in the field of empirical musicology (e.g. Huron, 2006) in which algorithmic processes reveal novel patterns.

We might also create conjectural models based on emergent cognitive properties of music perception, such as those of Narmour (1990), Temperley (2007) and Woolhouse (2009). Rather than construct a descriptive system through stylistic analysis, this approach incorporates sensory capabilities such as patterns of auditory perception that exist *behind* traditional systems of musical composition—the systems beneath the systems. Such models allow us to reflect on the meta-reasoning behind whole classes of compositional style, such as the Western diatonic tradition.

We can likewise develop our insight into wider cognitive processes through computational simulation. Tresset and Leymarie's *Aikon-II*[14] creates facial sketches by observing the subject's salient features and drawing with a mechanical plotter on paper, visually perceiving the sketch as it draws. The project aims towards gaining an understanding of our own observational mechanisms by computationally implementing them, and in doing so illuminating any irregularities in the theory that may not exposed by contemplation.

The above approaches can be viewed as applied forms of cultural study, serving to illuminate historical and social tendencies on a broad scale. Following Boden's (2004) distinction between H-creativity (historical creativity, novel to an entire historical frame) and P-creativity (personal creativity, novel only to its creator), we describe this pursuit of understanding through cultural modelling as *H-introspection*.

Its counterpart is *P-introspection*, which applies to tools used to reflect and understand the user's personal creative acts. An example of P-introspection is Pachet's *Continuator* (Pachet 2003), which uses a Markov model to reflect a player's performance style through its statistical properties. The approach taken by the Continuator is what Spiegel (1981) describes as "extrapolation": the "extension beyond that which already exists in such a way as to preserve continuity with it, to project from it...". The high-level characteristics of a style are maintained, whilst creating new works "projecting" from the original.

By mirroring certain properties in such a way, the player may become attuned to features that they were not aware they exhibited, leading towards a more insightful mode of creative development.

[14] http://www.aikon-gold.com/.

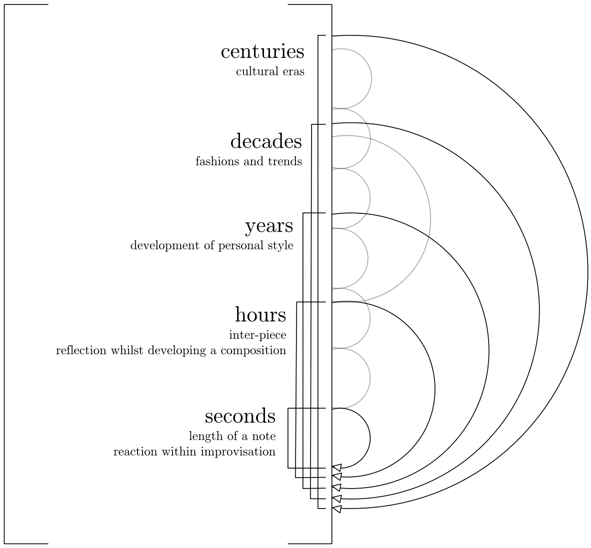


Fig. 7.7 Hierarchy of feedback timescales. Our actions at any point are a cumulative consequence of previous activity and reflection, with such reflection operating over a number of temporal levels

7.3.6 Time

A defining factor of the feedback loop between human and computational partners is the time taken for feedback to occur—that is, the period that it takes to produce and reflect upon each new act. Generation and reflection operate in a nested hierarchy, over multiple timescales (Fig. 7.7), each reflecting qualitatively different parts of the creative process.

We will briefly consider the representatives of digital technology within each of these timescale brackets: seconds and milliseconds, hours and minutes, years and months, centuries and decades. The boundaries of these temporal categories are not well defined and simply depicts a continuum from short to long time scales.

Seconds: On the micro-level, of seconds or less, an improvising musician produces sound events (notes, beats, timbres), observing their progression and relationships with the macroscopic structure of the piece in general. An error may be rapidly incorporated ("retrospectively contextualized" (Sawyer 2006)) into a performance and reclaimed as intentional, if the player possesses sufficient virtuosity.[15]

Numerous interactive pieces of software exist with which we can improvise and hone our skills. The likes of George Lewis' *Voyager* (Lewis 2000), Michael Young's *aur(or)a* (2008), and the field of live algorithms in general (see Chap. 6 in this volume) play the role of virtual partners, responding rapidly with semi-autonomy.

[15]See Pachet's discussion of bebop sideslips (Chap. 5) for a more in-depth treatment on how intentional error-like acts can be used to effectively demonstrate virtuosity.

Hours: On the scale of minutes and hours, we may develop a piece, adding phrases to sections and sections to movements. These can be replayed to observe their fit within the wider narrative.

Scaling beyond the length of a single piece of music, we have systems such as the *Continuator* (Pachet 2003), which reflects back the statistical properties of a user's musical behaviour over the length of entire phrases. The reward is that, through listening back to a distorted edition of their original patterns, the player can better understand their own habits by hearing them recontextualised.

Generative algorithms can be used to apply a similar process of segment organisation, perhaps with generated components or selections from a database. Applied in interactive composition environments, with an aesthetic fitness function provided by their human counterpart, such a process can provide an effective heuristic-based method of exploring musical possibilities (Wiggins et al. 1999).

The development of a single work is often achieved through iterated generation/evaluation with a particular interactive music system. It is also possible that an artist is able to modify the code of a music system co-evolving the work and the system. In this case a slower feedback loop can occur: the system is allowed to run, perhaps repeatedly, and its output observed (evaluation); based on this observation, the code is modified to alter or enhance the system's behaviour (generation). This process can be seen quite transparently in live coding performances, where code is written, run and modified as part of the live performance.

Years: Our personal style may develop with reference to previous works and external stimuli; a visit to a gallery may prompt a radical departure which causes us to rethink our trajectory, or consider it along a new axis. A prominent example of a system that evolved on this scale of feedback is AARON, an autonomous drawing system developed by Cohen (1995) over several years.

Developments at this scale can also be observed through data mining of musical corpus. For example, by matching musical phrases against a large corpus of recordings based on similarity measures, *Query-by-Example* (Jewell et al. 2010) enables its users to reflect on how their performances have developed over long periods—or relating them to bodies of other musicians' work. We could imagine such tools entering much more widely into the reflective practice of artists, allowing them to more closely understand their own historical lineage and their position within a wider context, potentially discovering hidden relationships with previously-unknown peers.

Decades: Over decades, cultural fashions ebb and flow. It is this temporal nature of styles which causes many works to fail to be accepted often for many decades. Punk, new-wave and dance music are all examples of cultural fashions in UK music for example.

Centuries: At the timescale of entire eras, we can interrogate historical tendencies through tools designed for H-introspection (Sect. 7.3.5). The work of empirical musicologists have laid some groundwork for computational analysis of trends at this scale, while musical models such as those of Cope (1996) study cultural movements by encoding them algorithmically and playing out their consequences. Insights from these approaches can aid us to better understand the mechanisms underlying these trends, potentially illuminating a class of valid compositions that can fall within the bounds of (say) a fugue or chorale.

7.3.7 Authorship

> **Tyrell**: "More human than human" is our motto.
>
> – (*Blade Runner, 1982*)

Collaborations with computational systems raise the issue of contribution toward and ownership of outcomes: when we replace parts of the creative process with an automated system, are we somehow dehumanising the resulting art object? Secondly, if such a system has been produced by another software designer, are we being invisibly driven by the tacit strategies and methods that have been encoded into the tool by their authorship? Finally, is it even *possible* to produce "creative" tools, and does it matter?

The "Inhuman" Argument If we accept that the output of a human-machine symbiosis will exhibit characteristics of both, it is frequently argued that we are introducing something (unfavourably) inhuman to a realm that is quintessentially human. At least as early as 1987, Charles Ames describes a "virulent" (Ames 1987, p. 1) resistance to the uptake of computer-aided composition on this basis.

We suggest that there are actually three underlying roots to this objection:

- that we are (knowingly or otherwise) cheating, by letting the tools do the work;
- that we are (presumably unknowingly) being directed by our tools into particular modes of operation;
- that recourse to reason alone has no place in musical composition in any case, a realm which should be driven by intuition, feeling, narrative, suffering, or other non-algorithmic concerns.

The last of these objections has been somewhat defunct in the world of avant-garde composition since Serialism or before. Barbaud (Ames 1987) responds with an elegant rejoinder:

> Music is generally called 'human' when it considers temporary or inherent tendencies of the mind, of part or all of a composer's personality. Such music is based on feeling and since it turns its back, in a sense, on pure knowledge, it might rather be called 'inhuman', for it celebrates what we have in common with all the animals rather than with what is individual to man: his reason. Algorithmic music is thus 'inhuman' only in one sense of the word, it is 'human' in as much as it is the product of rational beings. (Ames 1987, p. 173)

Similarly, in Nietzsche's comment on the typewriter "working on our thoughts", we are tempted to detect a certain pejorative tone in his voice: surrendering parts of our agency to technological devices, so the argument might go, means diluting our creative purity through the hidden bias effects of our supposedly passive tools.

Whether we prioritise intellectual or emotive forces, the acceptance of Gell's (1998) thesis negates such oppositions by arguing that *all* components of the creative process exert some agency. We suggest that the degree of such agency is not really of concern because the interactive nature of a creative partnership and the networked nature of a creative ecosystem inevitably involve some conflict and resolution, whether conscious or otherwise, and the only real concern is the status of the resultant art object itself. Contributions to the creative process can come from many directions, and while computational partners provide new opportunities, the complicated network of direct and indirect influences has long been acknowledged.

> … in truth, in literature, in science and in art, there are, and can be, few, if any, things, which in an abstract sense, are strictly new and original throughout. Every book in literature, science and art borrows, and must necessarily borrow, and use much which was well known and used before.[16]

No art production takes place in a vacuum, and is inherently made up of a nexus of eclectic forces, from the selection on instruments to the surroundings in which we develop our work. On the contrary, hermetically sealing our work within an isolation chamber would serve to starve it of the oxygen that it requires to live.

Despite this acknowledgement, the fear of technological control over our activities is deeply embedded in our culture. Themes such as these are pervasive in literature and film, from the 19th-century uncanny of Hoffmann's *The Sandman* to the dystopias of *1984*, the Borg species of *Star Trek* and the androids of *Blade Runner*. The ubiquity of networked agencies such as Web recommender systems, however, is surely beginning to allay these concerns in the public eye.

The "Invisible Hand" Argument Like with other tools, the design and development of generative music software locks in aspects of the maker's aesthetic judgement. When developing a tool which reflects a given process, certain decisions are made regarding the implementation, style, and scale of application. Further, when we incorporate general-purpose algorithmic tools the pertinence of this kind of argument rears its head in a different form: are we incorporating another *person's* creative work into our own?

As previously stated, our view is that all creative work is linked closely to its predecessors and the field in which it is located (Bown 2009). Insofar as we are taking a system and moulding it to our own goals and ends, adapting the frameworks of a third party are no more invidious than reading a magazine or visiting an exhibition in search of inspiration. Whether technological or conceptual, the raw material of ideas exists to be rebuilt, remixed and extended.

The "Creative Vitalism" Argument As we have seen previously, the objection to the idea that a computer can perform creative acts is deeply embedded in some parts of society. Noticing the level of emotive reactions to Cope's *EMI* computational composition system, Dennett comments:

[16] *Emerson v. Davies*, 8 F.Cas. 615, 619 (No. 4,436) (CCD Mass. 1845).

> It is apparently *not* crass, philistine, obscene ... to declare that all the first-order products of the tree of life—the birds and bees and the spiders and the beavers—are designed and created by such algorithmic processes, but outrageous to consider the hypothesis that creations of human genius might themselves be products of such algorithmic processes. (Dennett 2001, p. 284)

Prior to the 19th century, it was obvious to zoologists that the natural world could only exhibit its fantastic, interlocking adaptations by the hand of a designer. That a proposition is obvious, however, does not imply that it is true. The belief that the works of nature exceed the capacity of algorithmic processes is a failure of reasoning by analogy: nature appears to demonstrate the complexity of humankind's designership, and we have no better explanation, so we posit the existence of a superhuman designer. This kind of fuzzy reasoning may be useful as a rule of thumb, in the absence of a greater body of evidence, but is highly susceptible to the failings of human intuition.

However, we do not believe that this is critical to the proposition that there can be valuable creative partnerships with computational agents. Insofar as the creative acts are a result of both computer and human behaviours, the fundamentally important point is that the two together should exhibit some enhanced creativity. Rather than asking the question, "Can technology be creative?", the question can be formulated as "Can we be *more* creative with technology?" Surely, the history of human creativity with technology would suggest we can be optimistic about further extensions to this.

7.3.8 Value

During the early stages of an emergent media or technology, artworks often focus on the materiality of the medium itself. Take, for example, video art, sound sampling, and computer art. Over the embryonic years of each of these movements, many of the seminal works are those which place their medium at the forefront: Nam June Paik's distorted video signals highlighted the invisible ether of broadcast TV transmission; Christian Marclay's turntablism sonified the physical substrate of the wax record; Manfred Mohr's algorithmic drawings demonstrated the systematic, infinitely reproducible nature of computation.

These nascent experiments are undoubtedly a consequence of the exploratory and critical roles that art can play, acting as a speculum into the technology's intrinsic qualities. Subsequently, when a technology has been fully assimilated into society, it becomes a channel to convey other messages or perform other functions.

We see the same thing happening with computer-aided composition. Early practitioners such as Hiller and Isaacson (1958) and Xenakis (2001) foregrounded the formalised, computational nature of their compositions, explicitly presenting the work as being the result of automated systems. In doing so, this awareness became a part of the compositions' wider conceptual makeup: not just a piece of music, but a product of formal structures and mechanisms.

With the increasing maturity of such methods, the application of algorithms in composition has started to become more comfortably integrated with the rest of the cultural landscape. It is now incumbent on the critic to judge such hybrid human-computer works against the normal value scheme of creative works: responding to aspects of cultural fit, social impact, usefulness and beauty.

By incorporating generative processes into a feedback loop over which we then exercise selective control, one can effectively bypass the bulk of the arguments against the inhuman or uncontrollable nature of computational creativity: it is still the artist that exercises the decisive decision-making. For all the conceptual difficulty in realigning our technological understanding with our aesthetic past, the degree and complexity of reflection, development and conceptual weight are arguably all the greater.

Though a simplistic view of the human-computer creative partnership has the computer generating material and the human judging it, the reality in most systems is more complex. The degree to which the computational system or the human filters the results depends on the design of the system and/or the intent of the artists. Take, for example, the fairly hands-off approach (procedural interaction) of Iannis Xenakis with his *Gendyn* system, which was used to create the composition *Gendy 3* by generating complete works using handcrafted program settings. The final work was but one iteration selected by the composer. On the other hand, Biles' *GenJam* (Biles 1994) performs quite autonomously, improvising jazz solos created by a genetic algorithm and a database of human-performed solos. The user's control consists of playing solos that the system analyses and combines with other contextual musical information, including harmony and metre, to generate its own solos. During live performance with GenJam, there is no time for filtering of the computer's solos by the human partner.

Even though both these systems differ with regard to human filtering of the results, they both assume a considerable degree of autonomy over the generation of material. Generative systems with this degree of autonomy are often designed with a particular stylistic outcomes in mind in order to ensure that outputs fall within desired aesthetic boundaries. Other systems, such as *Nodal* (McCormack et al. 2008) and *Emily Howell* (Cope 2008), are more interactive, requiring the human to make frequent and often detailed decisions that guide the generative process. This approach can typically allow for a broader range of stylistic results because of the continual human guidance that is a check against undesirable output.

Regardless of the interaction and division of responsibility during the creation process, once music is completed by a human and generative system partnership, its value is judged like any other music by its audience appeal—whoever the audience is, and however value may be defined by them.

7.4 In Summary

> Most people who believe that I'm interested in chance don't realize that I use chance as a discipline. They think I use it as a way of giving up making choices. But my choices consist in choosing what questions to ask.
>
> – *John Cage* (Kostelanetz 1989, p. 17)

Over the course of this chapter, we have given a theoretical overview of computer-supported composition strategies, in which algorithmic systems serve to substantially augment an artist's creative activity. We hope to have convinced the reader that generative computational systems possess a distinctly new kind of agency within the creative loop, serving to increase novelty and productivity, with the distinct potential to transform creative behaviours even after our interaction with such a system has ended.

It should now be clear that there is no simple dividing line between passive and active tools; whether we explicitly encode autonomous functionality within our software or not, it still has a latent impact on the work that we do. Any description, therefore, of "normal" creative activity is a fallacy. Does our normative creativity occur after being locked in a room for a week, or after exposure to a buzzing cultural ecosystem of films, books, shops and media?

Given the capability of some interactive music systems to autonomously generate new creative trajectories on the same plane as their human counterparts, it seems only apt to characterise this relationship as a partnership. In many cases, however, it is likely that the less autonomous end of the spectrum—Absynth's randomised settings, for example—would not typically be considered as having any agency at all. Likewise, the tendency of certain production environments to funnel their users into certain modes of engagement is frequently overlooked as an active force within musical creation.

As computer music systems lend themselves to particular types of musical activity, they can be considered to embody a passive type of agency. Generative and analytical aspects of computational processes thus extend this agency to more active and significant levels. One hope is that the explicit consideration of generative tools as creative partners may heighten the awareness that even such minimal concerns *do*, in fact, impact on our creative behaviours more than we typically believe. How would the tone of this chapter have differed if it had been written in fountain pen on parchment, rather than plastic keys and a luminous LCD display?

7.4.1 Future Explorations

This research field, as with many areas of computational creativity, is still in its infancy. Partially due, no doubt, to the objections levelled in Sect. 7.3.7, these ideas have been gradual in taking hold within the musical world outside of avant-garde and academic composition. Moreover, for a composer to go beyond off-the-shelf tools and begin developing algorithmic approaches alongside their musical development has a major barrier of entry: namely, the technical know-how to do so, or the presence of an engineer-collaborator at hand.

In terms of a wider public perception, the most significant development for the field over the past decade has been a number of significant and high-profile incursions into the mainstream, often mediated through the gaming industry. The likes of *Rez*, *Elektroplankton* and *Bloom* enable casual players to make diverse and accomplished music through simple interfaces, giving a taster of what it may be like to engage with more advanced musical activities. A survey by UK charity Youth Music found that 19% of young people playing music games such as *Guitar Hero* were subsequently encouraged to start playing a musical instrument.[17]

Driven by this enlarged audience, new instruments are emerging which have some characteristics of these novice-friendly devices but with the scope to be used in more advanced, freeform contexts. The *Tenori-On*, a physical musical device designed by media artist Toshio Iwai, is a tactile sequencer with generative capabilities. Alongside attracting praise from untrained players, acting as an entry-level introduction to sequencing notions, the Tenori-On was used by innovative pop musician Björk on a recent tour. It can also be used to control user-created samples and external MIDI devices, eliminating another hidden limitation of many sound toys. With the inclusion of generative music capabilities in such devices, potential players are now presented with instruments which may reshape the gap between the beginner and the virtuoso musician, and enable many more of us to embrace creative partnerships.

7.4.2 Final Reflections

We could imagine quite different ways to group together and order these ideas. This format, however, has been brought about by our collective experiences within the field, based on the ideas, theories and questions which frequently emerge from applied use of computer-aided composition methods. It may well be that, as the field continues into maturity, further experiments will lead us to produce radically new sorts of questions and systemic categories.

Perhaps the single unifying factor which ties together each of these perspectives is that they all encourage us, in different ways, to reflect upon the entirety of creativity itself. To build generative software that operates appropriately in a creative ecosystem, we must secure some understanding of how we interact with our existing partners and tools, and how they interact with us. Likewise, designing new intimate interfaces to creativity means we must more fully understand what it means to develop a close relationship with an instrument, and the conditions necessary for virtuosity and value to arise from this.

Some understanding of the veiled process of creative partnerships with technology is necessary to drive the "productive entanglements" (Clark 2008) that we are here trying to foster. With luck, these entanglements should serve to reciprocally inform our understanding of creativity, creating another culture-scale feedback loop.

References

Ames, C. (1987). Automated composition in retrospect: 1956–1986. *Leonardo*, *20*(2), 169–185.

[17]"Why console-games are bigger than rock 'n' roll": http://www.youthmusic.org.uk/research-archive.html.

André, P., Schraefel, M. C., Teevan, J., & Dumais, S. T. (2009). Discovery is never by chance: designing for (un)serendipity. In *C&C'09: proceedings of the seventh ACM conference on creativity and cognition* (pp. 305–314). New York: ACM.
Beilhartz, K., & Ferguson, S. (2007). Gestural hyper instrument collaboration with generative computation for real time creativity. In *Creativity and cognition* (pp. 213–222). Washington: ACM.
Berry, R., & Dahlstedt, P. (2003). Artificial life: why should musicians bother? *Contemporary Music Review*, *22*(3), 57–67.
Biles, J. (1994). GenJam: A genetic algorithm for generating jazz solos. In *Proceedings of the international computer music conference* (pp. 131–137).
Boden, M. A. (2004). *The creative mind: myths and mechanisms*. New York: Routledge.
Bown, O. (2009). Against individual creativity. In *Dagstuhl seminar proceedings 09291. Computational creativity: an interdisciplinary approach*, Dagstuhl, Germany.
Bown, O., & Lexer, S. (2006). Continuous-time recurrent neural networks for generative and interactive musical performance. In *Lecture notes in computer science*. *Proceedings of EvoWorkshops 2006* (pp. 652–663). Berlin: Springer.
Brown, A. R. (2000). Modes of compositional engagement. *Mikropolyphonie*, *6*. http://pandora.nla.gov.au/tep/10054.
Brown, A. R. (2001). How the computer assists composers: a survey of contemporary practice. In G. Munro (Ed.), *Waveform 2001: the Australasian computer music conference* (pp. 9–16). The Australasian computer music association.
Brown, A. R. (2002). Opportunities for evolutionary music composition. In *Proceedings of the Australasian computer music conference* (pp. 27–34).
Cage, J. (1968). *Silence: lectures and writings*. London: Calder and Boyars.
Chadabe, J. (1984). Interactive composing: an overview. *Computer Music Journal*, *8*(1), 22–27.
Clark, A. (2003). *Natural-born cyborgs*. Oxford: Oxford University Press.
Clark, A. (2008). *Supersizing the mind*. New York: Oxford University Press.
Clark, A., & Chalmers, D. (1998). The extended mind. *Analysis*, *58*(1), 7–19.
Cohen, H. (1995). The further exploits of AARON, painter. *Stanford Humanities Review*, *4*(2), 141–158.
Cole, T., & Cole, P. (2008) Noatikl. http://www.intermorphic.com/tools/noatikl/index.html.
Collins, N. (2009). *Introduction to computer music*. Chichester: Wiley.
Cope, D. (1996). *Experiments in musical intelligence*. Madison: A-R Editions.
Cope, D. (2008). *Hidden structure: music analysis using computers*. Madison: A-R Editions.
Cornock, S., & Edmonds, E. (1973). The creative process where the artist is amplified or superseded by the computer. *Leonardo*, *6*(1), 11–16.
Csikszentmihalyi, M. (1992). *Flow: the psychology of happiness*. London: Rider Books.
De Bono, E. (1992). *Serious creativity: using the power of lateral thinking to create new ideas*. London: Harper Collins.
Dean, R. (2003). *Hyperimprovisation: computer-interactive sound improvisation*. Middleton: A-R Editions.
Dennett, D. (2001). Collision detection, muselot, and scribbles: some reflections on creativity. In D. Cope (Ed.), *Virtual music: computer synthesis of musical style* (pp. 282–291). Cambridge: MIT Press.
d'Inverno, M., & Luck, M. (2004). *Understanding agent systems*. *Springer series on agent technology*. Berlin: Springer.
Eno, B. (1996). Generative music. http://www.inmotionmagazine.com/eno1.html.
Eno, B., & Schmidt, P. (1975). *Oblique strategies: over 100 worthwhile dilemmas by Brian Eno and Peter Schmidt*. London: Apollo.
Essl, K. (1992). Lexikon sonate. http://www.essl.at/sounds.html#lexson-porgy.
Galanter, P. (2003). What is generative art? complexity theory as a context for art theory. In *GA2003—6th generative art conference*.
Gell, A. (1998). *Art and agency: an anthropological theory*. Oxford: Clarendon Press.
Goel, V. (1995). *Sketches of thought*. Cambridge: MIT Press.
Gudmundsdottir, B. (1996). Björk meets Karlheinz Stockhausen. *Dazed and Confused*, *23*.

Hedemann, C., Sorensen, A., & Brown, A. R. (2008). Metascore: user interface design for generative film scoring. In *Proceedings of the Australasian computer music conference* (pp. 25–30). Australasian computer music association.

Heidegger, M. (1977). *The question concerning technology and other essays*. New York: Harper & Row.

Hiller, L. (1968). *Music composed with computer[s]: an historical survey*. Experimental Music Studio.

Hiller, L. A., & Isaacson, L. M. (1958). Musical composition with a high-speed digital computer. *Journal of the Audio Engineering Society*, *6*(3), 154–160.

Hodges, A. (1985). *Alan Turing: the enigma of intelligence*. London: Unwin Paperbacks.

Huron, D. (2006). *Sweet anticipation: music and the psychology of expectation*. Cambridge: MIT Press.

Illich, I. (1973). *Tools for conviality*. London: Valder & Boyars.

Jewell, M. O., Rhodes, C., & d'Inverno, M. (2010). Querying improvised music: do you sound like yourself? In *ISMIR 2010*, Utrecht, NL (pp. 483–488).

Jones, D. (2008). AtomSwarm: a framework for swarm improvisation. In *Lecture notes in computer science*. *Proceedings of EvoWorkshops 2008* (pp. 423–432). Berlin: Springer.

Jones, D., Matthias, J., Hodgson, T., Outram, N., Grant, J., & Ryan, N. (2009). The fragmented orchestra. In *Proceedings of new interfaces for musical expression (NIME 2009) conference*, Pittsburgh, PA, USA.

Jordà, S., Geiger, G., Alonso, M., & Kaltenbrunner, M. (2007). The reactable: exploring the synergy between live music performance and tabletop tangible interfaces. In *Proceedings of the first international conference on tangible and embedded interaction (TEI)* (pp. 139–146). New York: ACM Press.

Kittler, F. A. (1999). *Gramophone, film, typewriter*. Stanford: Stanford University Press.

Klee, P. (1972). *Pedagogical sketchbook*. London: Faber and Faber.

Kostelanetz, R. (1989). *Conversing with cage*. London: Omnibus.

Laske, O. (1981). Composition theory in Koenig's project one and project two. *Computer Music Journal*, *5*(3), 54–65.

Latour, B. (1994). On technical mediation. *Common Knowledge*, *3*(2), 29–64.

Lewis, G. (2007). On creative machines. In N. Collins & J. d'Escriván (Eds.), *The Cambridge companion to electronic music*. Cambridge: Cambridge University Press.

Lewis, G. E. (2000). Too many notes: computers, complexity and culture in voyager. *Leonardo Music Journal*, *10*, 33–39.

Lubart, T. (2005). How can computers be partners in the creative process. *International Journal of Human-Computer Studies*, *63*(4–5), 365–369.

Magnusson, T. (2009). Of epistemic tools: musical instruments as cognitive extensions. *Organised Sound*, *14*(2), 168–176.

Matthews, H., & Brotchie, A. (2005). *The Oulipo compendium*. London: Atlas Press.

McCormack, J. (1996). Grammar based music composition. In R. Stocker, H. Jelinek, B. Durnota & T. Bossomaier (Eds.), *Complex systems 96: from local interactions to global phenomena* (pp. 321–336). Amsterdam: ISO Press.

McCormack, J. (2003). Evolving sonic ecosystems. *Kybernetes*, *32*(1/2), 184–202.

McCormack, J., McIlwain, P., Lane, A., & Dorin, A. (2008). Generative composition with nodal. In E. Miranda (Ed.), *Workshop on music and artificial life (part of ECAL 2007)*, Lisbon, Portugal.

McCullough, M. (1996). *Abstracting craft: the practiced digital hand*. Cambridge: MIT Press.

McGraw, G., & Hofstadter, D. (1993). Perception and creation of diverse alphabetic styles. *AISB Quarterly*, *85*, 42–49.

McLuhan, M. (1964). *Understanding media: the extensions of man*. London: Sphere Books.

Miranda, E. R., & Matthias, J. (2005). Granular sampling using a Pulse-Coupled network of spiking neurons. In *Lecture notes in computer science*. *Proceedings of EvoWorkshops 2005* (pp. 539–544). Berlin: Springer.

Narmour, E. (1990). *The analysis and cognition of basic melodic structures*. Chicago: University of Chicago Press.

Nierhaus, G. (2009). *Algorithmic composition: paradigms of automatic music generation*. Vienna: Springer.

Norman, D. (1991). Cognitive artifacts. In J. M. Carroll (Ed.), *Designing interaction: psychology at the human-computer interface* (pp. 17–38). Cambridge: Cambridge University Press.

Ortolani, B. (1990). *The Japanese theatre: from shamanistic ritual to contemporary pluralism*. Leiden: E. J. Brill.

Pachet, F. (2003). The continuator: musical interaction with style. *Journal of New Music Research*, *32*(3), 333–341.

Pólya, G. (1971). *How to solve it: a new aspect of mathematical method* (2nd ed.). Princeton: Princeton University Press.

Prévost, E. (2004). *Minute particulars: meanings in music—making in the wake of hierarchial realignments and other essays*. London: Copula.

Pritchett, J. (1993). *The music of John Cage*. Cambridge: Cambridge University Press.

Rowe, R. (1993). *Interactive music systems: machine listening and composing*. Cambridge: MIT Press.

Sawyer, R. K. (2006). Group creativity: musical performance and collaboration. *Psychology of Music*, *34*(2), 148–165.

Smith, H., & Dean, R. (1997). *Improvisation, hypermedia and the arts since 1945*. Amsterdam: Harwood.

Spiegel, L. (1981). Manipulations of musical patterns. In *Proceedings of the symposium on small computers and the arts* (pp. 19–22).

Supper, M. (2001). A few remarks on algorithmic composition. *Computer Music Journal*, *25*(1), 48–53.

Temperley, D. (2007). *Music and probability*. Cambridge: MIT Press.

van Andel, P. (1994). Anatomy of the unsought finding. Serendipity: origin, history, domains, traditions, appearances, patterns and programmability. *British Journal for the Philosophy of Science*, *45*(2), 631–648.

Wiggins, G., Papadopoulos, G., Phon-Amnuaisuk, S., & Tuson, A. (1999). Evolutionary methods for musical composition. In D. M. Dubois (Ed.), *Partial proceedings of the first international conference CASYS'97 on computing anticipatory systems*, Liége, Belgium, August 11–15, 1997. International Journal of Computing Anticipatory Systems.

Wiggins, G., Pearce, M., & Mullensiefen, D. (2009). Computational modelling of music cognition and musical creativity. In R. Dean (Ed.), *Oxford handbook of computer music* (pp. 383–420). Oxford: Oxford University Press.

Winkler, T. (1998). *Composing interactive music*. Cambridge: MIT Press.

Woolhouse, M. (2009). Modelling tonal attraction between adjacent musical elements. *Journal of New Music Research*, *38*(4).

Xenakis, I. (2001). *Formalized music: thought and mathematics in composition*. Hillsdale: Pendragon Press.

Young, L. M. (Ed.) (1963). *An anthology of chance operations*. New York: La Monte Young and Jackson Mac Low.

Young, M. (2008). NN music: improvising with a 'living' computer. In R. Kronland-Martinet (Ed.), *Computer music modelling and retrieval: sense of sounds* (pp. 337–350). Vienna: Springer.

Chapter 8
Between Material and Ideas: A Process-Based Spatial Model of Artistic Creativity

Palle Dahlstedt

Abstract In this chapter, I propose a model of an artistic creative process, based on study of my own creative processes over twenty years of activities as composer and improviser. The model describes the creative process as a structured exploration of the space of the possible, emphasising the interplay between a dynamic concept and the changing material form of the work. Combining ideas, tools, material and memory, creativity is described as a coherent, dynamic, and iterative process that navigates the space of the chosen medium, guided by the tools at hand, and by the continuously revised ideas, significantly extending previous spatial models of creativity. This involves repeated misinterpretation and coincidences, which are crucial in human creative processes, adding meaning and depth to the artwork. A few examples from real life are given as illustrations of the model, together with a discussion of phenomena such as appreciation, skill and collaborative creativity. Finally, I discuss how the proposed model could form a foundation for computer implementations of artistic creative process, to increase our understanding of human creativity, and to possibly enable believable artistic behaviour in machines.

8.1 Introduction

Humans have always wanted to build intelligent machines, with various degrees of success. One particularly elusive property of intelligent behaviour is creativity. How do we form new ideas? How do we create something nobody has previously seen? Creative insights may seem like momentary events, but under the surface of the consciousness, they are gradual processes, combining and elaborating previous knowledge into new thoughts, until the conditions are just right for them to surface.

In art, creativity is essential. The formation of ideas is important, but in my experience, the depth and meaning of an artwork emerge from the process of implementation of the original ideas, during which these ideas may very well change, drift and be elaborated upon, sometimes beyond recognition.

P. Dahlstedt (✉)
Dept. of Applied Information Technology, University of Gothenburg, 41296 Göteborg, Sweden
e-mail: palle@chalmers.se

J. McCormack, M. d'Inverno (eds.), *Computers and Creativity*,
DOI 10.1007/978-3-642-31727-9_8,

In this chapter I propose a spatial model of the artistic creative processes, which combines the conceptual aspects of a work with the implications of the artistic tools we are using and the material in which the work is created. I take a process-based perspective, founded primarily on introspective study of my own artistic creative processes, but also on experience from artistic collaborations and extensive artistic teaching and supervision.

The model combines key concepts such as ideas, tools, material and cultural background, and views creativity as a dynamic, iterative process that navigates the space of the theoretically possible (in the chosen medium) following paths defined by what is practically possible (by the tools at hand). The process is guided by a continuously revised conceptual representation—the changing ideas behind the work. The model also involves phenomena such as self-interpretation, coincidences and reformulation of the concepts behind a work, which are crucial in human creative processes. Both real-time creativity (e.g. improvisation) and non-linear processes (composition) are included in the discussion, as well as collaborative creative processes, such as group improvisation and larger collaborations.

I believe the presented model can help us understand the mechanisms of artistic creative processes better, and it provides a framework for the discussion and analysis of artistic creativity. And it can form the basis for experiments in computational creativity.

8.1.1 Background

Spatial models of creativity have been presented before. Perhaps the most well known is Margaret Boden's concept of exploration and transformation of spaces (Boden 2004), and the ideas presented here may be considered an extension of her ideas, primarily through the introduction of a material space in addition to her conceptual spaces, and the implications of the interplay between these two forms of representation.

The model is based on observation of my own creative work during more than two decades of artistic activities as a composer, improviser, programmer and sound artist, from collaborations with other artists from many genres, and from extensive artistic teaching and supervision in music and technology-related art. I have consciously observed my own creativity and creative processes since my early teens, up to the present. In the last ten years, I've pursued research into computer-aided creativity, primarily based on evolutionary algorithms, in parallel with and overlapping my work as a composer. From these two related fields, a number of originally unconnected observations have fallen into place, forming a coherent view of creative processes, as I see them unfold in my own daily artistic practice. Hopefully, it is also more generally applicable. The model was presented in a more preliminary form at the Computational Creativity Dagstuhl Seminar (Dahlstedt 2009a).

Being both researcher and professional artist/composer, I have a peculiar advantage, because I have access to a level of information about my creative process that

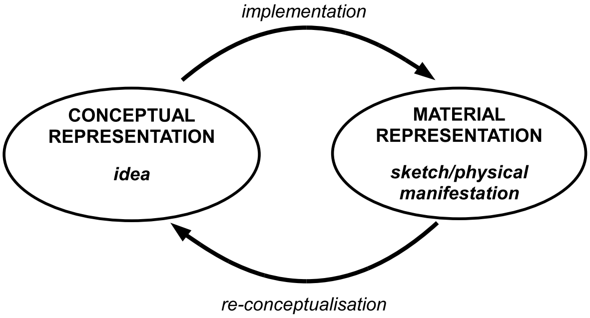


Fig. 8.1 The artistic work is developed in an iterated process between a conceptual representation, as ideas in the head of the artist, and the current temporary material form, as a sketch or unfinished work. Each translation step between these two forms helps adding detail and depth to the work

is unavailable to an outside observer. Aware of existing theories of creativity, and with knowledge about key concepts and mechanisms, I can systematically observe my own processes and draw conclusions which would be impossible if constrained to artistic results and rhapsodical accounts by others. So, as a researching artist, I am able to form theories and models based on observation. Then, these theories and models can be confirmed by others, if they fit their observations of their own or others' creative behaviour. And potentially, they can be confirmed by simulation in software, and by evaluation of the artistic outcome of those simulations.

The chapter is not primarily about the novelty aspect of creativity, or the social and cultural aspects (these issues are addressed in other chapters in this volume). It concentrates on what goes on in the mind of an artist during the birth and development of an artwork from concept to material form. It is primarily based on experience from music and sound art, but I believe the ideas are applicable to many other domains. I aim to provide a framework for how artists actually go about realising an artistic idea—maybe not all artists, but I believe many can feel at home in my description. For simplicity, I will often use examples from simple drawing in my explanations, to avoid musical terms that may be unfamiliar to the general reader.

The model also provides a terminology and apparatus for analysis of actual creative processes, and a new framework for the emulation of human artistic creativity. A lot of computational creativity research focuses on the birth of ideas, but as a practising artist and composer, I see in my daily practice that more important than the birth of ideas is this dialogue with the material; with given material (as for a sculptor and his archetypical marble block) or crafted material—temporary results and sketches. In my experience, ideas emerge from this dialogue, from misunderstandings, ambiguities and mistakes. A very small part is the original concept. It is but a seed, and in some creative processes it may not even exist, e.g. in certain kinds of improvisation. Much more important is the process, the bouncing between concept and material. Tools provide the paths to go from concept to material—I call this *implementation*, while *re-conceptualisation* takes us the back from material to idea, as illustrated in Fig. 8.1. This is the hard part, where we interpret our own temporary results, and extend, constrain or revise our concept. This process is repeated, until idea and material have converged into a finished artwork.

8.1.2 Outline

In the following section, I discuss the idea of tools, the implications of their use, and the notion of spaces and topologies related to these tools. Section 8.3 presents the model, explaining the main ideas on a general level, such as material and conceptual representation, and the interplay between them, including brief discussions on topics such as craft, skill, novelty and appreciation, and collaborative creativity, in the light of the proposed model. It is also discussed in the context of existing theories. How the model could possibly be implemented in computers is discussed in Sect. 8.4, followed by some concluding remarks.

8.2 Tools

The word *tool*, in a wide sense, is used a lot throughout this chapter, denoting everything from a traditional drawing tool (e.g. a paintbrush) or a musical instrument to an abstract organising principle (spectral harmony), a given musical form (the fugue), computer programs (Photoshop filters), generative procedures (grammar systems, evolutionary algorithms, Markov chains) or representational systems (Western music notation).

Artistic expression is clearly affected by the choice of tools. New genres and subgenres constantly emerge in music, triggered by the availability of new kinds of tools for music-making, such as loop samplers, live sequencers, time- and pitch-altering algorithms, and many more, allowing new ways to work with sound and structure. A tool embodies a complex behaviour (Gregory 1981) and enables lines of thoughts that would not be otherwise possible.

With more advanced tools, the contribution from the toolmaker cannot be ignored. It may be good or bad, but the artist has to be aware of it. Sometimes you do not want to spend time on developing your own tools, but prefer to be confronted with an existing tool, and take advantage of the extensive design effort put in by the tool maker. He helps transport me a fair way towards sophistication, through using his tool. A well-known risk is that the tool steers users towards similar results. But given that the tool is complex enough, i.e. it provides possibilities of considerable user-controlled variation, and that I spend a decent amount of effort on my work, the tool might not limit my artistic contribution.

Each tool defines a virtual space of possible results. It also defines a *topology* within this space. A topology is a set of neighbourhood relations within the space, determining which points are near each other, and consequently how we can traverse the space. A neighbour point, in this case, is another point that you can reach with a single application of the tool. These topologies defined by tools are very important, since they correspond, in different ways, to how we think about the work. First, we naturally think about ideas in terms of how to realise them, using tools. Second, I believe the realm of our imagination is to a large extent constructed from our knowledge about existing tools, from practice and studies, and what we have learnt from the results of their use in art, our own and others. The design of the tool also steers our thoughts and our imagination towards what is possible or easy, and towards what is achievable, practical, or challenging. This amounts to Norman's (1988) use of Gibson's (1977) term *affordance*.

When learning a new tool, I gradually form a cognitive model of how it works. Spaces of potential results open up in my mind, expanding as the cognitive model gets more elaborate and accurate. If it is reasonably adequate, it gives me a predictive capacity in relation to that specific tool. That is, I have some expectation of what will happen when I use the tool in a certain way. But the predictions are not always correct, because of my limited cognition, or because of mistakes or tool failures, which introduce unexpected results and irregularities to the material.

The topology inferred by the tool also brings a kind of metric—a system of distances. Different points in the result space are at different distances from each other, i.e. certain points are easier or more difficult to reach from where you are. This is dependent on a formal metric—the number of times you have to apply the tool to get there, but also on a perceived metric, affected by the tool precision, the difficulty of use, and the affordance of the tool—certain paths are more easily accessible than others, and narrow paths may be more rewarding. A skilled listener or viewer can perceive this metric, and it is part of the experience of the artwork; the perceived effort, respect for craftsmanship and skill, in a kind of empathetic appreciation.

As an example of how tools steer our thoughts, we can compare two common kinds of musical tools: *predesigned* and *modular synthesisers*.[1] The first category, the predesigned synthesiser, provides a certain number of functions in a fixed configuration, typically routing sound signals from tone generators through a filter and variable amplifier, modulated by a limited set of gestural modulators to shape the sound over time. All these functions are controlled by a fixed number of parameters. Behind such an instrument are careful considerations by the instrument designer regarding playability, choice of features, interface design, relevance of parameters, etc. A modular synthesiser, on the other hand, provides a large number of abstracted functions in modules that can be connected in any order and configuration, with free routing of audio and control signals. Typical modules include: oscillators, filters, modulation sources, amplifiers, mixers, etc. Digital modular systems, additionally, provide free configuration of processing resources, and their openness and flexibility essentially equals that of computer programming. The predesigned synthesiser is a subset of the modular synthesiser, and the latter can easily be configured to mimic most predesigned synthesisers. Despite this shared functionality, we seldom use them in the same way. Users of modular synths are predominantly occupied by changing the configuration and routing, adding and removing modules from the signal chain. It is only rarely used to build an optimal configuration which is then subject to extensive exploration of its parameter space. The main difference between the two is in the variables they provide. Their spaces are different in size and scope,

[1] These comments on how synthesisers are used, are based on background studies made in conjunction with the design and development of an interactive evolutionary sound design tool for the Nord Modular G2 synthesiser (Dahlstedt 2007).

and as users we tend to explore the space that the tool provides, and we tend to travel the easy and accessible paths first. If you *can* add new modules and connections, you will. To impose further constraints on this freedom requires discipline and knowledge, and an understanding of why you would want to lock certain variables. And sometimes the toolmaker provides that understanding for you.

The idea of a space of possibilities for a specific tool or representation is old, but it is not enough in itself to give a complete picture of the creative process. Also, very seldom do we use just *one* tool to create a work of art. We use a whole toolbox of them, and we switch between them, depending on what is needed at the moment. To understand the creative implications brought about by the tools, we need to be able to discuss and compare the different spaces and topologies provided by them. And equally important, we need to consider the constraints and possibilities of the material: the medium in which we create our work, such as image or sound. Tools are the ways we navigate the infinite space of inherent possibilities of the material, but only along the pathways offered by the tools. Hence, we must introduce the notion of a *material space*, a larger space containing all possible images or sounds, and which can be traversed along the topologies provided by the tools at hand.

And if we are going to emulate human creative behaviour, it is not enough to implement the tools. We also have to emulate the structured application of these tools by a human artist. Such a model thus operates on three levels: a material representation storing temporary results in simplest possible form, implementations of tools that provide a means of navigation in the space of possible results, and a model of how these tools are applied in a structured, iterated process in relation to ideas and cultural context. In the following section, I will describe a model based on these ideas.

8.3 The Model

I will first give an overview of the model, including the main concepts, each of which will be further detailed in separate sections. This is followed by a couple of real-world examples from composition and improvisation, and a discussion of how the model relates to existing theories. This is followed by a brief discussion of related concepts, such as skill, collaborative processes and tools, examined in the light of the proposed model.

The basic idea is that a creative process is an exploration of a largely unknown space of possibilities. The exploration follows paths that are not arbitrary. As an artist, I do not believe in free creation, since we are influenced by many things: the tools at hand, our knowledge of the tools, our ideas and concepts, what we have seen before, liked and unliked, and by our view of the world. Each of these form patterns in the space of possible results, in the form of possible or preferred outcomes—subspaces, and neighbourhood relations—topologies, which form possible paths for our search. These topological subspaces, one for each tool, form networks (or graphs, sometimes trees) in the larger material space, which intersects each other. For simplicity, in the following I will use the word *network* to denote such a topological subspace, for lack of a more suitable word.

While exploring, the work that is being created exists in two forms simultaneously: in a *material representation* and a *conceptual representation*. The material representation is the current form of the work in the chosen medium, e.g. as a sound sketch or an unfinished image. It corresponds to a single point in the material space, the space of all theoretically possible images. The conceptual representation is the current form of the work in terms of ideas and generative principles. It corresponds to a point in a conceptual space; the space of all possible conceptual representations. A particular conceptual representation defines a subspace in the material space—the set of all images or sounds, i.e. points, that could be interpreted as corresponding to this concept. In parallel to the topological tool networks, there is also a topology of subspaces in the material space, defined by the variability of the conceptual representation. If the conceptual representation is changed or revised, this subspace is transformed, and will cover new regions and allow new pathways in the material space. This system of related subspaces corresponds to topological networks in the conceptual space, but I will call them conceptual networks, for simplicity. An illustration of these related spaces is given in Fig. 8.2.

The focus of the creative process continuously changes between these two forms, and requires mechanisms to translate from one into the other, in both ways. Let us call them *implementation*, when we go from concept to material, and *re-conceptualisation*, when the concept is revised or recreated based on the current material form. The discrepancies between the two representations, and the imprecision of the translation in both directions fuels the creative exploration, embeds qualities of human expression in the work, and imprints a trace of the creative process onto the work itself.

The implementation of a concept into a material manifestation happens through the application of tools, and this process is imprecise due to the natural vagueness of ideas, the characteristic incompetence of the artist, the imperfection of the tools themselves, and his possible lacking mastery thereof—visible as a limitation in his predictive capacity.

In the other direction, the continuous re-conceptualisation of material form into a new conceptual representation, which may or may not resemble the previous one, is by its very nature imprecise and prone to misunderstandings. It is precisely this vagueness that is the heart of the field of interpretative arts, such as musical performance and theatre. But I think it is also crucial within the creative process of a single author, since he continuously interprets and re-interprets his own work as it is given form.

8.3.1 Material Space and Representation

The material representation is simply the current, temporary form of the work, e.g. as a drawing, a musical sketch, or a sound file. The material space is a theoretical

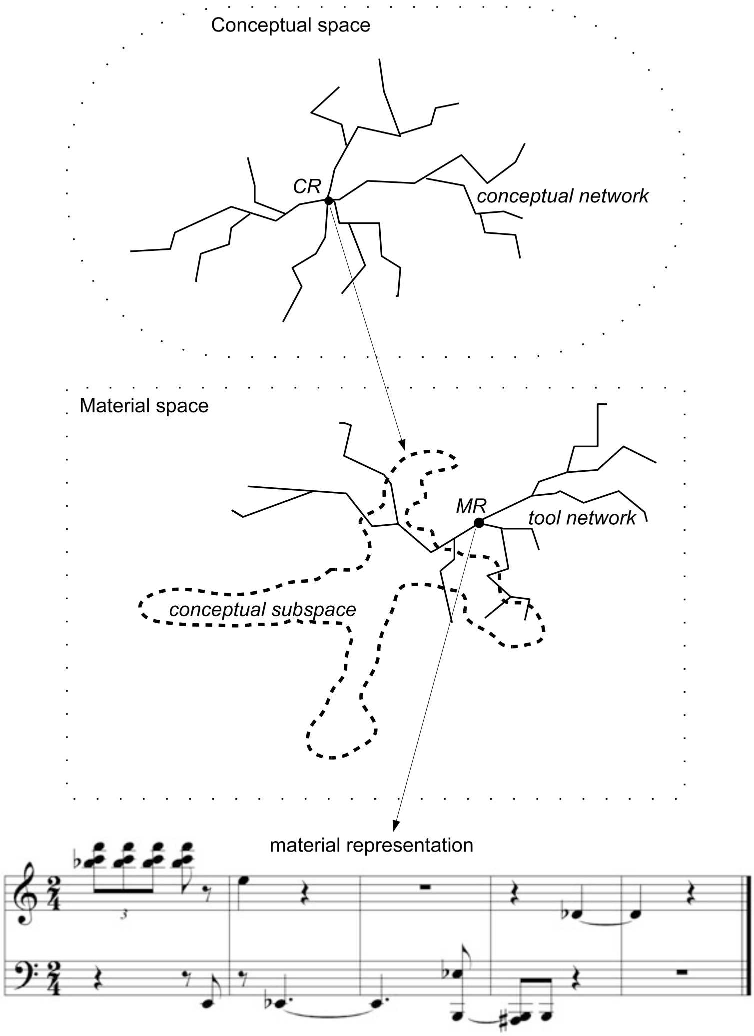


Fig. 8.2 At a certain moment, the artwork exists as a conceptual representation, which corresponds to a point (marked *CR*) in a conceptual space of all possible conceptual representations. Possible variations to the idea constitute a topological network in this space. The current conceptual representation defines a subspace in the material space of all possible material manifestations of this particular concept. The current material representation of the work is a point (marked *MR*) in the material space of all possible material results. This point can be either inside or outside of the current conceptual subspace. If it is outside, the artist can either alter the concept to include the current material representation, or change the material form, by the application of tools. Possible alterations by a specific tool are indicated, forming a topological network in the material space

construction that contains all its possible instances. If we work with images, the material space consists of all possible images, for example, a bitmap of a certain size and resolution, which theoretically can represent any conceivable image. If we work

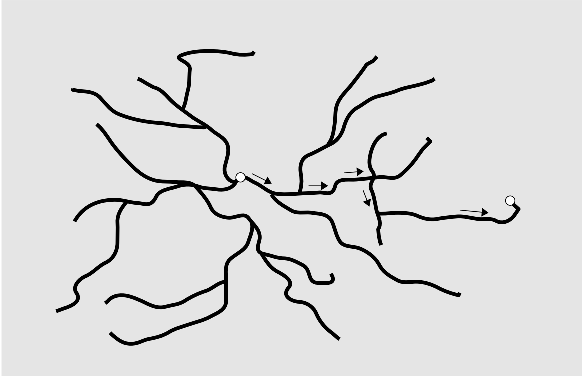


Fig. 8.3 The topological subspace defined by a specific tool forms a network in the material space. Each application of the tool (e.g. a brush) moves a small step along the accessible pathways. Repeated use of the tool can take us far

with sound or music, the material space consists of all theoretically possible sounds of a certain maximum length and bandwidth. These spaces are truly huge, with as many dimensions as there are sound samples or pixel colour values. Musicians or artists seldom conceive of sounds in these representations, since they are very distant from the conceptual level of a work, but as theoretical constructs they are convenient and important, as we shall see.

In other contexts, the material representation could be a three-dimensional form, a musical score, or a text, the latter two are slightly closer to a structural-conceptual description of a work, but the mechanisms are similar.

At any specific time, the temporary form of a work is represented by one point in the material space; one image out of the almost infinitely many possible images. Through the application of a specific tool, we can reach a number of neighbour points. In this way, a network of paths is formed, defining a topological subspace: a network (see Fig. 8.3). In some contexts that don't allow repeated configurations to occur (e.g. wood-carving), these networks are structured like trees, while in other cases periodic trajectories can occur.

Let us look at a simple example. A specific tool, e.g. a paintbrush or a filter in Photoshop, with some parameters, operates on a particular bitmap and returns another. That is, it goes from one point in the material space to another. From a specific image you can travel to a certain number of other images that are within reach by a single application of this particular tool. With this tool, I can only navigate the material space along these paths. I can go from an image of a red square to an image of a red square with a blue line by using the brush to make the line. But I need two steps to go to an image of a red square with two blue lines. Hence, the vertices of the topological network of this particular tool are the points in material space (representing in this case bitmap images), while the edges are connections between points that can be reached by one application of the particular tool.

The material space may also have an inherent topology, based on the most obvious neighbour relation—the change of a value of a single pixel colour or a single-sample. However, this topology is too far removed from the conceptual level of the human mind to be of particular use, and we cannot even imagine how it would be to navigate the material space in this way, since such a small part of it contains

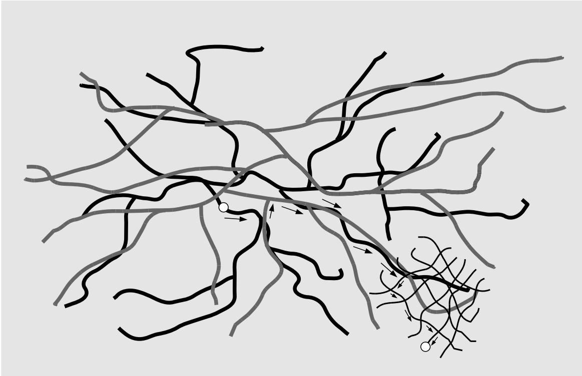


Fig. 8.4 The tool networks of a set of tools in material space. Each tool defines a different network, covering different areas, with different resolution. At each intersection, we can switch to another tool and continue navigation along different paths. Here, two coarse tools are used first (*black* and *grey thick lines*), followed by more fine-tuned editing by a tool of higher resolution

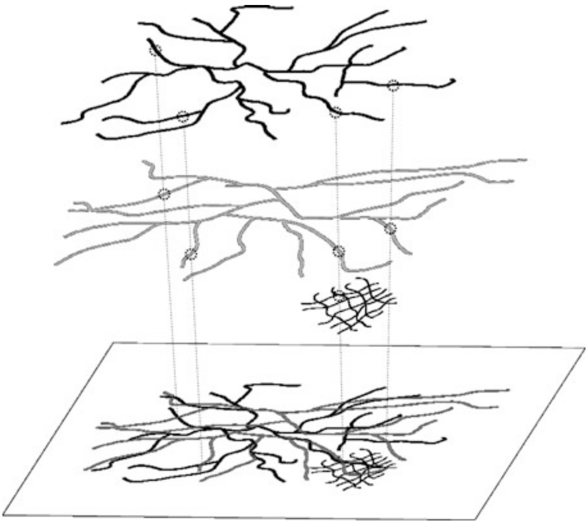


Fig. 8.5 The different tool networks are not separate structures, but different organisational principles in the same material space—represented by the *bottom rectangle*. A single point in this space, representing, e.g. a particular image, can be part of a number of different networks, that provide or control movement out of that point—how it can be varied

anything we would consider meaningful. Most of it is noise, or would appear completely disordered to our perception. We need tools to navigate this space; to get from one interesting point to the next, which do not get their proximity according to the inherent topology of the material, but by the tool-based networks.

Each tool defines a different topology in the same material space. Together they form intersecting networks, defining the possible paths of artistic exploration. Combinations of tools allow us to travel more freely in the material space, since the combined networks cover a larger subspace of the theoretically possible, and provide a larger selection of travel paths. At any intersection, I can switch to another tool, and hence to another network of accessible pathways, as illustrated in Figs. 8.4 and 8.5.

This can be compared with physical travel—some places can only be reached by car, because they are distant. When the road ends, we put on skis or snowshoes, or simply walk. Some locations can only be reached by airplane or helicopter, or require extra oxygen. Some points are easier to reach aided by GPS navigation, others require ropes and harness. Each means of transport provides certain navigable or facilitated paths, and where the path networks intersect, i.e. where both means are possible or needed, we can change our way of travelling. All of them bring different potential and constraints, just like different tools.

So, one idea behind the introduction of a material space is that we can start thinking about application of different tools in succession, since they all operate in the same space—the material space. They all define different topological networks in the material space, which intersect, and we can switch between tools at any time. Another reason is that the material representation introduces true open-endedness, since anything can happen in the material form. I can spill coffee on my score, or there can be a tool failure. A teacher or collaborator can alter my sketches. All this of course adds further complications to the process, but these cases still fit the model.

8.3.2 The Conceptual Representation

The conceptual representation of the work is how it is represented in the mind of the artist, in terms of abstract or concrete ideas and generative principles. This representation is vague with respect to the material representation. If my idea is a picture of ten monkeys forming a pyramid, this conceptual representation corresponds to the set of all images that can be interpreted as a pyramid of ten monkeys. Since nothing is said about the colour and species of the monkeys, where they are located, or from which angle we see them, there are a lot of images that fit this description.

In the course of the creative process, the conceptual representation is changed, e.g. by being made more specific, which shrinks the subspace, or altered, which transforms the subspace. The internal structure of the conceptual representation determines which transformations are possible, along the lines of the variable parameters of the representation. If my idea, again, is ten monkeys forming a pyramid, the variables in this representation are the kind of animals, the number of individuals, their formation, etc. If I decide that it should be specifically ten cotton-top tamarines, or ten monkeys in a pyramid under a rainbow, the subspace shrinks. If I elaborate my idea to be ten mammals forming an upside-down pyramid, or a number of monkeys in any formation, the subspace is restructured or expanded. This relates directly to the invention of new knobs to turn (Hofstadter 1985) or Boden's transformation of spaces, and is one of the challenges of computational creativity.

The conceptual representation can be vague in at least three different ways. First, there may be many points in the material space that comply with the ideas expressed—it defines a subspace of many possible results. Second, the conceptual representation may not yet include the necessary small design decisions that we often postpone to the implementation stage. Third, because of our limited predictive capacity, generative works can be exactly defined by concepts, but we don't know what the outcome will be. Our expectations—what we envision—form a subspace of the material space, but when we carry out the generative procedure, a single point will be the result. That point may or may not be a part of what we expected, possibly requiring a revision of the conceptual representation.

8.3.3 Interplay Between Representations

The philosopher Daniel Dennett has said (Denton 2004):

> The purpose of the brain is to predict the future, to make plans and hopes, and in following these predictions, we partially make the future.

The brain is good at prediction, because that is what it is evolved to do. The musician and writer Stephen Nachmanovitch (1990) said that life *is* improvisation. But creative processes also mimic what life is about—predicting, pursuing, acting, adjusting, etc. in a continuous circular process. So, in describing how we form our world, Dennett also gave us a good description of how we create art.

As a composer, I use generative processes to project my ideas beyond my predictive horizon (Dahlstedt 2001). I may understand the conceptual network in the immediate neighbourhood, and apply the algorithm or process to get further away, hoping that the interestingness will carry over to distant parts of the space. Or I may understand the broad paths in the conceptual network of the process, and apply it, leaving the details to the process. I may use generative processes that are too complex for my predictive capacity, in a trial-and-error fashion: adjusting parameters as I go, based on temporary results, and possibly, at the same time, adjust the actual algorithm itself. This amounts to the reiterated interplay between material and conceptual representation, through development and parsing.

This interplay is crucial to the proposed model. An idea expressed in a conceptual representation is realised by searching for a suitable material representation, either by gradually shrinking the set of points covered by the conceptual representation in an iterated process between idea and tools, or by searching for a unknown pleasing result by trying a sketch, evaluating it and modifying it until something interesting is found. Once again, this is an iterated process between ideas, tools and material, and can be illustrated in terms of these networks (tool networks, conceptual subspaces, etc.) that coexist as different organisational principles in the material space.

There has to be a path from the material representation back to the conceptual representation, to carry interesting coincidental results back into the conceptual representation, and to provide for feedback from temporary results to affect the conceptual representation. How do we recognise pregnant ideas and interesting coincidence? What we need is a kind of reverse development process: the parsing of a material representation into a conceptual description. This is a central part of the creative process; our brains do it all the time, but computationally it is a non-trivial problem. The translation from concept to material is essentially irreversible, and to form a cognitive model of a material is imprecise and gives a model different to the original. This difference gives birth to new material and creative variations. It is analogous to the concept of interpretation, as in classical music and theatre. We cannot recreate the conceptual model of the original composer or playwright, and each performance is different. During the creative process, the artist has to interpret his own work repeatedly, to be able to evaluate the temporary form of the work, and to take advantage of unpredicted results. The artist himself has the advantage of having access to the previous conceptual representation, and he can form a new model based on the current temporary material form of the work, and check if it corresponds to his original idea. On the other hand, this is not so easy, since the artist is so deeply engaged in the work that he cannot judge it like someone from the outside. For this reason, artists use various tricks, e.g. to let a work rest for a while, and start anew with fresh ears, or observing a painting upside down to fool perception and prejudice.

The self-interpretation and subsequent evaluation can be done rarely, to let a generative process finish. Or it can be done often, or even continuously, but this can obstruct the creative flow. Postponed judgement is liberating, as described so well by Nachmanovitch (1990) and many others.

In the process, we seek the intersections between the tool networks and the current conceptual subspace. When I paint with a brush I seek intersections between the network of the tool—the images that I am able to paint, and the conceptual subspace formed by the ideas I want to convey. These intersections have a kind of gravity. We are attracted to them, and this is where the final version of the work will be found—in an intersection between the idea networks and the tool networks—something that is realisable and contains a representation of our ideas. This is a feedback process. I observe what I do, see if it fits the concept, or if it can fit a slightly modified concept, then redo or continue to work on the image. The conceptual subspace changes because of the modifications, and the material representation change because of my actions. When the conceptual and material representations have converged, the work may be considered finished.

8.3.4 Example Scenarios

The creative process as a back-and-forth translation between material and conceptual representation is illustrated by the very simple example in Fig. 8.6, and by the following examples from my own works:

(1) When composing, I might have the idea to try to overlap certain rhythmic and melodic patterns of different lengths, a medieval technique called isorythm. I cannot predict the output in any other way than by implementing it, generating a sketch score—the first material form. It turns out OK, but at many places the two voices collide in an interesting manner, hinting at unusual chords, and sometimes the two voices combine into a single pregnant motive, where the two voices intertwine. I revise and extend my original concept to involve elaboration of these coincidences.

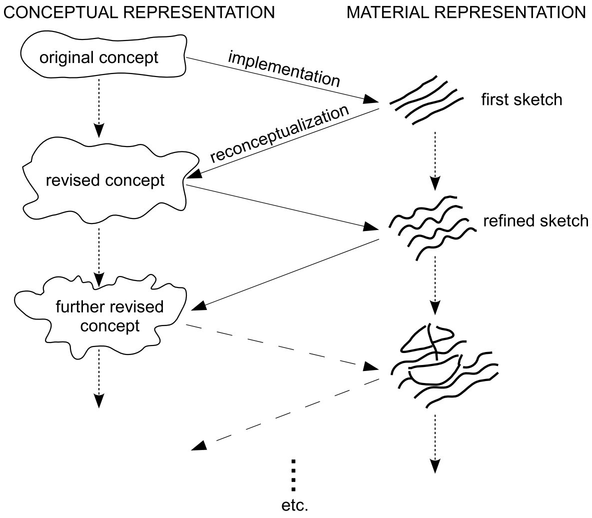


Fig. 8.6 The creative process as an iteration between idea and material. In each step, the conceptual representation is revised based on the previous material result. This particular process could be described like this: Draw something, maybe a few diagonal lines. One became a little wavy by accident. Hmmm...let's make them all wavy. Ah, that looks like the sea! Let's draw a boat, too. Interestingly, this trivial example came about exactly like that

I implement this, extrapolating from them in each place, modifying the formally derived skeleton, arriving at a new material form. This extrapolation at some places triggers new coincidences, which make their way into the conceptual representation, and so on, until I am satisfied with the result, i.e. there is no room for more elaboration, or all coincidences have been elaborated upon. The above is a true account of how I composed my own *Wedding March* for church organ, in 1999.

(2) When doing a free improvisation at the piano, I might just start and let the first notes be formed by pre-learnt ways of the hand, as described by Sudnow (2001), or by unconscious ideas. Hence, my initial conceptual representation is empty. But tool-based, cultural and physiological constraints guide my actions. These form topological networks and subspaces in the material space. The material representation is the sounds I hear, and I immediately try to make sense of it, to form a conceptual representation of what actually happened—because I am not always consciously aware of what I am playing. I might detect a certain rising figure in the left hand, or a particular unintended interval combination, and I choose to elaborate upon it. This is the re-conceptualisation step, and the next version of the conceptual representation is a variation on this idea. I perform it, but it does not end as I expected, because of my limited prediction capacity, and I re-conceptualise what happened, perform it, and so on. This is a real-time accumulative process, and in this case, the creative process unfolds right in front of the listener. The conceptual basis for the music emerges from the complex interplay between what I happen to play and what I extract from it, gradually becoming a composed whole, since for each new step in the process, there is more existing material to relate to in the re-conceptualisation.

The conceptual representation is nil to start with, but implicitly it may express itself in terms of a feeling or a state of mind, that affects what is emphasised in the reconceptualisation. You see or hear what you can relate to (by affinity and knowledge), subconscious states and associations are projected onto what "just happened" and gradually take shape in the iteration between what you hear and what you chose to elaborate upon in your playing. For example, as an improvising pianist, I tend not to relate to advanced jazz harmony implied by other players, because it is not part of my personal musical heritage. Instead, I concentrate on structures, gestures and note-to-note relationships, and extract conceptual representations from them, to elaborate upon in my response.

If I was instead improvising a drawing on an empty paper, the scenario would be similar. Improvised theatre can also work in this way[2]—you have to accept what has been done, extrapolate from it and build something on top, this time together with others. You see hints of meaning, possibly unintended, in the emerging material, and you enhance and clarify with modifications, which in turn provide new departure points for your co-players.

8.3.5 Appreciation and Novelty

Many factors affect our traversal of the material space. In addition to the interplay between conceptual and material representations, there are also factors such as cultural knowledge, of expectations and appreciation. We have learnt to recognise and appreciate certain subregions of the space, and there might be a pressure from the outside about what kind of art to produce, what conceptual contents to depict, which techniques to use, etc. This is evident when such constraints are unconsciously included in a conceptual representation, only to be realised when we are confronted with something that is "wrong" with respect to this property. It is so deeply embedded in our cultural heritage or social expectations that we did not realise that it was there as a constraint.

An artwork is not interesting *per se*. It is interesting in relation to something, to what has previously been written and said within that field. The interest is in what it adds to that, what it contradicts, and how it may provide food for new thoughts within the field. The cultural baggage of the artist acts as a guiding force in the creative process—it determines acceptable regions in the space of the possible, because it defines what the artist considers art, as understandable and interesting, and hence constrains his conceptual representations. By continuing a bit further on these paths, or deviating from them (but in relation to them), he creates something new, based on what was before.

Appreciation is an interesting phenomenon. It often coincides with the moving edge of an expanding conceptual network, and the corresponding material subspaces. New art has to connect in some way to this, and it can possibly go beyond the edge of a conceptual network a little bit. If it is completely within existing networks, it is uninteresting. If it is completely outside, it is difficult to relate to—there

[2] In the 1990s, I worked as an improvising musician with a theatre group, participating extensively in this kind of emerging performances.

is no path to get there, and it is conceptually disconnected from my existing networks. If it strikes the right balance with respect to the receiving individual, the new work extends his networks too, and forms a foundation for further extensions, further curious explorations in our continuous strive for novelty.

Novelty in creativity is often divided into Boden's private P-creativity and historical H-creativity (Boden 2004). Is it new for me, or new for a whole cultural domain, or even all of humanity? Novelty in relation to myself includes the expansion of my networks to gradually encompass new areas. Tool networks and conceptual networks both contribute to this process. For some artists the tool networks lead into new areas. For others, the ideas take the lead, and the tool networks are expanded as needed. The former is more common in music, as an abstract art, where tools and techniques play an important role. The latter may be more common in contemporary fine arts, where real-world concepts often are of primary importance.

When a conceptual representation develops, it expands to new areas in interplay with continually accumulating cultural input, pushing the individual artist to expand further, past what has been covered by others. Nobody has access to a global database, but only to the fraction of human culture contained in the artist's memory, defining his cultural networks and subspaces. To put it simply, his exploration of spaces happens in interplay with what he remembers from what he has seen or heard before.

8.3.6 The Model in Context

There are empirical studies of creative processes within psychology research (e.g. Barron 1972, Konecni 1991) and abundant recollections on the subject from artists (e.g. Barron 1997, Klein and Ödman 2003). These accounts from artists are sometimes contradictory and personal, and concentrate on rhapsodical and very personal details of particular processes. Artists not aware of existing psychological theories of creativity, may not be able to give a systematic account of what is happening. They sometimes reconfirm well-known phenomena and myths, but hesitate, consciously or not, to reveal their creative techniques, or are not able to verbalise the mechanisms of their own creativity. Some seem to preserve the romantic mystery around creativity. And since not all researchers have first-hand access to these processes (since they are not professional artists themselves) computational implementations directly derived from artists' processes are rare, with a few notable exceptions. Harold Cohen's autonomous painting program *AARON*, is based on his own analysis of how he went about composing and painting a picture, from sketching the first lines down to colouring. It works very well, within a limited domain of certain motives (McCorduck 1990). In the field of music, David Cope is well-known for his advanced computer-generated pastiches of classical music. Recently, he has changed direction, and developed a composing program called *Emily Howell* (Cope 2005), which develops its own musical style in a dialogue with Cope himself. In this case external musical input and human feedback gradually helps form the stylistic character of the program. Cope, himself a composer, has based his model on careful analysis of musical creativity; he stresses concepts such as allusion and analogy, and his model is based on complex associative networks between musical components, similar to that which humans develop through extensive listening. Cohen and Cope both emphasise process—in Cope's case also in a longer perspective, between works—but neither explicitly describe their models in spatial terms.

My proposed model is certainly not the first spatial theory of creativity, but it extends previous theories significantly (most notably Boden's, 2004) by introducing the idea of a material space, linked by the dynamic interplay between different descriptive levels—the conceptual and material representation of the work. The model of course relies on many previous results from peers, and various parts of it are related to previous theories. For example, Pearce and Wiggins (2002) provide a link between psychological research and the question of compositional creative processes, giving a rather detailed account of the cognitive aspects of musical composition. However, they do not dive deeper into the actual processes of composition itself.

Many previous attempts have focused on a formal approach, with the explicit generation of new ideas as the primary aim. In contrast, I believe that new ideas emerge from the process, and primarily from the iterated reconceptualisation and implementation, allowing for ambiguity, misunderstanding, associations and coincidences to contribute to the generation of new ideas and artistic results. This is a very rich process, involving all aspects of the artists mind, his cultural context, and of the material he is working in, with plenty of possibilities for unexpected results, leading to radically revised and new ideas.

The idea of iterated conceptual representations is related to Liane Gabora's work. She says:

> Creative thought is more a matter of honing in a vague idea through redescribing successive iterations of it from different real or imagined perspectives; in other words, actualising potential through exposure to different contexts. (Gabora 2005)

This also resonates well with Harrison's (1978) ideas about creativity being a goalless, non-rational process. Understanding of the re-conceptualisation mechanism could also be informed by a closer study of Karmiloff-Smith's (1994) thoughts on representational re-description in a developing mind, where knowledge gradually is transformed from simple procedural descriptions into conceptual contraptions of a higher level.

My model also transcends the distinction between exploratory, combinatorial and transformational creativity, for several reasons. The search space has been extended to the whole material space of the chosen medium, which includes all theoretical possibilities. A search in such a space equals a generative process, and is neither simply combinatorial nor transformational. Maybe it could be described as being based on *processual emergence*. This relates to Wiggins's (2006) idea that the search strategy can be more crucial than the definition of the conceptual space. He also presents a few theoretical devices for revising it depending on the results. In my model, the conceptual network is continuously being transformed as new conceptual dimensions—along which change can happen—appear from the re-conceptualisation.

The idea of including the material space in the model is related to McCormack's (2005) claim, that for true open-endedness, generative systems and their outcome must reside in the same domain, as is the case with evolutionary creativity in nature where genome, phenotype and developmental mechanisms share the physical environment. It is however unclear if this applies to human creativity in art, but it certainly would allow for more complexity and openness in the process, which is what the material representation adds in my model.

The importance of a material representation became clear to me when confronted with unexpectedly interesting musical results from my autonomous evolutionary composition system *Ossia* (Dahlstedt 2004; 2012), where configurations of notes resulting from separate branches of a generative processes are brought back into the same process in a finite recursion, allowing structural re-use of coincidental material. Essentially, temporary results from branch nodes of a generative tree structure are fed back to the leaves, forming a rudimentary iteration between concept and material. A generative representation provides access to high-level variation, but if used without a material representation, it may not permit behaviour as complex and unpredictable as human creativity.

The material representation is also, in most cases, the shared layer between creative agents in the human world. If we had direct access to each other's conceptual representations, we could have a perfect transfer of thoughts between people, removing a lot of the complexity from human culture. A hypothetical scenario, yes, but it illustrates the point. Continuous need for interpretation, and the unavoidable misunderstandings that follow, are quintessential properties of a society of creative agents. One of my points with this model, is to show that this is also happening in single-agent creative processes, in a type of dialogue between artist and the material.

This possible sharing of material is also interesting in another way. It allows for explanation of processes where the material is revised by someone else, e.g. a teaching situation, or when someone is tinkering with the artist's material. Hence, it is robust and flexible, fitting a number of scenarios that feature multiple creative agents.

8.3.7 Craft and Skill

If you know a tool well, you are able to predict the result of your actions, based on training and experience from the application of the tool in many different contexts and situations, and because you have a well developed cognitive model of the tool. Then, the tool network is fine-meshed. You can make a qualified guess about what is possible and what is not possible before you do it. When you navigate along the conceptual network, you adjust according to tool networks. Out of necessity, you sometimes adjust the idea so that it becomes possible to realise, i.e. so that the conceptual subspace intersect with a tool network. This is often possible without sacrificing any important part of the idea. Sometimes it actually adds something, since it forces you to deviate from your beaten tracks. If the tool network is sparse, due to lack of training or coarseness of the tool, it becomes more difficult to find these intersections. You might try to fill in the tool network when you have found a point you want to realise, by learning new tools, learn a tool better, or ask help from someone else.

Also, the better you know your tools, the more they become integrated in your conceptual thinking, and the tool networks may even overlap with the conceptual networks to a certain degree, because your concept may be constructed from your tools. This is especially evident in music, where abstract generative principles may be the main conceptual ideas behind a work, and at the same time the tools to create it.

8.3.8 New Tools and Tool Design

Especially in electronic music, there is a strong focus on the development of new tools, such as new synthesis techniques, signal processing algorithms and new physical interfaces to control the music. Why is that? And why do we need to learn new tools? A new tool might offer more precise manoeuvrability in certain regions of the material space, or let us reach completely new, hitherto unknown regions. It might take us faster to known regions, and hence push the limit of the possible, within a given time frame or within our cognitive capacity, by extending it—the tool embodies intelligent behaviour and thus enables new lines of thought. A new tool also creates new structural relationships, which will unfailingly be exploited in new artworks. If you can get from A to B in a new way during a compositional process, this can be used to create internal references within a musical piece, for example, and will eventually affect the cultural network through repertoire.

For example, tonal harmony as an organising principle dominated Western music until the early 20th century, in gradually more complex forms. All compositions were placed and composed along these networks in the space of possible music. When this constraint was removed (by Schoenberg and others), it was impossible to just start thinking freely. The minds of composers were literally wired along this network of tonal harmony, in addition to others of style, form and expression. New tools were needed, to provide pathways for composers' imagination and for the creative process. Most influential was the twelve-tone idea (no chromatic note must be repeated until all others have been heard) and serialism (the use of tone-rows and their various permutations and transpositions). They provided a framework for exploration of the unknown space outside the traditional tonal network. After some time, composers became more accustomed to these new modes of expression, and the tools became incorporated into cognitive and conceptual networks, with less explicit focus on the actual generative principles, and more on the sounding results. Some composers were able to compose aurally in the style of twelve-tone music, as described by Valkare (1997). If some other principles had been presented instead of twelve-tone serialism, the results would have been very different, in terms of both the music and the imagination of the composers. So, the development of tools is an essential part of the continuous discussion about what can be created, and what can be expressed—and this discussion is what I call art.

8.3.9 Social and Cultural Creativity

The discussion in this chapter has focused on the individual creative process, even though cultural aspects have been implicitly mentioned in terms of networks formed by cultural heritage in the material space. But we can see the advantage of this model also in analysis of collective creative activities, both real-time exchanges such as musical improvisation, or in slower processes such as the general artistic discourse within a particular field. Let us look at some examples.

In group improvisation, musicians communicate through the material representation, i.e. the sonic result, communicated through the air. This is possible thanks to the amazing human ability to interpret sound into conceptual musical structures. Once again, creative misunderstandings during this process will result, since the music is always ambiguous. Each musician makes his own re-conceptualisation of what is happening, and reacts to that musically, contributing to the continued material shape of the work.

In non-real-time activities based on verbal discussion, such as collaborative works, or a continuous artistic discourse, we communicate through conceptual representations, exchanging and developing ideas, but also through material results. And misunderstandings and re-conceptualisations thereof form the basis for new ideas.

This is interesting, because different individuals carry different networks, regarding concepts, tools, cognition and perception. The re-interpretation of a temporary result, an artwork or a musical output by someone else, can modify the concept in an unexpected direction, i.e. adjust it to fit his networks, so that he can develop it further along pathways available to him. When the originator is confronted with this re-interpretation, his own network can grow, to also include this kind of output. In this way, we learn from each other, in a continuous development of ideas.

8.3.10 Abstraction Levels

One aspect that has not been directly discussed is the problem of sketches as temporary material form. Sketches are in themselves conceptual and imprecise, but still more precise than the original thoughts that inspired them. The sketch is somewhere between the conceptual representation in your head and the final material result. In many domains, such as drawing, sketches are intentionally vague to allow the testing of ideas without requiring the development of complete detail. How can we account for this? A similar case is the various forms of concept-based artforms, where the final medium for the artwork is ideas. But I suggest that the proposed model can also be applied in these cases. A sketch can still be regarded as a material form in relation to a more abstract conceptual representation. It is the difference in level between the two representations that is important, and the interplay between them when going back and forth—not the exact nature of the material representation. In the case of score-based music, for example, the material representation (the score) is somewhere in between the conceptual and the material level. In the case of concept-based art, we can still think of different conceptual levels, with a number of idea-based tools (idea generation, idea transformation, refinement, deduction, induction, contradiction, etc.) that the artist can use to develop the final work. There are two abstraction levels, and an interplay between them.

The actual material level may also change in the course of the process. First I may work with an interplay between concepts in my head and sketches on paper as the material form. Later, when I am content with the sketches, I proceed to a level where the concept in the head, as formalised by sketches, interplays with the final material medium. Maybe any differences in degree of abstraction between representations would suffice for a creative process, and the transfers between them account for the complexity of the process?

8.4 Implications for Computational Creativity

Many experiments in computational creativity have been implemented within the traditional artificial intelligence (AI) paradigm, using techniques such as symbolic reasoning, knowledge-based systems, statistical models and heuristic search. They usually operate within a restricted domain, and the form of the search target is often strictly defined—a solution to a well-defined problem, a postulate that matches given data, etc. (for a couple of examples, see Lenat 1983, Lindsay et al. 1980). There is an awareness of these problems, and one proposed solution is to add meta-level reasoning to affect the process and domain itself (see e.g. Buchanan 2001). However, the approach at that level is of the same formal nature as the previous one, equally distant from how we think, and from the complexity of real life. And the tasks chosen for modelling are often of a scale that would not be considered particularly creative if performed by humans, such as the harmonisation of a Bach-style chorale (e.g. Ebcioglu 1988; see Papadopoulos and Wiggins 1999 for an overview of similar projects). They are reasonably complex search processes, yes, but more like optimisation processes than an exploration to extend our conceptual world. The form of the solution is known beforehand, and it will give us no surprises.

When going through the AI creativity literature, there is a lack of attention to process as a source for novelty and complexity. The AI approaches are mostly based on logical analysis of the concept of creativity and novelty, and not how a human artist goes about when creating something, at least not the creative processes I can observe in my own artistic practice. As an artist, I seldom know what I am looking for. Sudden ideas are often related to the domain I am working in, but I do not know exactly *what* idea I am searching for. Coincidences play a major role in triggering specific ideas and in shaping more complex creative output.

Maybe the most successful approach so far (according to Boden 2004) has been the use of evolutionary algorithms, i.e. simplified emulations of Darwinian evolution applied to data representations, as search techniques in open-ended conceptual spaces, inspired by nature's creativity. The numerous examples include works by Sims (1991), Todd and Werner (1999), Jacob (1996) and myself (Dahlstedt 2004; 2007; 2009b).

Well implemented evolutionary systems are capable of generating interesting novelty; they can be creative in a sense. But there are several problems with this approach. Firstly, while evolution is good at searching spaces, it has been difficult to design really open ended systems. Secondly, the kind of creativity it exhibits is not very similar to human artistic creativity. It uses blind variation directly on the genetic representation, which corresponds to the conceptual representation in my model. In artistic creativity, the variation is instead inferred by extracting a new conceptual representation from the current material form in whatever way this came to be. To understand human creativity, I think we need to base our implementations on a model of human creativity, and not on natural evolution. Evolution is one example of how new things or ideas *can* be created, but maybe not how *we* create. See Gabora (2005) for further discussion about this distinction.

In this context it might be interesting to consider two completely different types of creative processes, both existing in nature, but in different domains. The first is the reiteration of a particular generative process until it is "just right", with evaluation only of quasi-complete results. This is analogous to natural evolution, where each new individual is developed all the way from the blueprint, in each generation. From this perspective every living thing is a generative artwork. The other alternative is the accumulated treatment and processing of a temporary form, exemplified by natural structures such as mountains, rocks and terrain. They record their own history of coming into being, through generative and erosive processes. We may call these *generative* and *accumulative* creative processes. So, one is typical of living things, the other of dead matter exposed to additive, transformative, and destructive processes. Both can be accounted for by the proposed model, with different frequency of re-conceptualisation, and both types of process exist in art. I would say that the accumulative process is a crucial part of human artistic creativity, with the exception of explicitly generative art. Evolutionary algorithms, as powerful as the may be, are limited to generative creative processes, which may indicate that they are not entirely suitable for emulation of artistic creativity.

8.4.1 Implementation of the Model

Implementing the proposed model involves several difficult and challenging problems. They are discussed below, with some preliminary speculation about possible initial approaches.

To fully model human creativity, we would need to successfully model most essential features of the human mind, which is of course impractical. However, there are strategies to make this seemingly impossible problem at least partially tractable. One way is to look for the simplest possible implementation of each required component, still being sufficiently complex for the overall emergent creative behaviour to appear. Certain core features of each component may suffice to arrive at interesting results. It is a research problem in itself to find this level, though, as discussed by Cope (2005). But the more minimal the implementations are—while still functioning—the more general conclusions we can draw.

There are two hard problems involved. Firstly, how do we implement suitable conceptual representations? Secondly, there is the related problem of how to implement re-conceptualisation from material form into new conceptual models. I have stressed the importance of misunderstandings in the parsing process, since they help form a personal expression. Then a rather simple re-conceptualisation model might suffice to start with, or a combination of simple models running in parallel, to widen the repertoire of recognised material. Each model interprets the given material in a particular way, and the choice of models will contribute to the "personality" of the algorithm, in the same way as the characteristic shortcomings of a human artist contribute to his personal style.

8.4.2 Conceptual Representations

Knowledge and concept representation has always been a problem in computing. The conceptual representations in this model need to be flexible and open-ended, but we want to avoid the symbolic approach of traditional AI, for reasons explained earlier. While Pearce and Wiggins (2002) mention the ability to represent musical material as a hierarchical structure (see also Dahlstedt 2004; 2005), McCormack (2005) states that representation and generative mechanism should be on the same level as the material resulting from the process, hence a collapse of hierarchies. This is an important point, and I believe the iterated process between material form and conceptual form bridge this gap between levels, and provides a path between them in both directions.

A conceptual representation has two components: a description of what we want to achieve (e.g. desired properties, list of constraints), and a description of what we want to do (a generative procedure or list of tool actions). Let us call them *description* and *instruction*. In a goal-driven creative process, with a clear vision of the final form of the work, the description component is more important. But the clearest vision may be revised if something unexpected but interesting is found. Also, a determined idea about the description may still lack sufficient detail to form the basis of a full artwork. Hence, flexibility is still needed. On the other hand, a work based primarily on a generative idea may lack an description component, and instead give more weight to procedural instructions. In free improvisation, description may initially be empty, and both are open for change, according to how the process unfolds. So, both components are needed, in a weighted combination, to cover a wide range of processes.

As the process proceeds, the conceptual representation could also include accumulating existing parts of the material form of the work. As an example, consider that when a painting is finished, all we have is the actual material form of the work—the conceptual representation is gradually transformed into a material representation during the creative process. It is then up to the viewer to form his own conceptual representation of it.

8.4.3 Re-conceptualisation

The process of re-conceptualisation is a parallel to what an art-consumer does when observing an artwork—looking at the material result, possibly trying to recreate the process and the concepts behind it. Since the material form *is* the artwork, as it appears to others, nothing can be ignored: faults, context, imbalances, and so on. With both description and instruction included in the conceptual representation, the re-conceptualisation process would consist of an evaluation and modification of the previous conceptual representation, with respect to the material result. The process could involve, for example, perceptually based fuzzy pattern matching and feature detection, such as detecting entities, transitions and regularities in the material. There may be different kinds of discrepancy between description and material:

- Feature extraction may recognise a pregnant idea that is the result of a coincidence between results produced by different parts of the instruction;
- Emergent features in the material may not be explicitly represented in the conceptual representation;
- Computational or human mistakes may have distorted the result;
- The conceptual representation may not be visible at all, due to ambiguity, complexity and the nature of the generative process (an irreversible many-to-one mapping), and the re-conceptualisation step will have to be carried out from scratch.

A useful strategy for the implementation of re-conceptualisation would be to use double-linked representations, with pointers between the part of the implementation components of the conceptual representation and the material result, in both directions. In the material representation, a layer of pointers tell which part of the conceptual representation was involved in generating it: e.g. a node, an object, a branch of a tree, a block of generative code. This could help indicate coincidental material, as detected features or entities consisting of material emanating from widely separate parts of the conceptual representation. Borders between results from subpart of the conceptual representation can help distinguish entities that could form a basis for the next iteration of the conceptual representation, or help indicate which part of the conceptual representation needs modification. However, if implemented too strictly, this could counteract the idea of creative misunderstandings in parsing. But with overlapping of material coming from different parts of the conceptual representation, it may still allow sufficient ambiguity, since the pointers for overlapping material will make things more complex—it might not be, or should not be, a one-to-one mapping.

Such double-linking is probably not possible with all kinds of representations, but in the cases where it is applicable, it can provide valuable information about the morphological relationship between concept and material.

After detection of discrepancies, the conceptual representation needs to be revised, in one of the following ways:

- Extension/addition: adding new details or new source material (themes, motives, source images, "constants");
- Extension/generalisation: conceptually generalising the scope of the representation, e.g. when confronting coincidental material, extracting their core properties and including them in the next representation, to minimise the risk of losing them in subsequent iterations. Or when stagnating, remove hindering constraints, and backtrack;
- Variation: when the conceptual representation is tilted, shifted, or mutated, depending on the form of the conceptual representation;
- Restriction/narrowing: adding further constraints or removing unwanted material;
- Association: when something that resembles some pre-existing material or concept is replaced by a clearer reference, or related material from the same source is added;
- Replacement: when reaching a dead end or when the temporary form is completely reinterpreted into something else, the whole conceptual representation needs to be replaced.

Local implementations of heuristic search, such as evolutionary algorithms or hill climbing, could be used within the component of re-conceptualisation in order to find suitable modifications. As long as these techniques are kept within this component, any shortcomings should not influence the overall process.

8.4.4 Memory and Learning

A fundamental problem of computer-generated art is its relation to the real world—the cultural and semantic content. To be appreciated by humans and to produce meaningful content of any depth, the system must have access to the outside world. This link could be provided on a rudimentary level by an intentionally imperfect associative memory model, where previously experienced material is retrieved by association and incorporated into the creative process. This would account for cultural constraints and generative aspects of culture, such as references, metaphors (Hofstadter et al. 1995) and associations (Mednick 1962). It is also strongly emphasised by Cope (2005). Such a memory model should also include memories of previous results from the system, and in this case it could hold both material fragments and the conceptual representations behind them.

When designing computational models of creativity, it is an important question how to evaluate the success of the implementation and hence how to draw any conclusions from the experiments. As I see it, there are two different cases. Either you make an implementation that tries to generate art that is credible and interesting to human observers, or you make a minimal model and evaluate it if it exhibits the right kind of emergent behaviour in relation to its own context, i.e. its limited amount of data about the outside world.

One could argue that a computational model that produces results interesting to human observers, having been exposed to a very small amount of human art, could not be a faithful model. A human with such limited experiences and limited contact with the outside world would certainly not produce very interesting art. So, the creator of such a system simply must have put in a lot of his own experience into the model, consciously or not, in the form of informed design choices.

On the other hand it is difficult to evaluate a minimal model, since it will not be possible to judge the output as art in itself. The novelty, complexity and interest of it must be valued in some way in relation to the context and scope of the program. Creative behaviour may be there, but we may fail to recognise it, since it does not appear as anything we are used to. Only if a computational model can perceive and internalise substantial amounts of humanly produced artistic material, combined with human feedback on its own output over an extended time, can we judge the output by human standards.

8.5 Final Remarks

I have presented a model of creative artistic processes that is founded on artistic practice. There is a long way to go before it can be implemented fully in software, but it is my firm belief that it could help us create more believable artistic results and behaviour from computational creative systems, and it may form a foundation for discussion, analysis and increased understanding of human creative processes. In my own artistic practice as composer and improviser, I can clearly see how it fits a wide variety of creative activities, and I present it here to be tested by others in relation to their experiences from art practice and in future computational implementations.

Though preliminary, the model already provides a framework for analysis, discussion and possibly emulation of a number of important concepts and phenomena directly related to creativity:

- of the relationship between the theoretically, practically and conceptually possible; between material, tools and ideas;
- of the relationship between the artist and his tools;
- of ideas, concepts and generative processes as guiding mechanisms for realisation of a work;
- of choices, and how we navigate the space of the possible;

- of the realisation of a work as a non-linear process;
- of our cognitive preconditions—our ability to structurally interpret material, to create variation, to see connections between different parts of the space of the possible, and to find or design tools that take us there;
- of re-conceptualisation as an essential part of the iterated process of realising a work;
- of personal style as characteristics of the personal topologies in material space.

In this chapter, I have mostly discussed how we go about realising the artistic artefact and give it form in a particular framework or context. The model does not cover what *we* want to express, depict or give form to as artists, or the value of the outcome, which is included in some definitions of creativity. According to this view, the result not only has to be new or novel, but valued by the community where it appears—or else it is not judged as creative. If we speak about value as in good art vs. bad art, then value is not intrinsic in the work, but relative to the observer. It lies in the consistency of ideas, depth and detail of implementation; in the relevance to the observer of the ideas conveyed. As long as it can provide an adequately complex reflective surface for the observer, to enable her to make her own re-conceptualisation and arrive at something which resonates with her thoughts, it can be good art. I think this kind of value and meaning in a computer-generated artwork may emerge from a faithfully implemented creative process.

Based on thorough observation of my own creative processes, and experience from artistic teaching, from development of creative tools, and from my research into applications of creative algorithms, I am quite convinced that the proposed model could provide the basis for such implementations, providing a deeper understanding of artistic creative processes: in humans and in machines.

Acknowledgements A major part of the research behind this chapter was funded by a research grant from the Swedish Research Council, for the project "Potential Music".

References

Barron, F. (1972). *Artists in the making*. New York: Seminar Press.

Barron, F. (Ed.) (1997). *Creators on creating*. Tarcher.

Boden, M. (2004). *The creative mind: myths and mechanisms* (2nd ed.). London: Routledge.

Buchanan, B. G. (2001). Creativity at the meta-level: AAAI 2000 presidential address. *AI Magazine Volume*, *22*(3), 13–28.

Cope, D. (2005). *Computer models of musical creativity*. Cambridge: MIT Press.

Dahlstedt, P. (2001). A MutaSynth in parameter space: interactive composition through evolution. *Organised Sound*, *6*(2), 121–124.

Dahlstedt, P. (2004). *Sounds unheard of: evolutionary algorithms as creative tools for the contemporary composer*. PhD thesis, Chalmers University of Technology.

Dahlstedt, P. (2005). *Defining spaces of potential art: the significance of representation in computer-aided creativity*. Paper presented at the description & creativity conference, King's College, Cambridge, UK, 3–5 July 2005

Dahlstedt, P. (2007) Evolution in creative sound design. In E. R. Miranda & J. A. Biles (Eds.), *Evolutionary computer music* (pp. 79–99). London: Springer.

Dahlstedt, P. (2009a). Ideas and tools in material space—an extended spatial model of creativity. In M. Boden, M. d'Inverno & J. McCormack (Eds.), *Dagstuhl seminar proceedings: Vol. 09291. Computational creativity: an interdisciplinary approach*. Dagstuhl: Schloss Dagstuhl—Leibniz-Zentrum fuer Informatik. http://drops.dagstuhl.de/opus/volltexte/2009/2198.

Dahlstedt, P. (2009b). Thoughts on creative evolution: a meta-generative approach to composition. *Contemporary Music Review*, *28*(1), 43–55.

Dahlstedt, P. (2012). Ossia II: autonomous evolution of complete piano pieces and performances. In *A-Life for music:music and computer models of living systems*. Middleton: A-R Editions.

Denton, R. (2004). The atheism tapes: Jonathan Miller in conversation (TV program), episode 6, interview with Daniel Dennet. London: BBC, TV program.

Ebcioglu, K. (1988). An expert system for harmonising four-part chorales. *Computer Music Journal*, *12*(3), 43–51.

Gabora, L. (2005). Creative thought as a non-Darwinian evolutionary process. *Journal of Creative Behavior*, *39*(4), 65–87.

Gibson, J. (1977). The theory of affordances. In R. Shaw & J. Bransford (Eds.), *Perceiving, acting and knowing*, Hillsdale: Erlbaum.

Gregory, R. L. (1981). *Mind in science*. London: Weidenfeld and Nicolson.

Harrison, A. (1978). *Making and thinking*. Harvester Press.

Hofstadter, D. (1985). Variations on a theme as the crux of creativity. In *Metamagical themas*. New York: Basic Books.

Hofstadter, D., & The Fluid Analogies Research Group (1995). *Fluid concepts and creative analogies: computer models of the fundamental mechanisms of thought*. New York: Basic Books.

Jacob, B. (1996). Algorithmic composition as a model of creativity. *Organised Sound*, *1*(3), 157–165.

Karmiloff-Smith, A. (1994). Beyond modularity: a developmental perspective on cognitive science. *Behavioral and Brain Sciences*, *17*(4), 693–745.

Klein, G., & Ödman, M. (Eds.) (2003). *Om kreativitet och flow*. Bromberg.

Konecni, V. J. (1991). Portraiture: an experimental study of the creative process. *Leonardo*, *24*(3).

Lenat, D. (1983). Eurisko: a program that learns new heuristics and domain concepts. *Artificial Intelligence*, *21*, 61–98.

Lindsay, R. K., Buchanan, B. G., Feigenbaum, E. A., & Lederberg, J. (1980). *Applications of artificial intelligence for organic chemistry: the DENDRAL project*. New York: McGraw-Hill.

McCorduck, P. (1990). *AARON'S CODE: meta-art, artificial intelligence, and the work of Harold Cohen*. New York: Freeman.

McCormack, J. (2005). Open problems in evolutionary music and art. In F. Rothlauf et al. (Eds.), *LNCS: Vol. 3449. EvoWorkshops 2005* (pp. 428–436). Berlin: Springer.

Mednick, S. A. (1962). The associative basis of the creative process. *Psychological Review*, *69*(3), 220–232.

Nachmanovitch, S. (1990). *Free play: improvisation in life and art*. New York: Jeremy P. Tarcher/Penguin-Putnam Publishing.

Norman, D. A. (1988). *The psychology of everyday things*. New York: Basic Books.

Papadopoulos, G., & Wiggins, G. (1999). AI methods for algorithmic composition: a survey, a critical view and future prospects. In A. Patrizio (Ed.), *Proceedings of the AISB'99 symposium on musical creativity*, Edinburgh, UK.

Pearce, M., & Wiggins, G. A. (2002). Aspects of a cognitive theory of creativity in musical composition. In *Proceedings of 2nd international workshop on creative systems, European conference on artificial intelligence*, Lyon, France.

Sims, K. (1991). Artificial evolution for computer graphics. In *ACM SIGGRAPH '91 conference proceedings*, Las Vegas, Nevada, July 1991 (pp. 319–328).

Todd, P. M., & Werner, G. M. (1999). Frankensteinian methods for evolutionary music composition. In N. Griffith & P. M. Todd (Eds.), *Musical networks: parallel distributed perception and performance*, Cambridge: MIT Press—Bradford Books.

Valkare, G. (1997). *Det audiografiska fältet: om musikens förhållande till skriften och den unge Bo Nilssons strategier*. PhD thesis, University of Gothenburg.
Wiggins, G. A. (2006). A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems*, *19*, 449–458.

Chapter 9
Computer Programming in the Creative Arts

Alex McLean and Geraint Wiggins

Abstract Computer programming is central to the digital arts, and is a comparatively new creative activity. We take an anthropocentric view of computer programming in the arts, examining how the creative process has been extended to include the authorship and execution of algorithms. The role of human perception in this process is a focus, contrasted and ultimately combined with a more usual linguistic view of programming. Practical impacts on the notation of programs in the arts are highlighted, both in terms of space and time, marking out this new domain for programming language design.

9.1 Introduction

Computer programming for the arts is a subject laden with misconceptions and far-flung claims. The perennial question of *authorship* is always with us: if a computer program outputs art, who has made it, the human or the machine? Positions on creativity through computer programming tend towards opposite poles, with outright denials at one end and outlandish claims at the other. The present contribution looks for clarity through a human-centric view of programming as a key activity behind computer art. We view the artist-programmer as engaged in an inner human relationship between perception, cognition and computation, and relate this to the notation and operation of their algorithms.

The history of computation is embedded in the history of humankind. Computation did not arrive with the machine: it is something that humans do. We did not invent computers: we invented machines to help us compute. Indeed, before the arrival of mechanical computers, "*computer*" was a job title for a human employed

A. McLean (✉)
Interdisciplinary Centre for Scientific Research in Music (ICSRiM), University of Leeds, Leeds, LS2 9JT, UK
e-mail: alex@slab.org

G. Wiggins
School of Electronic Engineering and Computer Science, Queen Mary, University of London, E1 4NS, London, UK
e-mail: geraint.wiggins@eecs.qmul.ac.uk

J. McCormack, M. d'Inverno (eds.), *Computers and Creativity*,
DOI 10.1007/978-3-642-31727-9_9,

to carry out calculations. In principle, these workers could compute anything that modern digital computers can, given enough pencils, paper and time.

The textile industry saw the first programmable machine to reach wide use: the head of the Jacquard loom, a technology still used today. Long strips of card are fed into the Jacquard head, which reads patterns punched into the card to guide intricate patterning of weaves. The Jacquard head does not itself compute, but was much admired by Charles Babbage, inspiring work on his mechanical *analytical engine* (Essinger 2004), the first conception of a programmable universal computer. Although Babbage did not succeed in building the analytical engine, his design includes a similar card input mechanism to the Jacquard head, but with punched patterns describing abstract calculations rather than textile weaves.

This early computer technology was later met with theoretical work in mathematics, such as Church's lambda calculus (Church 1941) and the Turing machine (Turing 1992, orig. 1947), which seeded the new field of computer science. Computer programmers may be exposed to these theoretical roots through their education, having great impact on their craft. As it is now practised, however, computer programming is far from a pure discipline, with influences including linguistics, engineering and architecture, as well as mathematics.

From these early beginnings programmers have pulled themselves up by their bootstraps, creating languages within languages in which great hierarchies of interacting systems are expressed. Much of this activity has been towards military, business or scientific ends. However, there are numerous examples of alternative programmer subcultures forming around fringe activity without obvious practical application. The Hacker culture at MIT was an early example (Levy 2002), a group of male model-railway enthusiasts and phone network hackers who dedicated their lives to exploring the possibilities of new computers, under the pay of the military. Many other programming cultures have since flourished. Particularly strong and long-lived is the *demoscene*, a youth culture engaged in pushing computer animation to the limits of available hardware, using novel algorithmic techniques to dazzling ends. The demoscene spans much of the globe but is particularly strong in Nordic countries, hosting annual meetings with thousands of participants (Polgár 2005).

Another, perhaps looser, programmer culture is that of Esoteric Programming Languages or *esolangs*, which Wikipedia defines as "programming language(s) designed as a test of the boundaries of computer programming language design, as a proof of concept, or as a joke". By pushing the boundaries of programming, esolangs provide insight into the constraints of mainstream programming languages. For example, *Piet* is a language notated with fluctuations of colour over a two dimensional matrix. Programs are generally parsed as one dimensional sequences, and colour is generally *secondary notation* (Blackwell and Green 2002) rather than primary syntax. Piet programs, such as that shown in Fig. 9.1, intentionally resemble abstract art, the language itself named after the modernist painter Piet Mondrian. We return to secondary notation, as well as practical use of two dimensional syntax in Sect. 9.4.

Members of the demoscene and esolang cultures do not necessarily self-identify as artists. However, early on, communities of experimental artists looking for new

Fig. 9.1 Source code written in the Piet language with two dimensional, colour syntax. Prints out the text "Hello, world!". Image © Thomas Schoch 2006. Used under the Creative Commons BY-SA 2.5 license

means of expression grew around computers as soon as access could be gained. In Great Britain, interest during the 1960s grew into the formation of the Computer Arts Society (CAS)[1] (Brown et al. 2009). However after a creative boom CAS entered a period of dormancy in the mid-1980s, perhaps drowned out by extensive commercial growth in the computer industry at that time. CAS has, however, been revived in more recent years, encouraged by a major resurgence of software as a medium for the arts. This has seen a wealth of new programming environments designed for artists and musicians, such as *Processing* (Reas and Fry 2007), *SuperCollider* (McCartney 2002), *ChucK* (Wang and Cook 2004), *VVVV* (http://vvvv.org) and *OpenFrameworks* (openframeworks.cc), joining more established environments such as the Patcher languages (Puckette 1988), *PureData* and *Max*. These have gained enthusiastic adoption outside a traditional base focused on academic institutions, and have proved useful for teaching the conceptual visualisation required to program computers.

Several artist-programmers have made their own, novel languages in which to make their art. These often seem like esoteric languages that have found practical application. For example unique representations of time are central features of ChucK and SuperCollider. Programming languages have themselves been exhibited as works of art, such as the *Al-Jazari* music programming environment shown in Fig. 9.2 (McLean et al. 2010). Programming languages made for artists have created new and emerging approaches to language design. This is not just a matter of technical achievement, but brings important psychological issues to the fore.

What is the relationship between an artist, their creative process, their program, and their artistic works? We will look for answers from perspectives of psychology, cognitive linguistics, computer science and computational creativity, but first from the perspective of an artist.

[1] www.computer-arts-society.org.

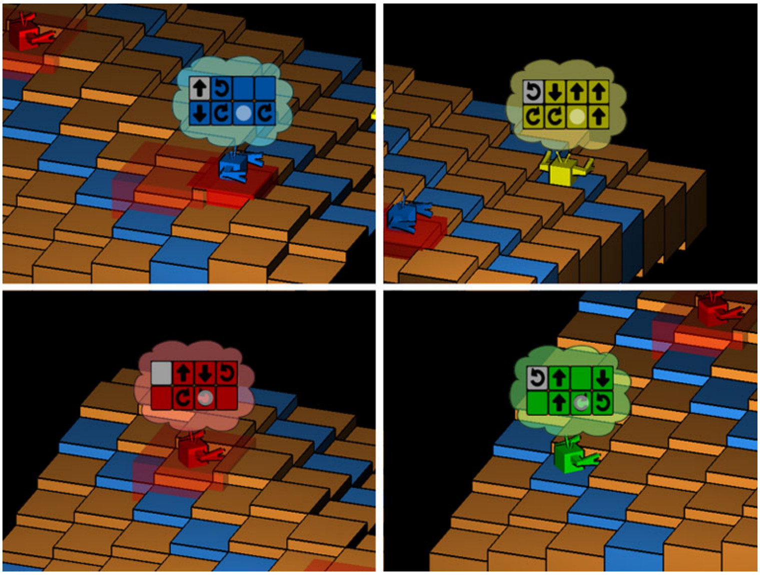


Fig. 9.2 The robots of the Al-Jazari language by Dave Griffiths (McLean et al. 2010). Each robot has a thought bubble containing a small program, edited through a game pad

9.2 Creative Processes

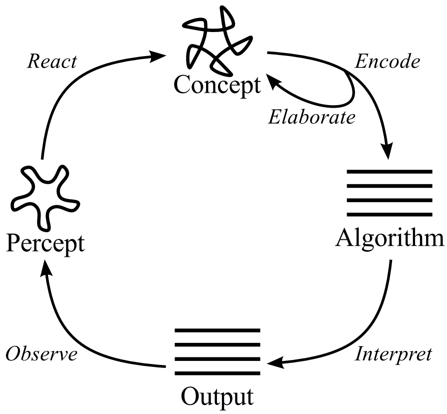
The painter Paul Klee describes a creative process as a feedback loop:

> Already at the very beginning of the productive act, shortly after the initial motion to create, occurs the first counter motion, the initial movement of receptivity. This means: the creator controls whether what he has produced so far is good. The work as human action (genesis) is productive as well as receptive. It is **continuity**. (Klee 1953, p. 33, original emphasis)

This is creativity without planning, a feedback loop of making a mark on canvas, perceiving the effect, and reacting with a further mark. Being engaged in a tight creative feedback loop places the artist close to their work, guiding an idea to unforeseeable conclusion through a flow of creative perception and action. Klee writes as a painter, working directly with his medium. Programmer-artists instead work using computer language as text representing their medium, and it might seem that this extra level of abstraction could hinder creative feedback. We will see however that this is not necessarily the case, beginning with the account of Turkle and Papert (1992), describing a *bricolage* approach (after Lévi-Strauss 1968) to programming by analogy with painting:

> The bricoleur resembles the painter who stands back between brushstrokes, looks at the canvas, and only after this contemplation, decides what to do next. Bricoleurs use a mastery of associations and interactions. For planners, mistakes are missteps; bricoleurs use a navigation of mid-course corrections. For planners, a program is an instrument for premeditated control; bricoleurs have goals but set out to realize them in the spirit of a collaborative venture with the machine. For planners, getting a program to work is like "saying one's piece";

Fig. 9.3 The process of action and reaction in bricolage programming



> for bricoleurs, it is more like a conversation than a monologue. (Turkle and Papert 1990, p. 136)

This concept of bricolage accords with Klee's account, and is also strongly related to that of the *reflective practice* (Schon 1984). This distinguishes the normal conception of knowledge, as gained through study of theory, from that which is learnt, applied and reflected upon while "in the work". Reflective practice has strong influences in professional training, particularly in the educational and medical fields. This suggests that the present discussion could have relevance beyond our focus on the arts.

Although Turkle and Papert address gender issues in computer education, this quote should not be misread as dividing all programmers into two types; while associating bricolage with feminine and planning with male traits (although note Blackwell 2006a), they are careful to state that these are extremes of a behavioural continuum. Indeed, programming style is clearly task specific: for example a project requiring a large team needs more planning than a short script written by the end user.

Bricolage programming seems particularly applicable to artistic activity, such as writing software to generate music, video animation or still images. Imagine a visual artist, programming their work using Processing. They may begin with an urge to draw superimposed curved lines, become interested in a tree-like structure they perceive in the output of their first implementation, and change their program to explore this new theme further. The addition of the algorithmic step would appear to affect their creative process as a whole, and we seek to understand how in the following.

9.2.1 Creative Process of Bricolage

Figure 9.3 characterises bricolage programming as a creative feedback loop encompassing the written algorithm, its interpretation, and the programmer's perception and reaction to its output or behaviour. Creative feedback loops are far from unique to programming, but the addition of the algorithmic component makes an additional inner loop explicit between the programmer and their text. At the beginning, the programmer may have a half-formed concept, which only reaches internal consistency through the process of being expressed as an algorithm. The inner loop is where the programmer elaborates upon their imagination of what might be, and the outer where this trajectory is grounded in the pragmatics of what they have actually made. Through this process both algorithm and concept are developed, until the programmer feels they accord with one another, or otherwise judges the creative process to be finished.

The lack of forward planning in bricolage programming means the feedback loop in Fig. 9.3 is self-guided, possibly leading the programmer away from their initial motivation. This straying is likely, as the possibility for surprise is high, particularly when shifting from the inner loop of implementation to the outer loop of perception. The output of a generative art process is rarely exactly what we intended, and we will later argue in Sect. 9.5 that this possibility of surprise is an important contribution to creativity.

Representations in the computer and the mind are evidently distinct from one another. Computer output evokes perception, but that percept will both exclude features that are explicit in the output and include features that are not, due to a range of effects including attention, knowledge and illusion. Equally, a human concept is distinct from a computer algorithm. Perhaps a program written in a declarative rather than imperative style is somewhat closer to a concept, being not an algorithm for how to carry out a task, but rather a description of what is to be done. But still, there is a clear line to be drawn between a string of discrete symbols in code, and the morass of both discrete and continuous representations which underlie cognition (Paivio 1990).

There is something curious about how the programmer's creative process spawns a second, computational one. In an apparent trade-off, the computational process is lacking in the broad cognitive abilities of its author, but is nonetheless both faster and more accurate at certain tasks by several orders of magnitude. It would seem that the programmer uses the programming language and its interpreter as a cognitive resource, augmenting their own abilities in line with the extended mind hypothesis (Clark 2008). We will revisit this issue within a formal framework in Sect. 9.5, after first looking more broadly at how we relate programming to human experience, and related issues of representation.

9.3 Anthropomorphism and Metaphor in Programming

Metaphor permeates our understanding of programming. Perhaps this is due to the abstract nature of computer programs, requiring metaphorical constructs ground programming language in everyday reasoning. Petre and Blackwell (1999) gave subjects programming tasks, and asked them to introspect upon their imagination

COMPONENTS ARE AGENTS OF ACTION IN A CAUSAL UNIVERSE.
PROGRAMS OPERATE IN HISTORICAL TIME.
PROGRAM STATE CAN BE MEASURED IN QUANTITATIVE TERMS.
COMPONENTS ARE MEMBERS OF A SOCIETY.
COMPONENTS OWN AND TRADE DATA.
COMPONENTS ARE SUBJECT TO LEGAL CONSTRAINTS.
METHOD CALLS ARE SPEECH ACTS.
COMPONENTS HAVE COMMUNICATIVE INTENT.
A COMPONENT HAS BELIEFS AND INTENTIONS.
COMPONENTS OBSERVE AND SEEK INFORMATION IN THE EXECUTION ENVIRONMENT.
COMPONENTS ARE SUBJECT TO MORAL AND AESTHETIC JUDGEMENT.
PROGRAMS OPERATE IN A SPATIAL WORLD WITH CONTAINMENT AND EXTENT.
EXECUTION IS A JOURNEY IN SOME LANDSCAPE.
PROGRAM LOGIC IS A PHYSICAL STRUCTURE, WITH MATERIAL PROPERTIES AND SUBJECT TO DECAY.
DATA IS A SUBSTANCE THAT FLOWS AND IS STORED.
TECHNICAL RELATIONSHIPS ARE VIOLENT ENCOUNTERS.
PROGRAMS CAN AUTHOR TEXTS.
PROGRAMS CAN CONSTRUCT DISPLAYS.
DATA IS A GENETIC, METABOLIZING LIFEFORM WITH BODY PARTS.
SOFTWARE TASKS AND BEHAVIOUR ARE DELEGATED BY AUTOMATICITY.
SOFTWARE EXISTS IN A CULTURAL/HISTORICAL CONTEXT.
SOFTWARE COMPONENTS ARE SOCIAL PROXIES FOR THEIR AUTHORS.

Fig. 9.4 Conceptual metaphors derived from analysis of Java library documentation by Blackwell (2006b). Program components are described metaphorically as actors with beliefs and intentions, rather than mechanical imperative or mathematical declarative models

while they worked. These self reports are rich and varied, including exploration of a landscape of solutions, dealing with interacting creatures, transforming a dance of symbols, hearing missing code as auditory buzzing, combinatorial graph operations, munching machines, dynamic mapping and conversation. While we cannot rely on these introspective reports as authoritative on the inner workings of the mind, the diversity of response hints at highly personalised creative processes, related to physical operations in visual or sonic environments. It would seem that a programmer uses metaphorical constructs defined largely by themselves and not by the computer languages they use. However mechanisms for sharing metaphor within a culture do exist. Blackwell (2006b) used corpus linguistic techniques on programming language documentation in order to investigate the conceptual systems of programmers, identifying a number of conceptual metaphors listed in Fig. 9.4. Rather than finding metaphors supporting a mechanical, mathematical or logical approach as you might expect, components were instead described as actors with beliefs and intentions, being social entities acting as proxies for their developers.

It would seem, then, that programmers understand the structure and operation of their programs by metaphorical relation to their experience as a human. Indeed the feedback loop described in Sect. 9.2 is by nature anthropomorphic; by embedding the development of an algorithm in a human creative process, the algorithm itself becomes a human expression. Dijkstra strongly opposed such approaches:

> I have now encountered programs wanting things, knowing things, expecting things, believing things, etc., and each time that gave rise to avoidable confusions. The analogy that underlies this personification is so shallow that it is not only misleading but also paralyzing. (Dijkstra 1988, p. 22)

Dijkstra's claim is that by focusing on the operation of algorithms, the programmer submits to a combinatorial explosion of possibilities for how a program might run; not every case can be covered, and so bugs result. He argues for a strict, declarative approach to computer science and programming in general, which he views as so radical that we should not associate it with our daily existence, or else limit its development and produce bad software.

The alternative view presented here is that metaphors necessarily structure our understanding of computation. This view is sympathetic to a common assumption in the field of cognitive linguistics, that our concepts are organised in relation to each other and to our bodies, through conceptual systems of metaphor (Lakoff and Johnson 1980). Software now permeates Western society, and is required to function reliably according to human perception of time and environment. Metaphors of software as human activity are therefore becoming ever more relevant.

9.4 Symbols and Space

We now turn our attention to how the components of the bricolage programming process shown in Fig. 9.3 are represented, in order to ground understanding of how they may interrelate. Building upon the anthropocentric view taken above, we propose that in bricolage programming, the human cognitive representation of programs centres around perception. Perception results in a low-dimensional representation of sensory input, giving us a somewhat coherent, spatial view of our environment. By spatial, we do not merely mean "in terms of physical objects"; rather, we speak in terms of features in the spaces of all possible tastes, sounds, tactile textures and so on. This scene is built through a process of dimensional reduction from tens of thousands of chemo-, photo-, mechano- and thermoreceptor signals. Algorithms on the other hand are represented in discrete symbolic sequences, as is their output, which must go through some form of digital-to-analogue conversion before being presented to our sensory apparatus, for example, as light from a monitor screen or sound pressure waves from speakers, triggering a process we call observation. Recall the programmer from Sect. 9.2, who saw something not represented in the algorithm or even in its output, but only in their own perception of the output; observation is itself a creative act.

The remaining component to be dealt with from Fig. 9.3 is that of programmers' concepts. A concept is "a mental representation of a class of things" (Murphy 2002, p. 5). Figure 9.3 shows concepts mediating between spatial perception and discrete algorithms, leading us to ask: are concepts represented more like spatial geometry, like percepts, or symbolic language, like algorithms? Our focus on metaphor leads us to take the former view, that conceptual representation is grounded in perception and the body. This view is taken from Conceptual Metaphor Theory (CMT) introduced by Lakoff and Johnson (1980), which proposes that concepts are primarily structured by metaphorical relations, the majority of which are *orientational*, understood relative to the human body in space or time. In other words, the conceptual system is grounded in the perceptual system. The expressive power of orientational metaphors is that it structures concepts not in terms of one another, but in terms of the orientation of the physical body. These metaphors allow concepts to be related to one another as part of a broad, largely coherent system.

Returning to Fig. 9.4, showing programming metaphors in the Java language, we find the whole class of orientational metaphors described as a single metaphor PROGRAMS OPERATE IN A SPATIAL WORLD WITH CONTAINMENT AND EXTENT. In line with CMT, we suggest this is a major understatement, that orientational metaphors structure the understanding of the majority of fundamental concepts. For example, a preliminary examination leads us to hypothesise that orientational metaphors such as ABSTRACTION IS UP and PROGRESS IS FORWARD would be consistent with this corpus, but further work is required.

Gärdenfors (2000) formalises orientational metaphor by further proposing that the semantic meanings of concepts, and the metaphorical relationships between them are represented as geometrical properties and relationships. Gärdenfors posits that concepts themselves are represented by geometric regions of low dimensional spaces, defined by quality dimensions. These dimensions are either mapped directly from, or structured by metaphorical relation to perceptual qualities. For example "red" and "blue" are regions in perceptual colour space, and the metaphoric semantics of concepts within the spaces of mood, temperature and importance may be defined relative to geometric relationships of such colours.

Gärdenforsian conceptual spaces are compelling when applied to concepts related to bodily perception, emotion and movement, and Forth et al. (2008) report early success in computational representations of conceptual spaces of musical rhythm and timbre, through reference to research in music perception. However, it is difficult to imagine taking a similar approach to computer programs. What would the quality dimensions of a geometrical space containing all computer programs be? There is no place to begin to answer this question; computer programs are linguistic in nature, and cannot be coherently mapped to a geometrical space grounded in perception.

For clarity, we turn once again to Gärdenfors (2000), who points out that spatial representation is not in opposition to linguistic representation; they are distinct but support one another. This is clear in computing, where hardware exists in our world of continuous space, but thanks to reliable electronics, conjures up a world of discrete computation. As we noted in the introduction, humans are able to conjure up this world too, for example by computing calculations in our head, or encoding concepts into phonetic movements of the vocal tract or alphabetic symbols on the page. We can think of ourselves as spatial beings able to simulate a discrete environment to conduct abstract thought and open channels of communication. On the other hand, a piece of computer software is able to simulate spatial environments, perhaps to host a game world or guide robotic movements, both of which may include some kind of model of human perception.

A related theory lending support to this view is that of *Dual Coding,* developed through rigorous empirical research by Paivio (1990). Humans have a capacity to simultaneously attend to both the discrete codes of language and the analogue codes of imagery. We are also able to reason by invoking quasi-perceptual states, for example by performing mental rotation in shape matching tasks (Shepard and Metzler 1971). Through studying such behaviour Paivio (1990) concludes that humans have a dual system of symbolic representation; an analogue system for relating to modes of perception, and a discrete system for the arbitrary, discrete codes of language. These systems are distinct but interrelate, with "high imagers" being those with high integration between their linguistic and quasi-perceptual symbolic systems (Vogel 2003).

Returning to our theme of programming, the above theories lead us to question the role of continuous representation in computer language. Computer language operates in the domain of abstraction and communication but in general does not at base include spatial semantics. Do programmers simply switch off a whole channel of perception to focus only on the discrete representation of code? It would appear not. In fact, spatial layout is an important feature of *secondary notation* in all mainstream programming languages (Blackwell and Green 2002), which generally allow programmers to add white-space to their code freely with little or no syntactical meaning. Programmers use this freedom to arrange their code so that geometrical features may relate its structure at a glance. That programmers need to use spatial layout as a crutch while composing discrete symbolic sequences is telling; to the interpreter, a block may be a subsequence between braces, but to an experienced programmer it is a perceptual gestalt grouped by indentation. From this we assert that concordant with Dual Coding theory, the linguistic work of programming is supported by spatial reasoning, with secondary notation helping bridge the divide.

There are few examples of spatial arrangement being part of primary syntax. In the large majority of mainstream programming languages geometric syntax does not go beyond one-dimensional adjacency, although in the Python and Haskell languages statements are grouped according to two dimensional rules of indentation. Even visual programming languages, such as the Patcher Languages mentioned in Sect. 9.1, generally do not take spatial arrangement into account (execution order in Max is given by right-left ordering, but the same can be said of 'non-visual' programming languages).

As we noted in Sect. 9.1, the study of "Programming Languages for the Arts" is pushing the boundaries of programming notation, and geometrical syntax is no exception. There are several compelling examples of geometry used in the syntax of languages for music, often commercial projects emerging from academic research. The *ReacTable* (Jordà et al. 2005) is a tangible, multi-user interface, where blocks imprinted with computer readable symbols are placed on a circular display surface (Fig. 9.5). We consider the ReacTable as a programming language environment, although it is not presented as such by its creators. Each symbol represents a sound synthesis function, with a synthesis graph formed based upon the pairwise proximity of the symbols. Relative proximity and orientation of connected symbols are used as parameters modifying the operation of synthesis nodes. Figure 9.6 shows a

Fig. 9.5 The ReacTable (Jordà et al. 2005): a tangible interface for live music, presented here as a programming language environment

screenshot of *Text*, a visual language inspired by the ReacTable and based upon the pure functional Haskell programming language. In Text, functions and values may be placed freely on the page, and those with compatible types are automatically connected together, closest first. Functions are *curried*, allowing terse composition of higher order functions. Text could in theory be used for general programming, but is designed for improvising live music, using an underlying musical pattern library (McLean and Wiggins 2010b). A rather different approach to spatial syntax is taken by *Nodal*, where distance between symbols represents elapsed time during interpretation (McCormack and McIlwain 2011). The result is a control flow graph where time relationships in musical structure can be easily seen and manipulated as spatial relationships.[2] In all of these examples, the graphs may be changed while they are executed, allowing interactive composition and indeed live improvisation of the like examined in Sect. 9.6.

An important assertion within CMT is that a conceptual system of semantic meaning exists within an individual, and not as direct reference to the world. Through language, metaphors become established in a culture and shared by its participants, but this is an effect of individual conceptual systems interacting, and not individuals inferring and adopting external truths of the world (or of possible worlds). This would account for the varied range of programming metaphors discussed in Sect. 9.3, as well as the general failure of attempts at designing fixed metaphors into computer interfaces (Blackwell 2006c). Each programmer has a different set of worldly interests and experiences, and so establishes different

[2]This space/time syntax can also be seen in Al-Jazari mentioned earlier and shown in Fig. 9.2.

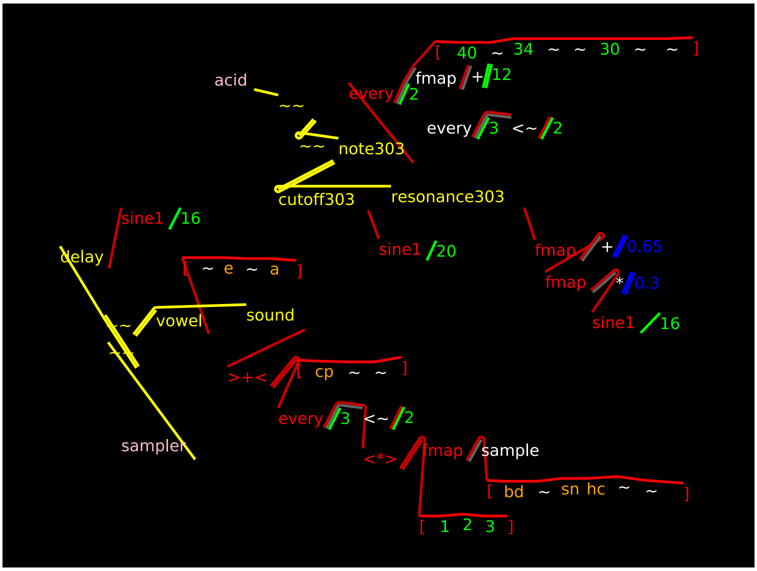


Fig. 9.6 Text, a visual programming language designed for improvised performance of electronic dance music. Functions automatically connect, according to their distance and type compatibility

metaphorical systems to support their programming activities. However, by building orientational and spatial metaphors into programming notation, such as TIME IS DISTANCE, PROXIMITY IS CONNECTIVITY and ORIENTATION IS EXTENT, universal bodily relationships are employed. This results in metaphors that are more readily understood, employing general cognitive resources to artistic expression.

9.5 Components of Creativity

We now have grounds to formally characterise how the creative process operates in bricolage programming. For this we employ the *Creative Systems Framework* (CSF), a high-level formalisation of creativity introduced by Wiggins (2006a,b) and based upon the work of Boden (2003). Creativity is characterised as a *search* in a space of concepts, using the quasi-Platonic idea, common in AI, that there is an effective duality between exploration of an extant range of items, that conform to rules, and construction of new items according to those rules, in a context where the extent of the space is unknown. Within the CSF, a creative search has three key aspects: the conceptual *search space* itself, *traversal* of the space and *evaluation* of concepts found in the space. In other words, creativity requires somewhere to search, a manner of searching, and a means to judge what you find. However, creative behaviour may make use of introspection, self-modification and need boundaries to be broken. That is, the constraints of search space, traversal and evaluation are not fixed, but are examined, challenged and modified by the creative agent following

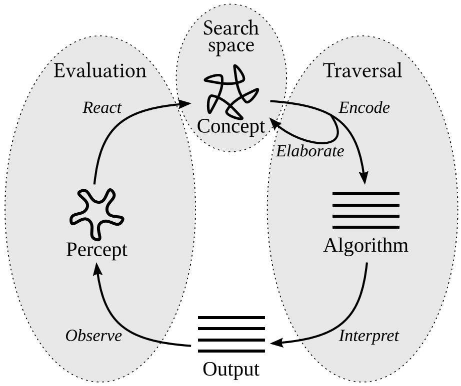


Fig. 9.7 The process of action and reaction in bricolage programming from Fig. 9.3, showing the three components of the Creative Systems Framework, namely search space, traversal strategy and evaluation

(and defined by) them. The CSF supplies tests for particular kinds of *aberration* from the expected conceptual space and suggests approaches to addressing them.

Again using the terminology of Gärdenfors (2000), the search spaces of the CSF are themselves concepts, defining regions in a universal space defined by quality dimensions. Thus, transformational creativity is a geometrical transformation of these regions, motivated by a process of searching through and beyond them; crucially, the search space is not closed. As we will see, this means that a creative agent may creatively push beyond the boundaries of the search. While acknowledging that creative search may operate over linguistic search spaces, we focus on geometric spaces grounded in perception. This follows our focus on artistic bricolage (Sect. 9.2), which revolves around perception. For an approach unifying linguistic and geometric spaces see Forth et al. (2010).

We may now clarify the bricolage programming process introduced in Sect. 9.2.1 within the CSF. As shown in Fig. 9.7, the search space defines the programmer's concept, being their current artistic focus structured by learnt techniques and conventions. The traversal strategy is the process of attempting to generate part of the concept by encoding it as an algorithm, which is then interpreted by the computer. Finally, evaluation is a perceptual process in reaction to the output.

In Sect. 9.2, we alluded to the extended mind hypothesis (Clark 2008), claiming that bricolage programming takes part of the human creative process outside of the mind and into the computer.[3] The above makes clear what we claim is being externalised: part of the traversal strategy. The programmer's concept motivates a development of the traversal strategy, encoded as a computer program, but the programmer does not necessarily have the cognitive ability to fully evaluate it. That task is taken on by the interpreter running on a computer system, meaning that traversal encompasses both encoding by the human and interpretation by the computer.

The traversal strategy is structured by the techniques and conventions employed to convert concepts into operational algorithms. These may include *design patterns*, a standardised set of *ways of building* that have become established around many

[3] See also Chap. 14 by Bown.

classes of programming language. Each design pattern identifies a kind of problem, and describes a generalised structure towards a solution.[4]

The creative process is guided by the programmer's concept of what is a valid end result. This is shaped by the programmer's current artistic focus, being the perceptual qualities they are currently interested in, perhaps congruent with a cultural theme such as a musical genre or artistic movement. Transformational creativity can be triggered in the CSF when traversal extends outside the bounds of the search space. If the discovered conceptual instance is valued, then the search space may be extended to include it. If, however, it is not valued, then the traversal strategy may be modified to avoid similar instances in the future.

Because the traversal strategy of a programmer includes external notation and computation, they are likely to be less successful in writing software that meets their preconceptions, or in other words more successful in being surprised by the results. A creative process that includes external computation will follow less predictable path as a result. Nonetheless the process has the focus of a search space, and is guided by value in relation to a rich perceptual framework, and so while unpredictable, this influence is far from random, being meaningful interplay between human/computer language and human perceptual experience. The human concepts and algorithm are continually transformed in respect to one another, and to perceptual affect, in creative feedback.

According to our embodied view, not only is perception crucial in evaluating output within bricolage programming, but also in structuring the space in which programs are conceptualised. Indeed if the embodied view of CMT holds in general, the same would apply to all creative endeavour. From this we find motivation for the field of computational creativity in grounding an artificial creative agent in its environment. This is done by acquiring computational models of perception sufficient for the agent to both evaluate its own works and structure its conceptual system. Then the agent would have a basis for guiding changes to its own conceptual system and generative traversal strategy, able to modify itself to find artifacts that it was not programmed to find, and place value judgements on them. Such an agent would need to adapt to human culture in order to interact with shifting cultural norms, keeping its conceptual system and resultant creative process coherent within that culture. For now, however, this is wishful thinking, and we must accept generative computer programs which extend human creativity, but are not creative agents in their own right.

9.6 Programming in Time

> "She is not manipulating the machine by turning knobs or pressing buttons. She is writing messages to it by spelling out instructions letter by letter. Her painfully slow typing seems

[4]This structural heuristic approach to problem solving is inspired by work in the field of urban design (Alexander et al. 1977).

laborious to adults, but she carries on with an absorption that makes it clear that time has lost its meaning for her." Sherry Turkle (2005, p. 92), on Robin, aged 4, programming a computer.

Having investigated the representation and operation of bricolage programming we now examine how the creative process operates in time. Dijkstra might argue that considering computer programs as operating in time at all, rather than as entirely abstract logic, is itself a form of the anthropomorphism examined in Sect. 9.3. However from the above quotation it seems that Robin stepped out of any notion of physical time, and into the algorithm she was composing, entering a timeless state. This could be a state of optimum experience, the "flow" investigated by Csikszentmihalyi where "duration of time is altered; hours pass by in minutes, and minutes can stretch out to seem like hours" (Csikszentmihalyi 2008, p. 49). Perhaps in this state a programmer is thinking in algorithmic time, attending to control flow as it replays over and over in their imagination, and not to the world around them. Or perhaps they are not attending to the passage of time at all, thinking entirely of declarative abstract logic, in a timeless state of building. In either case, it would seem that the human is entering time relationships of their software, rather than the opposite, anthropocentric direction of software entering human time. While programmers can appear detached from "physical" time, there are ways in which the timelines of program development and operation may be united, which we will come to shortly.

Temporal relationships are generally not represented in source code. When a programmer needs to do so, for example, as an experimental psychologist requiring accurate time measurements, or a musician needing accurate synchronisation between processes, they run into problems of accuracy and latency. With the wide proliferation of interacting embedded systems, this is becoming a broad concern (Lee 2009). In commodity systems time has been decentralised, abstracted away through layers of caching, where exact temporal dependencies and intervals between events are not deemed worthy of general interest. Programmers talk of "processing cycles" as a valuable resource which their processes should conserve, but they generally no longer have programmatic access to the high frequency oscillations of the central processing units (now, frequently plural) in their computer. The allocation of time to processes is organised top-down by an overseeing scheduler, and programmers must work to achieve what timing guarantees are available. All is not lost, however, as realtime kernels are now available for commodity systems, allowing psychologists (Finney 2001) and musicians (e.g. via http://jackaudio.org/) to get closer to physical time. Further, the representation of time semantics in programming is undergoing active research in a sub-field of computer science known as *reactive programming* (Elliott 2009), with applications emerging in music (McLean and Wiggins 2010a).

9.6.1 Interactive Programming

Interactive programming allows a programmer to examine an algorithm while it is interpreted, taking on live changes without restarts. This unites the time flow of a program with that of its development, using dynamic interpretation or compilation. Interactive programming makes a dynamic creative process of test-while-implement possible, rather than the conventional implement-compile-test cycle, so that arrows shown in Figs. 9.3 and 9.7 show concurrent influences between components rather than time-ordered steps.

Interactive programming not only provides a more efficient creative feedback loop, but also allows a programmer to connect software development with time based art. Since 2003 an active group of practitioners and researchers have been developing new approaches to making computer music and video animation, collectively known as *Live coding* (Blackwell and Collins 2005, Ward et al. 2004, Collins et al. 2003, Rohrhuber et al. 2005). The archetypal live coding performance involves programmers writing code on stage, with their screens projected for an audience, while the code is dynamically interpreted to generate music or video. Here the process of development is the performance, with the work generated not by a finished program, but through its journey of development from nothing to complex algorithm, generating continuously changing musical or visual form along the way. This is bricolage programming taken to a logical and artistic conclusion.

9.7 Conclusion

What we have discussed provides strong motivation for addressing the concerns of artist-programmers. These include concerns of workflow, where elapsed time between source code edits and program output slows the creative process. Concerns of programming environment are also important, which should be optimised for the presentation of shorter programs in their entirety to support bricolage programming, rather than hierarchical views of larger codebases. Perhaps most importantly, we have seen motivation for the development of new programming languages, pushing the boundaries to greater support artistic expression.

From the embodied view we have taken, it would seem useful to integrate time and space further into programming languages. In practice, integrating time can mean, on one hand, including temporal representations in core language semantics, and on the other, uniting development time with execution time, as we have seen with interactive programming. Temporal semantics and interactive programming both already feature strongly in some programming languages for the arts, as we saw in Sect. 9.6, but how about analogous developments in integrating geometric relationships into the semantics and activity of programming? It would seem the approaches shown in Nodal, the ReacTable and Text described in Sect. 9.1 are showing the way towards greater integration of computational geometry and perceptual models into programming language. This is already serving artists well, and could become a new focus for visual programming language research.

We began with Paul Klee, a painter whose production was limited by his two hands. The artist-programmer is limited differently to the painter, but shares what Klee called his limitation of reception, by the “limitations of the perceiving eye”.

This is perhaps a limitation to be expanded but not overcome: celebrated and fully explored using all we have, including our new computer languages. We have characterised a bricolage approach to artistic programming as an embodied, creative feedback loop. This places the programmer close to their work, grounding discrete computation in orientational and temporal metaphors of their human experience. However, the computer interpreter extends the programmer's abilities beyond their own imagination, making unexpected results likely, leading the programmer to new creative possibilities.

Acknowledgements Alex McLean was supported by a Doctoral grant awarded by the UK EPSRC.

References

Alexander, C., Ishikawa, S., & Silverstein, M. (1977). *A pattern language: towns, buildings, construction* (1st ed.) London: Oxford University Press.
Blackwell, A. (2006a). Gender in domestic programming: from bricolage to séances d'essayage. In *CHI workshop on end user software engineering*.
Blackwell, A., & Collins, N. (2005). The programming language as a musical instrument. In *Proceedings of PPIG05*. University of Sussex.
Blackwell, A. F. (2006b). Metaphors we program by: space, action and society in java. In *Proceedings of the psychology of programming interest group 2006*.
Blackwell, A. F. (2006c). The reification of metaphor as a design tool. *ACM Transactions on Computer-Human Interaction*, *13*(4), 490–530.
Blackwell, A., & Green, T. (2002). *Notational systems—the cognitive dimensions of notations framework* (pp. 103–134). San Mateo: Morgan Kaufmann.
Boden, M. A. (2003). *The creative mind: myths and mechanisms* (2nd ed.). London: Routledge.
Brown, P., Gere, C., Lambert, N., & Mason, C. (Eds.) (2009). *White heat cold logic: British computer art 1960–1980. Leonardo books*. Cambridge: MIT Press.
Church, A. (1941). *The calculi of lambda conversion*. Princeton: Princeton University Press.
Clark, A. (2008). *Supersizing the mind: embodiment, action, and cognitive extension. Philosophy of mind series*. OUP USA.
Collins, N., McLean, A., Rohrhuber, J., & Ward, A. (2003). Live coding in laptop performance. *Organised Sound*, *8*(03), 321–330.
Csikszentmihalyi, M. (2008). *Flow: the psychology of optimal experience*. HarperCollins eBooks.
Dijkstra, E. W. (1988). *On the cruelty of really teaching computing science (EWD-1036)*. E.W. Dijkstra Archive. Center for American History, University of Texas at Austin.
Elliott, C. (2009). Push-pull functional reactive programming. In *Haskell symposium*.
Essinger, J. (2004). *Jacquard's web: how a Hand-Loom led to the birth of the information age* (1st ed.). London: Oxford University Press.
Finney, S. A. (2001). Real-time data collection in Linux: a case study. *Behavior Research Methods, Instruments, & Computers*, *33*(2), 167–173.
Forth, J., McLean, A., & Wiggins, G. (2008). Musical creativity on the conceptual level. In *IJWCC 2008*.
Forth, J., Wiggins, G., & McLean, A. (2010). Unifying conceptual spaces: concept formation in musical creative systems. *Minds and Machines*, *20*(4), 503–532.
Gärdenfors, P. (2000). *Conceptual spaces: the geometry of thought*. Cambridge: MIT Press.
Jordà, S., Kaltenbrunner, M., Geiger, G., & Bencina, R. (2005). The reacTable. In *Proceedings of the international computer music conference (ICMC 2005)* (pp. 579–582).
Klee, P. (1953). *Pedagogical sketchbook*. London: Faber and Faber.

Lakoff, G., & Johnson, M. (1980). *Metaphors we live by* (1st ed.). Chicago: University of Chicago Press.

Lee, E. A. (2009). Computing needs time. *Communications of the ACM*, *52*(5), 70–79.

Lévi-Strauss, C. (1968). *The savage mind*. *Nature of human society*. Chicago: University of Chicago Press.

Levy, S. (2002). *Hackers: heroes of the computer revolution*. Baltimore: Penguin Putnam.

McCartney, J. (2002). Rethinking the computer music language: SuperCollider. *Computer Music Journal*, *26*(4), 61–68.

McCormack, J., & McIlwain, P. (2011). Generative composition with nodal. In E. R. Miranda (Ed.), *A-Life for music: music and computer models of living systems, computer music and digital audio* (pp. 99–113). A-R Editions.

McLean, A., Griffiths, D., Collins, N., & Wiggins, G. (2010). Visualisation of live code. In *Electronic visualisation and the arts*, London, 2010.

McLean, A., & Wiggins, G. (2010a). Petrol: reactive pattern language for improvised music. In *Proceedings of the international computer music conference*.

McLean, A., & Wiggins, G. (2010b). Tidal—pattern language for the live coding of music. In *Proceedings of the 7th sound and music computing conference*.

Murphy, G. L. (2002). *The big book of concepts*. *Bradford books*. Cambridge: MIT Press.

Paivio, A. (1990). *Mental representations: a dual coding approach*. *Oxford psychology series* (new ed.). London: Oxford University Press.

Petre, M., & Blackwell, A. F. (1999). Mental imagery in program design and visual programming. *International Journal of Human-Computer Studies*, *51*, 7–30.

Polgár, T. (2005). *Freax*. CSW-Verlag.

Puckette, M. (1988). The patcher. In *Proceedings of international computer music conference*.

Reas, C., & Fry, B. (2007). *Processing: a programming handbook for visual designers and artists*. Cambridge: MIT Press.

Rohrhuber, J., de Campo, A., & Wieser, R. (2005). Algorithms today: notes on language design for just in time programming. In *Proceedings of the 2005 international computer music conference*.

Schon, D. A. (1984). The reflective practitioner: how professionals think. In *Action* (1st ed.). New York: Basic Books.

Shepard, R. N., & Metzler, J. (1971). Mental rotation of three-dimensional objects. *Science (New York, N.Y.)*, *171*(972), 701–703.

Turing, A. M. (1992). Intelligent machinery. Report, national physics laboratory. In D. C. Ince (Ed.), *Collected works of A. M. Turing: mechanical intelligence* (pp. 107–127). Amsterdam: Elsevier.

Turkle, S. (2005). *The second self: computers and the human spirit* (20 anv. ed.). Cambridge: MIT Press.

Turkle, S., & Papert, S. (1990). Epistemological pluralism: styles and voices within the computer culture. *Signs*, *16*(1), 128–157.

Turkle, S., & Papert, S. (1992). Epistemological pluralism and the revaluation of the concrete. *Journal of Mathematical Behavior*, *11*(1), 3–33.

Vogel, J. (2003). Cerebral lateralization of spatial abilities: a meta-analysis. *Brain and Cognition*, *52*(2), 197–204.

Wang, G., & Cook, P. R. (2004). On-the-fly programming: using code as an expressive musical instrument. In *Proceedings of the 2004 conference on new interfaces for musical expression* (pp. 138–143). National University of Singapore.

Ward, A., Rohrhuber, J., Olofsson, F., McLean, A., Griffiths, D., Collins, N., & Alexander, A. (2004). Live algorithm programming and a temporary organisation for its promotion. In O. Goriunova & A. Shulgin (Eds.), *read_me—software art and cultures*.

Wiggins, G. A. (2006a). A preliminary framework for description, analysis and comparison of creative systems. *Journal of Knowledge Based Systems*, *19*, 449–458.

Wiggins, G. A. (2006b). Searching for computational creativity. *New Generation Computing*, *24*(3), 209–222.

Part III
Theory

Chapter 10
Computational Aesthetic Evaluation: Past and Future

Philip Galanter

Abstract Human creativity typically includes a self-critical aspect that guides innovation towards a productive end. This chapter offers a brief history of, and outlook for, computational aesthetic evaluation by digital systems as a contribution towards potential machine creativity. First, computational aesthetic evaluation is defined and the difficult nature of the problem is outlined. Next, a brief history of computational aesthetic evaluation is offered, including the use of formulaic and geometric theories; design principles; evolutionary systems including extensions such as coevolution, niche construction, agent swarm behaviour and curiosity; artificial neural networks and connectionist models; and complexity models. Following this historical review, a number of possible contributions towards future computational aesthetic evaluation methods are noted. Included are insights from evolutionary psychology; models of human aesthetics from psychologists such as Arnheim, Berlyne, and Martindale; a quick look at empirical studies of human aesthetics; the nascent field of neuroaesthetics; new connectionist computing models such as hierarchical temporal memory; and computer architectures for evolvable hardware. Finally, it is suggested that the effective complexity paradigm is more useful than information or algorithmic complexity when thinking about aesthetics.

10.1 Introduction

This chapter looks at computers and aesthetic evaluation. In common usage the word *creativity* is associated with bringing the new and innovative into being. The term, whether used in reference to the arts or more generally, connotes a sort of self-directedness and internal drive. Evaluation or criticism is by its very nature reactive. Something is first created and only then can it be evaluated. Evaluation and creativity at first seem to be two different kinds of activity performed at different times.

But almost any exploration of creativity will quickly reveal evaluation threaded throughout the entire process. For accomplished artists there are usually at least

P. Galanter (✉)
Department of Visualization, Texas A&M University, College Station, Texas, USA
e-mail: galanter@viz.tamu.edu

J. McCormack, M. d'Inverno (eds.), *Computers and Creativity*,
DOI 10.1007/978-3-642-31727-9_10,

three ways evaluation becomes an intrinsic part of the creative process. First, artists typically exercise evaluation as they experience, study, and find inspiration in the work of other artists. In practice artists will execute countless micro-evaluations as part of making aesthetic decisions for works-in-progress. Once completed, artists evaluate the final product, gaining new insights for the making of the next piece.

If computers are to become artistically creative their need for an evaluative function will be no less acute. Computer artists have invented a great variety of fecund computational methods for generating aesthetic possibilities and variations. But computational methods for making aesthetically sound choices among them have lagged far behind.

This chapter provides specific examples of computational methods for making aesthetic choices. Longer examples have been selected as good illustrations of a particular approach, with shorter examples providing variations. Some examples show where a path is already known to lead, while others are provided as trail heads worthy of further exploration.

10.1.1 What Do We Mean by Computational Aesthetic Evaluation?

The word *evaluation* is sometimes prone to ambiguous use due to the multiple meanings of the word *value*. For example, a mathematician can be said to evaluate an expression or formula. An art expert might evaluate a given object for market value or authenticity. Part of that might involve an evaluation of style and provenance.

For this discussion *aesthetic evaluation* refers to making normative judgements related to questions of beauty and taste in the arts. It's worth noting that the word "aesthetics" alone can imply a broader critical contemplation regarding art, nature, and culture. The topic of aesthetics, including evaluation, goes back at least to Plato and Aristotle in the West (for a good overview of philosophical aesthetics see Carroll 1999).

The term *computational aesthetics* has been somewhat instable over time. For some the term includes both generative and analytic modes, i.e. both the creation and evaluation of art using a computer. For others it purely refers to the use of computers in making aesthetic judgements. This chapter concentrates on systems for making normative judgements, and to emphasise this I've used the terms "computational aesthetic evaluation", "machine evaluation", and "computational evaluation" as synonyms (Hoenig 2005, Greenfield 2005b).

Computational aesthetic evaluation includes two related but distinct application modes. In one mode aesthetic evaluations are expected to simulate, predict, or cater to human notions of beauty and taste. In the other mode machine evaluation is an aspect of a meta-aesthetic exploration and usually involves aesthetic standards created by software agents in artificial worlds. Such aesthetics typically feel alien and disconnected from human experience, but can provide insight into all possible aesthetics including our own.

Finally, it's worth noting that *aesthetic evaluation* and the *evaluation of creativity* are somewhat related but quite distinct. For example, accomplishments in non-artistic fields such as science and mathematics can also be evaluated as to their degree of creativity. And in the arts it's possible to have an artwork of high aesthetic value but without much creativity, or a highly creative artwork where the aesthetics are poor or even irrelevant.

10.1.2 Why Is Computational Aesthetic Evaluation so Difficult if not Impossible?

It should be noted at the outset that computational aesthetic evaluation is an extremely difficult problem. In the abstract, notions of computational aesthetic evaluation and computational creativity lead to deep philosophical waters regarding phenomenology and consciousness. Let's assume a computational evaluation system is created that appears to duplicate human aesthetic judgement. Would such a machine actually experience a sense of redness, brightness or other qualia? How would we know? Can machine evaluation be successful without such experience? If such a machine isn't conscious does that mean human aesthetic judgement and computational aesthetic evaluation are quite different? Or could it be that they aren't so different after all because the brain is itself a machine? All of these interesting questions are outside of the scope of this chapter.

Some feel that effective practical computational evaluation will remain out of reach in our lifetime and perhaps forever. The complications begin with the likely fact that the human aesthetic response is formed by a combination of genetic predisposition, cultural assimilation, and unique individual experience. Despite a growing research literature, the psychology of aesthetics is a mostly incomplete science, and our understanding of each component is limited.

Even if we had a full understanding of aesthetics' genetic, cultural, developmental, and psychological modalities, the creation of comparable computational functionality would remain a daunting task. It would probably require the resolution of a number of standing hard problems in artificial intelligence. A model of human aesthetics, or human intelligence in general, has to represent more than a hypothetical brain-in-a-jar. Our aesthetic sense and psychological makeup are in part the result of embodied experience situated in a specific environment. Machine evaluation will have to account for perception not as a passive mental process, but rather as a dynamic interaction between our bodies and the world (Davis and Rebelo 2007, McCormack 2008). Additionally, it will have to allow for emotions and the irrational Dionysian element in the arts.

10.2 A Brief History of Computational Aesthetic Evaluation

Any suggested computational aesthetic evaluation mechanism is going to contain, at least implicitly, a theory of aesthetics. Most theories from the history of aesthetics do not immediately suggest algorithms, quantifiable properties, or objective formulas. But some do and it is with those that our discussion begins.

10.2.1 Formulaic and Geometric Theories

The mathematician George David Birkhoff published a mostly speculative book in 1933 titled "Aesthetic Measure". Birkhoff limits his theory to aspects of pure form (the "formal") and doesn't address symbolic meaning (the "connotative"). He then proposes the formula $M = O/C$ where M is the measure of aesthetic effectiveness, O is the degree of order, and C is the degree of complexity. Birkhoff (1933) notes, "The well known aesthetic demand for 'unity in variety' is evidently closely connected with this formula."

Birkhoff warns that his measure can only be applied within a group of similar objects and not across types such as a mix of oil and watercolour paintings. He also finesses variation in experience and taste intending M to be a measure for an "idealised 'normal observer' " as a sort of mean of the population.

While most of the book is presented from a mathematical point of view, it is sometimes forgotten that Birkhoff begins with an explicit psychoneurological hypothesis. He describes complexity (C) as the degree to which unconscious psychological and physiological effort must be made in perceiving the object. Order (O) is the degree of unconscious tension released as the perception is realised. This release mostly comes from the consonance of perceived features such as "repetition, similarity, contrast, equality, symmetry, balance, and sequence." While Birkhoff views complexity and order as ultimately psychological phenomena, for analysis he operationalises those concepts using mathematical representations. He then goes on to analyse examples such as polygons, vases, and harmonic structures in music to illustrate his theory.

Birkhoff's theory has been disputed from its first publication. For example, in 1939 Wilson published experimental results showing that Birkhoff's measure did not correlate with actual subjects' stated aesthetic preferences regarding polygons (Wilson 1939). Alternate formulas have been offered that seem to correlate more closely with the judgements of subjects (Boselie and Leeuwenberg 1985, Staudek 1999). And for some, Birkhoff's formula seems to measure orderliness rather than beauty, and penalises complexity in a rather unqualified way (Scha and Bod 1993).

But there are at least two aspects of Birkhoff's work that remain in legitimate play today. First is the intuition that aesthetic value has something to do with complexity and order relationships. Second is the idea that modelling brain function can illuminate discussions of aesthetics. Indeed, both of these reappear as themes throughout this chapter.

The positing of mathematical bases for aesthetics long predate Birkhoff. Pythagoras is traditionally credited with the discovery that dividing a vibrating string following simple consecutive integer ratios such as 1:2, 2:3, and 3:4 yields pleasing harmony relationships. The Golden Ratio ϕ, an irrational constant approximately equal to 1.618, and the related Fibonacci series have been said to generate proportions of optimal aesthetic value. It is claimed they are embedded in great works of art, architecture, and music.

Psychologist Gustav Fechner is credited with conducting the first empirical studies of human aesthetic response in the 1860s. His experiments seemed to show that golden rectangles had the greatest appeal relative to other aspect ratios. But subsequent studies have cast strong doubt on those results. As noted in a special issue of the journal *Empirical Studies of the Arts*, there were methodological flaws and cultural bias in previous confirmatory studies (McCormack 2008, Holger 1997).

In addition, Livio has credibly debunked supposed Golden Ratio use in works including the Great Pyramids, the Parthenon, the Mona Lisa, compositions by Mozart, and Mondrian's late paintings. However, he notes that use of the Golden Ratio as an aesthetic guide has become something of a self-fulfilling myth. For example, Le Corbusier's Modulator, a design aid for proportions, was consciously based on the Golden Ratio (Livio 2003).

On a bit firmer ground is a principle credited to linguist George Kingsley Zipf commonly referred to as *Zipf's law*. As first applied to natural language, one can begin with a large body of text and tally every word counting each occurrence. Then list each word from the most to the least frequent. The observed result is that for the frequency P_i of a given word with a given rank i:

$$P_i \approx \frac{1}{i^a} \tag{10.1}$$

where the exponent a is near 1 (Zipf 1949).

Manaris et al. (2005; 2003) note that this power law relationship has not only been verified in various bodies of musical composition, but also "colours in images, city sizes, incomes, music, earthquake magnitudes, thickness of sediment depositions, extinctions of species, traffic jams, and visits of websites, among others." They go on to show how Zipf metrics can be used to classify specific works as to composer, style, and an aesthetic sense of "pleasantness". In addition Machado et al. (2007) apply Zipf's law in the creation of artificial art critics. Much earlier work showed that both frequency and loudness in music and speech conform to a $1/f$ statistical power law. The authors suggest using $1/f$ distributions in generative music (Voss and Clarke 1975).

Studies by Taylor have shown that late period "drip" paintings by Jackson Pollock are fractal-like. He has also suggested that the fractal dimension of a given Pollock painting is correlated with its aesthetic quality. Fractals are mathematical objects that exhibit self-similarity at all scales. Examples of real world objects that are fractal-like in form include clouds, mountains, trees, and rivers. In the case of Pollock's paintings the fractal dimension is a measure of the degree to which the canvas is filled with finely detailed complex structures. A paint mark with a fractal dimension of 1 will no more fill the canvas with detailed structures than a typical straight line. A paint mark with a fractal dimension of 2 will entirely fill the canvas with fine detail. These correspond well with our everyday topological sense of one and two dimensional spaces (Peitgen et al. 1992).

Pollock's paint marks exhibit detail between these two extremes, and have a non-integer dimension somewhere between 1 and 2. When measured empirically the fractal dimension of his paintings increases over time from 1.12 in 1945 to 1.72 in 1952. Presumably Pollock's innovative "dripping" technique improved over time and in this very limited realm the fractal dimension can be used for aesthetic evaluation (Taylor 2006). Use of a related measure applied to non-fractal two-dimensional patterns correlates well with beauty and complexity as reported by human subjects (Mori et al. 1996).

Work has been done in the fields of medical reconstructive and cosmetic surgery to quantify facial and bodily beauty as an objective basis for evaluating the results of medical procedures. Hönn and Göz (2007) in the field of orofacial orthopaedics cite studies indicating that infants preferentially select for facial attractiveness, and that such judgements by adults are consistent across cultures. Atiyeh and Hayek (2008) provide a survey for general plastic surgery, indicating a likely genetic basis for the perception of both facial and bodily attractiveness. Touching on rules of proportion used by artists through the centuries they seem ambivalent or even supportive of the Golden Ratio standard. However, in conclusion they write, "The golden section phenomenon may be unreliable and probably is artifactual".

To date when it comes to quantifying human facial and bodily beauty there is no medical consensus or standardised measure. More broadly, many now feel that any simple formulaic approach to aesthetic evaluation will be inadequate. Beauty seems to be too multidimensional and too complex to pin down that easily.

10.2.2 Design Principles

Another source of aesthetic insight is the set of basic principles taught in typical design foundations courses. A standard text in American classrooms includes considerations such as: value and distribution; contrast; colour theory and harmony; colour interaction; weight and balance; distribution and proportion; and symmetrical balance. Also included are Gestalt-derived concepts like grouping, containment, repetition, proximity, continuity, and closure (Stewart 2008).

However, to date there is very little in the way of software that can extract these features and then apply rule-of-thumb evaluations. Among the few is a system that makes aesthetic judgements about arbitrary photographs. Datta et al. (2006; 2007) began with a set of photos from a photography oriented social networking site. Each photo was rated by the membership. Image processing extracted 56 simple measures related to exposure, colour distribution and saturation, adherence to the "rule of thirds," size and aspect ratio, depth of field, and so on. The ratings and extracted features were then processed using both regression analysis and classifier software. This resulted in a computational model using 15 key features. A software system was then able to classify photo quality in a way that correlated well with the human ratings.

Some work has been done using colour theory as a basis for machine evaluation. Tsai et al. (2007) created a colour design system using genetic searching and noted, "... auto-searching schemes for optimal colour combinations must be supervised by appropriate colour harmony theories since if such supervision is not applied, the search results are liable to be dull and uncoordinated…" Others have applied a variation of Birkhoff's aesthetic measure for colour harmony attempting to better define order in colour schemes (Li and Zhang 2004).

But overall there has been little progress in automating design principles for aesthetic evaluation. Feature extraction figures heavily in this problem, so perhaps future computer vision researchers will take on this problem.

10.2.3 Artificial Neural Networks and Connectionist Models

Artificial neural networks are software systems with designs inspired by the way neurones in the brain are thought to work. In the brain neurone structures called *axons* act as outputs and *dendrites* act as inputs. An axon to dendrite junction is called a *synapse*. In the brain, electrical impulses travel from neurone to neurone where the synaptic connections are strong. Synapse connections are strengthened when activation patterns reoccur over time. Learning occurs when experience leads to the coherent formation of synapse connections.

In artificial neural networks virtual neurones are called *nodes*. Nodes have multiple inputs and outputs that connect to other nearby nodes similar to the way synapses connect *axons* and *dendrites* in the brain. Like synapses these connections are of variable strength, and this is often represented by a floating point number. Nodes are typically organised in layers, with an input layer, one or more hidden layers, and finally an output layer. Connection strengths are not manually assigned, but rather "learned" by the artificial neural network as the result of its exposure to input data.

For example, a scanner that can identify printed numbers might be created by first feeding pixel images to the input layer of an artificial neural network. The data then flows through the hidden layer connections according to the strength of each connection. Finally, one of ten output nodes is activated corresponding to one of the digits from "0" to "9". Before being put into production the scanner would be trained using known images of digits.

Some of the earliest applications of neural network technology in the arts consisted of freestanding systems used to compose music (Todd 1989). Later in this chapter artificial neural networks will be described as providing a component in evolutionary visual art systems (Baluja et al. 1994).

A significant challenge in using artificial neural networks is the selection, conditioning, and normalisation of data presented to the first layer of nodes. It was noted in Sect. 10.2.1 that ranked music information following Zipf's law can be used to identify composers and evaluate aesthetics. Manaris et al. (2005; 2003) reported an impressive success rate of 98.41 % in attempting to compute aesthetic ratings within one standard deviation of the mean from human judges.

A similar effort was made to evaluate a mix of famous paintings and images from a system of evolved expressions. The machine evaluation used Zipfian rank-frequency measures as well as compression measures as proxies for image complexity. The authors reported a success rate of 89 % when discriminating between human and system-produced images. Using famous paintings in the training set provided stability and human-like standards of evaluation. Using system produced images allowed the evolution of more discerning classifiers (Machado et al. 2008). In a related paper the authors demonstrate artificial neural networks that can discriminate works between: Chopin and Debussy; Scarlatti and Purcell; Purcell, Chopin, and Debussy; and other more complicated combinations. In another demonstration, a neural network was able to discriminate works between Gauguin, Van Gogh, Monet, Picasso, Kandinsky, and Goya (Machado et al. 2004, Romero et al. 2003).

Without explicit programming, artificial neural networks can learn and apply domain knowledge that may be fuzzy, ill defined, or simply not understood. Phon-Amnuaisuk (2007) has used a type of artificial neural network called *self-organising maps* to extract musical structure from existing human music, and then shape music created by an evolutionary system by acting as a critic. Self-organising map-based music systems sometimes produce reasonable sequences of notes within a measure or two, but lack the kind of global structure we expect music to have. In an attempt to address this problem self-organising maps have been organised in hierarchies so that higher-level maps can learn higher levels of abstraction (Law and Phon-Amnuaisuk 2008). In another experiment, artificial neural networks were able to learn viewer preferences among Mondrian-like images and accurately predict preferences when viewing new images (Gedeon 2008).

10.2.4 Evolutionary Systems

The evolutionary approach to exploring solution spaces for optimal results has had great success in a diverse set of industries and disciplines (Fogel 1999). Across a broad range of approaches some kind of evaluation is typically needed to steer evolution towards a goal. Much of our discussion about computational aesthetic evaluation will be in the context of evolutionary systems. But first consider the following simplified industrial application.

Assume the problem at hand is the design of an electronic circuit. First, chromosome-inspired data structures are created and initially filled with random values. Each chromosome is a collection of simulated genes. Here each gene describes an electronic component or a connection, and each chromosome represents a circuit that is a potential solution to the design problem. The genetic information is referred to as the *genotype*, and the objects and behaviours they ultimately produce are collectively called the *phenotype*. The process of genotype-creating-phenotype is called *gene expression*. A chromosome can reproduce with one or more of its genes randomly mutated. This creates a variation of the parent circuit. Or two chromosomes can recombine creating a new circuit that includes aspects of both parents.

In practice, a subset of chromosomes is selected for variation and reproduction, and the system evaluates the children as possible solutions. In the case of circuit design a chromosome will be expressed as a virtual circuit and then tested with a software-based simulator. Each circuit design chromosome is assigned a score based on not only how well its input and output match the target specification, but perhaps other factors such as the cost and number of parts, energy efficiency, and ease of construction.

The formula that weights and combines these factors into a single score is called a *fitness function*. Chromosomes with higher fitness scores are allowed to further reproduce. Chromosomes with lower fitness scores are not selected for reproduction and are removed from evolutionary competition. Using a computer this cycle of selection, reproduction, variation, and fitness evaluation can be repeated hundreds of times with large populations of potential circuits. Most initial circuits will be quite dysfunctional, but fortuitous random variations will be retained in the population, and eventually a highly optimised "fit" circuit will evolve. For an excellent introduction to evolutionary systems in computer art see Bentley and Corne (2002). In that same volume, Koza et al. (2002) illustrate the application of genetic programming in real world evolutionary circuit design.

Evolutionary systems have been used to create art for more than 20 years (Todd and Latham 1992). But an evolutionary approach to art is particularly challenging because it is not at all clear how aesthetic judgement can be automated for use as a fitness function. Nevertheless, evolution remains a popular generative art technique despite this fundamental problem (for an overview of current issues in evolutionary art see McCormack 2005 and Galanter 2010).

From the outset there have been two popular responses to the fitness function problem. The first has been to put the artist in the loop and assign fitness scores manually. The second has been to use computational aesthetic evaluation and generate fitness scores computationally. More recently there have been efforts to create systems with fitness functions that are emergent rather than externally determined.

10.2.5 Interactive Evolutionary Computation

From the earliest efforts interactive (i.e. manual) assignment of fitness scores has dominated evolutionary art practice (Todd and Latham 1992, Sims 1991). There was also early recognition that the human operator creates a "fitness bottleneck" (Todd and Werner 1998). This labour-intensive bottleneck forces the use of fewer generations and smaller populations than in other applications (for a comprehensive overview of interactive evolutionary computing across a number of industries, including media production, see Takagi 2001).

There are additional problems associated with the interactive approach. For example, human judges become fatigued, less consistent, and prone to skew towards short term novelty at the expense of aesthetic quality (Takagi 2001, Yuan 2008). One suggested remedy for such fatigue problems has been to *crowd-source* evaluation. This involves recruiting large numbers of people for short periods of time to render judgements. In Sim's *Galapagos*, choices viewers make as to which of a number of monitors to watch are used as implicit fitness measures (Sims 1997). The *Electric Sheep* project provides evolutionary fractal flame art as a screen saver on thousands of systems around the world. Users are invited to provide online feedback regarding their preferences (Draves 2005).

But the crowd-sourcing solution is not without its own potential problems. Artists Komar and Melamid executed a project called *The People's Choice* that began by polling the public about their preferences in paintings. Based on the results regarding subject matter, colour, and so on they created a painting titled *America's Most Wanted*. The result is a bland landscape that would be entirely unmemorable if it were not for the underlying method and perhaps the figure of George Washington and a hippopotamus appearing as dada-like out-of-context features. As should be expected the mean of public opinion doesn't seem to generate the unique vision most expect of contemporary artists. Komar and Melamid's critique in this project was directed at the politics of public relations and institutions that wield statistics as a weapon. But the aesthetic results advise caution to those who would harness crowd-sourced aesthetic evaluation in their art practice (Komar et al. 1997, Ross 1995). It's also worth noting that Melamid observed that some aesthetic preferences are culturally based but others seemed to be universal. The evolutionary implications of this will be discussed later in the section on Denis Dutton and his notion of the "art instinct", Sect. 10.3.1.

Another approach has been to manually score a subset, and then leverage that information across the entire population. Typically this involves clustering the population into similarity groups, and then only manually scoring a few representatives from each (Yuan 2008, Machado et al. 2005). Machwe (2007) has suggested that artificial neural networks can generalise with significantly fewer scored works than the interactive approach requires.

10.2.6 Automated Fitness Functions Based on Performance Goals

The Mechanical Turk was a purported mechanical chess-playing machine created in the late 18th century by Wolfgang von Kempelen. But it was really more a feat of stage magic than computation. Exhibitors would make a great show of opening various doors revealing clockwork-like mechanisms. Despite appearances, a human operator was hidden inside the cabinet, so the chess game was won or lost based on the decisions the operator made (Aldiss 2002, Standage 2002).

To some extent using interactive evolutionary computing for art is a similar trick. These systems can generate and display a variety of options at every step, but ultimately the aesthetic challenge is won or lost based on the decisions made by the artist-operator.

Fully automated evolutionary art systems call for, rather than offer, a solution to the challenge of computational aesthetic evaluation. Machine evaluation can be relatively simple when the aesthetic is Louis H. Sullivan's principle that "form follows function" (Sullivan 1896). Computational evaluation here is tractable to the extent the needed functionality can be objectively evaluated via computation. For example, Gregory Hornby and Jordan Pollack created an evolutionary system for designing furniture (tables). Their fitness function sought to maximise height, surface structure, and stability while minimising the amount of materials required. This approach is similar to the optimisation-oriented evolutionary systems found in industry (Hornby and Pollack 2001).

Similarly, specific performance goals can provide a fitness function in a straightforward way in art applications. Sims' *Evolved Virtual Creatures* is an early example. His evolutionary system bred virtual organisms with simple "neuron" circuitry and actuators situated in a world of simulated physics. The initial creatures, seeded with random genes, would typically just twitch in an uncoordinated way. But then selection pressure was applied to the evolving population using a simple fitness function that might reward jumping height, walking speed, or swimming mobility. As a result, the evolved creatures exhibited very competent locomotion behaviour. Some seemed to rediscover movement found in the natural world, while others exhibited strange and completely novel solutions (Sims 1994).

Performance goals can also be useful in the development of characters for computer games through evolution. For example, the amount of time a character survives can be used as a fitness function yielding incrementally stronger play (Wu and Chien 2005).

Diffusion limited aggregation (DLA) systems can be used to create growing frost- or fern-like patterns, and have been studied using evolutionary performance goals. They grow as particles in random Brownian motion adhere to an initial seed particle. To study optimal seed placement, Greenfield (2008a) applied an evolutionary system where the size of the resulting pattern served as an effective fitness measure. In another project he used an evolutionary system to explore the effect of transcription factors on morphology. Each transcription factor was assigned a different colour. The performance and aesthetics of the result were improved by using a fitness function that rewarded transcription factor diversity (Greenfield 2004). Similarly, an evolutionary sculpture system using cubic blocks as modules has produced useful emergent forms simply by rewarding height or length (Tufte and Gangvik 2008).

In their project "Breed" Driessens and Verstappen created a subtractive sculpture system. Each sculpture is started as a single cube treated as a cell. This cell is subdivided into eight smaller sub-cells, one for each corner. Rules driven by the state of neighbouring cells determine whether a sub-cell is kept or carved away. Then each of the remaining cells has the subdivision rules applied to them. And so on. The final form is then evaluated for conformance to goals for properties such as volume, surface area and connectivity. In "Breed" the rule-set is the genotype, the final sculpture is the phenotype, and evaluation relative to performance goals is used as a fitness function. Unlike most other evolutionary systems there is a population size of just one. A single mutation is produced and given an opportunity to unseat the previous result. At some point the gene, i.e. rule set, ceases to improve by mutation and the corresponding sculpture is kept as the result.

Whitelaw (2003) points out that unlike industrial applications where getting stuck on a local maximum is seen as an impediment to global optimisation, this project uses local maxima to generate a family of forms (differing solutions) related by their shared fitness function. Also Whitelaw points out that unlike some generative systems that reflect human selection and intent, Driessens and Verstappen have no particular result in mind other than allowing the system to play itself out to a final self-directed result. In this case performance goals play quite a different role than those used in optimisation-oriented industrial systems.

10.2.7 Evolutionary Fitness Measured as Error Relative to Exemplars

Representationalism in visual art began diminishing in status with the advent of photographic technologies. Other than use as an ironic or conceptual gesture, mimesis is no longer a highly valued pursuit in contemporary visual art. Similarly a difference or error measure comparing a phenotype to a real-world example is not typically useful as an aesthetic fitness function. In the best case such a system would merely produce copies. What have proven interesting, however, are the less mimetic intermediate generations where error measures can be reinterpreted as the degree of abstraction in the image.

For example, Aguilar and Lipson (2008) constructed a physical painting machine driven by an evolutionary system. A scanned photograph serves as the target and each chromosome in the population is a set of paint stroke instructions. A model of pigment reflectance is used to create digital simulations of the prospective painting in software. A software comparison of pixel values from the simulated painting and the original image generates a fitness score. When a sufficient fitness score is achieved the chromosome is used to drive a physical painting machine that renders the brush strokes on canvas with acrylic paint.

Error measurement makes particularly good sense when programming music synthesisers to mimic other sound sources. Comparisons with recordings of traditional acoustic instruments can be used as a fitness function. And before the evolutionary system converges on an optimal mimesis interesting timbres may be discovered along the way (McDermott et al. 2005, Mitchell and Pipe 2005).

Musique concrete is music constructed by manipulating sound samples. For evolutionary musique concrete short audio files can be subjected to operations similar to mutation and crossover. They are then combined and scored relative to a second target recording. Again mimesis is not the intent. What the audience hears is the evolving sound as it approaches but does not reach the target recording (Magnus 2006, Fornari 2007). Gartland-Jones (2002) has used a similar target tracking approach with the addition of music theory constraints for evolutionary music composition.

In a different music application Hazan et al. (2006) have used evolutionary methods to develop regression trees for expressive musical performance. Focusing on note duration only, and using recordings of jazz standards as a training set, the resulting regression trees can be used to transform arbitrary flat performances into expressive ones.

There are numerous other examples of error measures used as fitness functions. For example, animated tile mosaics have been created that approach a reference portrait over time (Ciesielski 2007). The fitness of shape recognition modules have been based on their ability to reproduce shapes in hand drawn samples (Jaskowski 2007). An automated music improviser has been demonstrated that proceeds by error minimisation of both frequency and timbre information (Yee-King 2007). Alsing (2008) helped to popularise the error minimisation approach to mimetic rendering with a project that evolved a version of the "Mona Lisa" using overlapping semi-transparent polygons.

10.2.8 Automated Fitness Functions Based on Complexity Measures

Fitness scores based on aesthetic quality rather than simple performance or mimetic goals are much harder to come by. Machado and Cardoso's *NEvAr* system uses computational aesthetic evaluation methods that attempt to meet this challenge. They generate images using an approach first introduced by (Sims 1991) called *evolving expressions*. It uses three mathematical expressions to calculate pixel values for the red, blue, and green image channels. The set of math expressions operates as a genotype that can reproduce with mutation and crossover operations.

Machado and Cardoso take a position related to Birkhoff's aesthetic measure. The degree to which an image resists JPEG compression is considered an "image complexity" measure. The degree it resists fractal compression is considered to be proportional to the "processing complexity" that will tax an observer's perceptual resources. Image complexity is then essentially divided by processing complexity to calculate a single fitness value.

Machado and Cardoso reported surprisingly good imaging results using evolving expressions with their complexity-based fitness function. But the authors were also careful to note that their fitness function only considers one formulaic aspect of aesthetic value. They posit that cultural factors ignored by NEvAr are critical to aesthetics. In later versions of NEvAr a user guided interactive mode was added (Machado and Cardoso 2002; 2003, Machado et al. 2005, see also Chap. 11 in this volume for their extended work in this vein).

10.2.9 Automated Fitness Functions in Evolutionary Music Systems

For evolutionary music composition some have calculated fitness scores using only evaluative rules regarding intervals, tonal centres, and compliance to key and meter. Others, like *GenOrchestra*, are hybrid systems that also include some form of listener evaluation. The GenOrchestra authors note that unfortunately without human evaluation "the produced tunes do not yet correspond to a really human-like musical composition" (Khalifa and Foster 2006, De Felice and Fabio Abbattista 2002).

Others have used music theory-based fitness functions for evolutionary bass harmonisation (De Prisco and Zaccagnino 2009), or to evolve generative grammar expressions for music composition (Reddin et al. 2009). For mimetic evolutionary music synthesiser programming McDermott et al. (2005) used a combination of perceptual measures, spectral analysis, and sample-level comparison as a fitness function to match a known timbre.

Weinberg et al. (2009) have created a genetically based robotic percussionist named *Haile* that can "listen" and trade parts in the call and response tradition. Rather than starting with a randomised population of musical gestures Haile begins with a pool of pre-composed phrases. This allows Haile to immediately produce musically useful responses. As Haile runs, however, the evolutionary system will create variations in real time. The fitness function used for selection uses an algorithm called *dynamic time warping*.

Dynamic time warping here provides a way to measure the similarity between two sequences that may differ in length or tempo. In response to a short rhythmic phrase played by a human performer, Haile applies the dynamic time warping-based fitness function to its population of responses and then plays back the closest match. The goal is not to duplicate what the human player has performed, but simply to craft a response that is aesthetically related and thus will contribute to a well-integrated performance.

10.2.10 Multi-objective Aesthetic Fitness Functions in Evolutionary Systems

Aesthetic judgements are typically multidimensional. For example, evaluating a traditional painting involves formal issues regarding colour, line, volume, balance, and so on. A fitness function that has to include multiple objectives like these will typically have a sub-score for each. Each sub-score will be multiplied by its own coefficient that serves as a weight indicating its relative importance. The weighted sub-scores are then summed for a final total score.

However, the weights are typically set in an ad hoc manner, and resulting evaluations may not push the best work to the front. And there is no reason to assume that the weights should maintain a static linear relationship regardless of the sub-score values. For example, various aspects of composition may influence the relative importance of colour.

Pareto ranking can address some of these concerns as an alternative to simple weights. In Pareto ranking one set of scores is said to *dominate* another if it is at least as good in all component sub-scores, and better in at least one. A *rank 1* set of scores is one that isn't dominated. When there are multiple objectives there will typically be multiple rank 1 sets of scores. The dimension of the problem they dominate is what differentiates rank 1 genotypes, and all can be considered viable.

Genotypes of less than rank 1 can be considered redundant. Note, however, that some redundancy in the gene pool is usually considered a good thing. In situations where a single genotype must be selected, a rank 1 genotype is sometimes selected based on its uniqueness relative to the current population (Neufeld et al. 2008, Ross and Zhu 2004, Greenfeld 2003).

Both weighting and Pareto ranking are approaches to the more general problem of multi-objective optimisation. For multidimensional aesthetics a computational evaluation system will have to deal with multi-objective optimisation either explicitly as above, or implicitly as is done in the extensions to evolutionary computation noted below.

10.2.11 Biologically Inspired Extensions to Simple Evolutionary Computation

Evolutionary art faces significant challenges beyond machine evaluation-based fitness functions. For example, the expression of genes in nature doesn't happen in a single step. There is a cascading sequence of emergence across a number of scales from DNA, to proteins, organelles, cells, tissues, and finally organs resulting in an individual. Life's capacity for complexification is unmatched in the known universe. By comparison evolutionary computing systems are simple in that they typically only support a single level of emergence, i.e. the genotype directly generates the phenotype (Casti 1994, Galanter 2010).

And so current evolutionary computing technologies have a very limited capacity for the creation of complexity. This isn't a problem in most industrial applications because their solution spaces are well explored by the search and optimisation strategies evolutionary computing offers. But art is one of the most complex activities of the arguably most complex unitary system known, the human mind.

A number of nature-inspired extensions for evolutionary art have been explored in part to meet this need for increased complexity. Each suggests new perspectives regarding computational aesthetic evaluation. For example, with the addition of coevolution two or more species may compete for survival. This can create an evolutionary "arms race" making fitness a moving target for all. But it is also possible that species will coevolve to fill mutually beneficial symbiotic roles, and possibly exhibit convergent aesthetics. In such systems the ecology is a dynamic system offering increased complexity. Some species will specialise and dominate an ecological niche while others remain flexible generalists. And some species may in fact alter the ecology creating a niche for itself. Meanwhile, within a species individuals may interact via social transactions further modulating what constitutes fitness. These extensions are explored in the following sections.

10.2.11.1 Coevolution

Coevolution in evolutionary art and design has been investigated since at least 1995. Poon and Maher (1997) note that in design a fixed solution space is undesirable because the problem itself is often reformed based on interim discoveries. They suggest that both the problem space and solution space evolve with each providing feedback to the other. Each genotype in the population can combine a problem model and a solution in a single chromosome. Or there can be two populations, one for problems and one for solutions. Then current best solutions are used to select problem formulations, and current best problem formulations are used to select solutions. Both methods allow a form of multi-objective optimisation where the problem emphasis can shift and suggest multiple solutions, and well-matched problem formulations and solutions will evolve.

One challenge with coevolutionary systems is deciding when to stop the iterative process and accept a solution. The authors note termination can be based on satisfactory resolution of the initial problem, but that such an approach loses the benefit of the coevolved problem space. Other termination conditions can include the amount of execution time allowed, equilibrium where both the solution and problem spaces no longer exhibit significant change, or where a set of solutions cycle. The last case can indicate the formation of a Pareto-optimal surface of viable solutions (Poon and Maher 1997).

Todd and Werner were early adopters of a coevolutionary approach to music composition. Prior to their work there had been attempts to create fitness functions based on rule-based or learning-based *critics*. But such critics typically encouraged compositions that were too random, too static, or otherwise quite inferior to most human composition. It's worth remembering that genes in evolutionary systems seek high fitness scores and only secondarily produce desirable compositions. Sometimes trivial or degenerate compositions will exploit brittle models or faulty simulations, thereby "cheating" to gain a high score without providing a useful result.

Based on the evolution of bird songs through sexual selection, the system devised by Todd and Werner consists of virtual male composers that produce songs and virtual female critics that judge the songs for the purpose of mate selection. Each female maintains an expectation table of probabilities for every possible note-to-note transition. This table is used to judge males' songs in three ways. The first two methods reward males the more they match the female's expectations. In the third method males are rewarded for surprising females. And for each of these three methods transition tables can be static, or they can coevolve and slowly vary with each new generation of females.

The first two matching methods quickly suffered from a lack of both short term and long term variety. However, rewarding surprise lead to greater variety. One might expect that rewarding surprise would encourage random songs. But this didn't happen because random songs accidentally contain more non-surprise elements than songs specifically structured to set up expectations and then defy them.

Initially the females were created with transition tables derived from folk songs. At first this resulted in human-like songs. But the authors note:

> One of the biggest problems with our coevolutionary approach is that, by removing the human influence from the critics (aside from those in the initial generation of folk-song derived transition tables), the system can rapidly evolve its own unconstrained aesthetics. After a few generations of coevolving songs and preferences, the female critics may be pleased only by musical sequences that the human user would find worthless.

Todd and Werner suggest that adding some basic musical rules might encourage diversity while also encouraging songs that are human-like. Additionally a learning and cultural aspect could be added by allowing individual females to change their transition tables based on the songs they hear (Todd and Werner 1998).

Greenfield (2008b) has presented an overview of coevolutionary methods used in evolutionary art including some unpublished systems made by Steven Rooke. Rooke first evolved critics by training them to match his manually given scores for a training set of images. The critics then coevolve with new images. Individual critics are scored by comparing their evaluations to those of previous critics. Critics are maintained over time in a sliding window of 20 previous generations. Rooke found that while the coevolved critics duplicated his taste, the overall system didn't innovate by exploring new forms.

Greenfield then describes his own system where images and 10×10 convolution filters are coevolved. Parasite filters survive by generating result images similar to original. Images survive by making the parasite filter results visible. A number of subtleties require attention such as setting thresholds that define similarity, the elimination of do-nothing filters, adjusting the evolutionary rates of parasites versus images, and the balancing of unary and binary operators to control high frequency banding. He cites Ficici and Pollack (1998) and confirms observing *evolutionary cycling*, where genotypes are rediscovered again and again, and *mediocre stable* states where the coevolving populations exhibit constant change with little improvement. Greenfield notes:

> In all of the examples we have seen: (1) it required an extraordinary effort to design a population to coevolve in conjunction with the population of visual art works being produced by an underlying image generation system, and (2) it was difficult to find an evaluation scheme that made artistic sense. Much of the problem with the latter arises as a consequence of the fact that there is very little data available to suggest algorithms for evaluating aesthetic fitness... It would be desirable to have better cognitive science arguments for justifying measurements of aesthetic content.

In later sections we will survey some of the work in psychology and the nascent field of neuroaesthetics that may contribute to computational aesthetic evaluation as Greenfield suggests.

10.2.11.2 Niche Construction by Agents

As discussed in McCormack and Bown (2009) an environment can be thought of as having both *properties* and *resources*. Properties are environmental conditions such as temperature or pH, and resources are available consumables required by organisms such as water and specific kinds of food. Each organism will have specific needs as to the properties and resources it requires of its environment. A given organism's preferred properties and resources define its ecological niche.

In typical "artificial life" systems evolutionary computing is implemented within the context of a simulated ecosystem. In those systems adaptation to ecological niches can increase diversity and enhance multi-objective optimisation. But beyond simple adaptation genotypes within a species can actively construct niches to their own advantage. McCormack and Bown have demonstrated both a drawing system and a music system that exploit niche construction.

In the first system drawing agents move leaving marks, are stopped when they intersect already existing marks, and sense the local density of already existing marks. Each agent also has a genetic preference for a given density. Initially agents that prefer low density will succeed in dividing large open sections of the canvas. Over time some agents will create higher densities of marks, which in turn act as constructed niches for progeny with a predisposition for high density. As a result some, but not all, sections of the canvas become increasingly dense and provide niches for high-density genotypes. The visual result exhibits a wide range of densities. Similar agent-based systems without niche construction tend to create drawings with homogeneous density. This system is further discussed in Chap. 2.

In the second system a single row of cells is connected head-to-tail as a toroid. Each cell generates a sine wave creating a single frequency tone. A line runs through all of the cells, and at each cell the line height is mapped into the loudness of its sine wave. Agents inhabit the cells, and each has a genetic preference for line height and slope. Each agent applies these preferences as pressure to the line in its cell as well as the cell to its left. Depending on the local state of their niche, i.e. the line height and slope in their cell, agents will stay alive and reproduce or die and not pass on their genotype. This sets up a dynamic system with localities that benefit certain genotypes. Those genotypes then modify the ecosystem, i.e. the line, to the benefit of their progeny. The resulting sound exhibits a surprising diversity of dynamics even though it is initialised at zero. As with many evolutionary and generative systems, this is due to the random variation in the initial population of agents.

10.2.11.3 Agent Swarm Behaviour

In most of the evolutionary systems discussed so far there is no interaction between phenotypes. Each is independently evaluated via user selection or fitness function. Other than this comparison a given phenotype has no impact on another. When phenotypes begin to interact in other ways, typically in the context of a simulated ecosystem, they can be thought of as simulated organisms or *agents* that exhibit behaviours. With niche creation agents modify their ecology establishing a mediated form of agent interaction. But agents can also interact directly creating an emergent group behaviour or *swarm behaviour*.

The canonical natural example of such an agent is the ant. An ant colony uses swarm intelligence to optimise the gathering and retrieval of food. As an ant finds food and brings it back to the nest it selectively leaves a chemical pheromone trail. Other ants happening upon the chemical trail will follow it, in effect joining a food retrieval swarm. Each ant adds more pheromone as they retrieve food. Because the pheromone spreads as it dissipates ants will discover short cuts if the initial path has excessive winding. In turn those short cuts will become reinforced with additional pheromone. Once the food is gone the ants stop laying down pheromone as they leave the now depleted site, and soon the pheromone trail will disappear. This behaviour can be simulated in software agents (Resnick 1994).

Artists have simulated this behaviour in software using agents that lay down permanent virtual pigment as well as temporary virtual pheromone trails. Variation and some degree of aesthetic control can be gained by breeding the ant-agents using an interactive evolutionary system (Monmarché et al. 2003).

Greenfield (2005a) automates the fitness function based on a performance metric regarding the number of cells visited randomly or due to pheromone following behaviour. Measuring fitness based only on the number of unique cells visited results in "monochromatic degeneracies". Rewarding only pheromone following creates a slightly more attractive blotchy style. Various weightings of both behaviours produce the best aesthetic results exhibiting organic and layered forms.

Urbano (2006) has produced striking colourful patterns using virtual micropainters he calls "Gaugants". In the course of one-to-one transactions his agents exert force, form consensus, or exhibit dissidence regarding paint colour. The dynamics are somewhat reminiscent of scenarios studied in game theory. Elzenga's agents are called "Arties". They exhibit mutual attraction/repulsion behaviour based on multiple sensing channels and genetic predisposition. The exhibited emergence is difficult to anticipate, but the artist can influence the outcome by making manual selections from within the gene pool (Elzenga and Pontecorvo 1999).

10.2.11.4 Curious Agents

Saunders and Gero (2004), and Saunders (2002) have extended swarming agents to create what they have called *curious agents*. They first note that agents in swarm simulations such as the above are mostly reactive. Flocking was originally developed by Reynolds (1987) and then extended by Helbing and Molnar (1995; 1997) to add *social forces* such as goals, drives to maximise efficiency and minimise discomfort, and so on. Social forces have been shown, for example, to create advantages in foot traffic simulation.

Sanders and Gero expand the dynamics of aesthetic evaluation behaviour by adding curiosity as a new social force. Their implementation uses a pipeline of six primary modules for sensing, learning, detecting novelty, calculating interest, planning, and acting. Sensing provides a way to sample the world for stimulus patterns. Learning involves classifying a pattern and updating prototypes kept in long term memory. Novelty is assessed as the degree to which error or divergence from previous prototypes is detected. Based on novelty a measure of interest is calculated.

Changes in interest result in goals being updated, and the current ever-changing goals determine movement.

Unsupervised artificial neural networks are used for classification, and classification error for new inputs is interpreted as novelty. But greater novelty doesn't necessarily result in greater interest. The psychologist Daniel Berlyne proposed that piquing interest requires a balance of similarity to previous experience and novelty. So, as suggested by Berlyne (1960; 1971), a *Wundt curve* is used to provide the metric for this balance and produces an appropriate interest measure. More about Berlyne's work follows in Sect. 10.3.2.

Based on this model Sanders created an experimental simulation where agents enter a gallery, can sense other agents, and can also view the colours of monochrome paintings hanging on nearby walls. There are also unseen monochrome paintings with new colours in other rooms. Along with other social behaviours agents learn the colours presented in one room, and then are potentially curious about new colours in other rooms. Depending on the sequence of colour exposure and the related Wundt-curve mapping, agents may or may not develop an interest and move to other areas.

10.2.11.5 Human Aesthetics, Meta-aesthetics, and Alternatives to Fitness Functions

Commenting on systems like those above using coevolution, niche creation, swarms, and curiosity Dorin (2005) notes:

> ... the "ecosystemic" approach permits simultaneous, multidirectional and automatic exploration of a space of virtual agent traits without any need for a pre-specified fitness function. Instead, the fitness function is implicit in the design of the agents, their virtual environment, and its physics and chemistry.

This avoids the problem of creating a computational aesthetic evaluation system by hand, and allows for the creation of evolutionary systems that generate surprising diversity and increased dynamics. Thus, if the goal is the creation of robust systems for meta-aesthetic exploration these evolutionary system extensions seem to be quite beneficial.

However, if the goal is to evolve results that appeal to our human sense of aesthetics there is no reason to think that will happen. Recall the earlier differentiation between human aesthetic evaluation and meta-aesthetic explorations. Creating evolutionary diversity and dynamics via artificial aesthetics foreign to our human sensibility is one thing. Appealing to human aesthetics is quite another. As observed by Todd and others, to date extensions and emergent aesthetics like those above do not provide machine evaluation that mirrors human aesthetic perception.

10.2.12 Complexity Based Models of Aesthetics

One of the recurring themes in computational aesthetics is the notion that aesthetic value has something to do with a balance of complexity and order. Birkhoff's aesthetic measure proposed the simple ratio $M = O/C$ where M is the measure of aesthetic effectiveness, O is the degree of order, and C is the degree of complexity.

But what is complexity? And what is order? Birkhoff suggested that these are proxies for the effort required (complexity) and the tension released (order) as perceptual cognition does its work. As a practical matter Birkhoff quantified complexity and order using counting operations appropriate to the type of work in question. For example, in his study of polygonal compositions complexity was determined by counting the number of edges and corners. His formula for order was:

$$O = V + E + R + HV - F \tag{10.2}$$

Here he sums the vertical symmetry (V), equilibrium (E), rotational symmetry (R), horizontal-vertical relation (HV), and unsatisfactory or ambiguous form (F). These notions of complexity and order at first appear to be formulaic and objective, but they nevertheless require subjective decisions when quantified.

In an attempt to add conceptual and quantitative rigour, Bense (1965) and Moles (1966) restated Birkhoff's general concept in the context of Shannon (1948)'s information theory creating the study of *information aesthetics*. Shannon was interested in communication channels and the quantification of information capacity and signal redundancy. From this point of view an entirely unpredictable random signal maximises information and complexity, and offers no redundancy or opportunity for lossless compression. In this context disorder or randomness is also called *entropy*. Extending this, Moles equated low entropy with order, redundancy, compressibility, and predictability. High entropy was equated with disorder, complexity, incompressibility, and surprise (see Chap. 3 for further discussion of information aesthetics).

As previously noted, Machado (1998) has updated this approach by calculating aesthetic value as the ratio of *image complexity* to *processing complexity*. Processing complexity refers to the amount of cognitive effort that is required to take in the image. Image complexity is intrinsic to the structure of the image. This lead them to propose functional measures where image complexity is inversely proportional to JPEG compressibility and processing complexity is directly proportional to fractal compressibility.

With the advent of complexity science as a discipline defining order and complexity has become much more problematic. This account begins with *algorithmic complexity* or *algorithmic information content* as independently developed by Kolmogorov (1965), Solomonoff (1964), Chaitin (1966). In this paradigm the complexity of an object or event is proportional to the size of the shortest program on a universal computer that can duplicate it. From this point of view the most complex music would be white noise and the most complex digital image would be random pixels. Like information complexity, algorithmic complexity is inversely proportional to order and compressibility.

For physicist Murray Gell-Mann the information and algorithmic notions of complexity don't square with our experience. When we encounter complex objects or situations they aren't random. Despite being difficult to predict they also have some degree of order maintaining integrity and persistence.

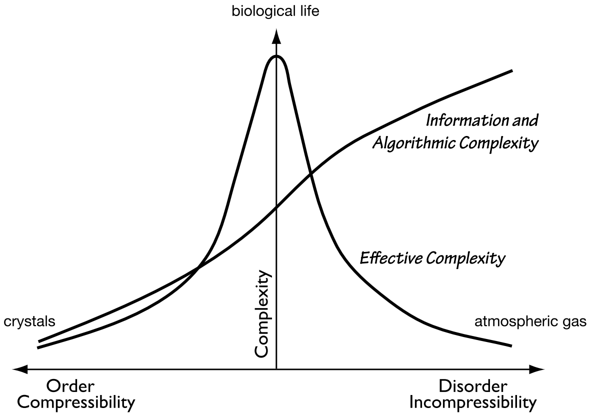


Fig. 10.1 Information and algorithmic complexity increase monotonically with increasing disorder. Effective complexity peaks where there is a mix of order and disorder such as is found in biological life

Consider two situations, one where there is a living frog and another where there is a long dead and decaying frog. The decaying frog has greater entropy because relative to the living frog it is more disordered, and over time it will become more even more disordered to the point where it will no longer be identifiable as a frog at all. Intuitively we would identify the living frog as being more complex. It displays a repertoire of behaviours, operates a complex system of biochemistry to process food, water, and oxygen to generate energy and restore tissues, maintains and exchanges large amounts of genetic information in the course of reproduction, and so on. Along with these orderly processes the frog remains flexible and unpredictable enough to be adaptive and avoid becoming easy prey. In terms of entropy our highly complex living frog is somewhere between simple highly ordered crystals and simple highly disordered atmospheric gases.

To better capture our intuitive sense of complexity Gell-Mann has proposed the notion of *effective complexity*, a quantity that is greatest when there is a balance of order and disorder such as that found in the biological world (Gell-Mann 1995). Unlike information and algorithmic complexity, effective complexity is not inversely proportional to order and compressibility. Rather both order and disorder contribute to complexity (Fig. 10.1, please note that this graph is only meant as a qualitative illustration with somewhat arbitrary contours).

Complexity science continues to offer new paradigms and definitions of complexity. In a 1998 lecture by Feldman and Crutchfield at the Santa Fe Institute well over a dozen competing theories were presented (Feldman and Crutchfield 1998)—the debate over complexity paradigms continues. Measuring aesthetic value as a relationship between complexity and order is no longer the simple proposition it once seemed to be. (For an alternate view of complexity and aesthetics see Chap. 12.)

Artists working in any media constantly seek a balance between order and disorder, i.e. between fulfilling expectations and providing surprises. Too much of the former leads to boredom, but too much of the latter loses the audience. It is a dynamic that applies to visual art, music, and the performing arts alike. And it helps differentiate genres in that styles that cater to established expectations are considered to be more "traditional" while styles that serve up more unorthodox surprises are considered to be "cutting edge."

Notions of Shannon information and algorithmic complexity have their place. But in aesthetics it is misleading to treat order and complexity as if they are polar opposites. My suggestion is that the notion of effective complexity better captures the balance of order and disorder, of expectation and surprise, so important in the arts. This offers the challenge and potential benefit that effective complexity can serve as a measure of quality in computational aesthetic evaluation.

10.3 The Future of Computational Aesthetic Evaluation

As should be obvious by now, computational aesthetic evaluation is a very difficult and fundamentally unsolved problem. To date any marginal successes have tended towards narrow application niches using methods that do not generalise very well.

The irony is that aesthetic evaluation is something we all do quite naturally. Could it be that the solution to the computational aesthetic evaluation problem is within us and just not yet understood?

Artists and engineers have always learned from nature. There is a significant and growing literature around the psychology and neurology of aesthetics. But this challenge to understanding seems no less daunting than the difficulty of machine evaluation. The human brain that gives rise to the human mind is arguably the most complex unitary system known. The brain includes approximately 10^{15} neural connections. In addition, recent research regarding the brain's glial cells reveals that they contribute to active information processing rather than, as previously thought, merely providing mechanical support and insulation for neurones. Glial cells make up 90 % of the brain and some scientists speculate that they are specifically engaged in creative thought (Koob 2009). Computing hardware can only make up for part of this gap by exploiting electronic switching speeds that are about 10^7 times faster than human neurones.

Nevertheless, it seems reasonable that an improved understanding of natural aesthetic perception will contribute to computational aesthetic evaluation efforts, and science has made some significant progress in this regard. Perhaps a good place to start is recent scientific thinking as to the origins of human aesthetics.

10.3.1 The Origins of Art and the Art Instinct

Denis Dutton notes that evolutionary scientist Stephen Jay Gould claims that art is essentially a nonadaptive side effect, what Gould calls a *spandrel*, resulting from an excess of brain capacity brought about by unrelated adaptations. Dutton (2009) argues that the universality of both art making behaviour and some aesthetic preferences imply a more direct genetic linkage and something he calls the *art instinct*.

Dutton points out that like language every culture has art. And both language and art have developed far beyond what would be required for mere survival. The proposed explanation for the runaway development of language is that initially language provided a tool for cooperation and survival. Once language skills became important for survival language, fluency became a mate selection marker. The genetic feedback loop due to mate selection then generated ever-increasing language ability in the population leading to a corresponding *language instinct* (Pinker 1994).

Additionally, Dutton posits that early human mate selection was, in part, based on the demonstration of the ability to provide for material needs. Like language, this ability then became a survival marker in mate selection subject to increasing development. Just as a peacock's feather display marks a desirable surplus of health, works of art became status symbols demonstrating an excess of material means. It is not by coincidence then that art tends to require rare or expensive materials, significant time for learning and making, as well as intelligence and creativity. And typically art has a lack of utility, and sometimes an ephemeral nature. All of these require a material surplus.

One could argue that even if art making has a genetic basis it may be that our sense of aesthetics does not. In this regard, Dutton notes the universal appeal, regardless of the individual's local environment, for landscape scenes involving open green spaces trees and ample bodies of water near by, an unimpeded view of the horizon, animal life, and a diversity of flowering and fruiting plants. This scene resembles the African savannah where early man's evolution split off from other primate lines. It also includes numerous positive cues for survivability. Along with related psychological scholarship Dutton quotes the previously noted Alexander Melamid:

> … I'm thinking that this blue landscape is more serious than we first believed… almost everyone you talk to directly—and we've already talked to hundreds of people—they have this blue landscape in their head… So I'm wondering, maybe the blue landscape is genetically imprinted in us, that it's the paradise within, that we came from the blue landscape and we want it… We now completed polls in many countries—China, Kenya, Iceland, and so on—and the results are strikingly similar.

That our aesthetic capacity evolved in support of mate selection has parallels in other animals. This provides some hope for those who would follow a psychological path to computational aesthetic evaluation, because creatures with simpler brains than man practice mate selection. In other words perhaps the computational equivalent of a bird or an insect is "all" that is required for computational aesthetic evaluation. But does mate selection behaviour in other animals really imply brain activity similar to human aesthetic judgement? One suggestive study by Watanabe (2009) began with a set of children's paintings. Adult humans judged each to be "good" or "bad". Pigeons were then trained through operant conditioning to only peck at good paintings. The pigeons were then exposed for the first time to a new set of already judged children's paintings. The pigeons were quite able to correctly classify the previously unseen paintings as "good" or "bad".

10.3.2 Psychological Models of Human Aesthetics

Conspicuously missing from most work by those pursuing machine evaluation that mimics human aesthetics are models of how natural aesthetic evaluation occurs. Rudolf Arnheim, Daniel Berlyne, and Colin Martindale are three researchers who stand out for their attempts to shape the findings of empirical aesthetics into general aesthetic models that predict and explain. Each has left a legacy of significant breadth and depth that may inform computational aesthetic evaluation research. The following sections provide an introduction to their contributions.

10.3.2.1 Arnheim—Gestalt and Aesthetics

If one had to identify a single unifying theme for Arnheim it would have to be the notion of perception as cognition. Perception isn't something that happens to the brain when events in the world are passively received through the senses. Perception is an activity of the brain and nothing short of a form of cognition. And it is this perceptual cognition that serves as the engine for gestalt phenomena.

First written in 1954 and then completely revised in 1974, Arnheim's book *Art and Visual Perception: A Psychology of the Creative Eye* established the relevance of gestalt phenomena as art and design principles (Arnheim 1974). *The law of prägnanz* in gestalt states that the process of perceptual cognition endeavours to order experience into wholes that maximise clarity of structure. From this law come the notions of closure, proximity, containment, grouping, and so on now taught as design principles (Wertheimer 2007).

The neurological mechanisms behind these principles were not, and still are not, well understood. Arnheim wrote of *forces* and *fields* as existing both as psychological and physical entities; the physical aspects being neurological phenomenon in the brain itself. Some have suggested it is more useful to take these terms metaphorically to describe the dynamic tensions that art exercises (Cupchik 2007).

Arnheim's theory of aesthetics is much more descriptive than normative. Nevertheless, those interested in computational aesthetic evaluation have much to take away with them. That perception is an active cognitive process, and that the gestalt whole is something more than the sum of the parts, is now taken by most as a given. And the difference between maximising clarity of structure and maximising simplicity of structure is a nuance worthy of attention (Verstegen 2007).

10.3.2.2 Berlyne—Arousal Potential and Preferences

Daniel E. Berlyne published broadly in psychology, but his work of note here regards physiological arousal and aesthetic experience as a neurological process (Konečni 1978). One of Berlyne's significant contributions is the concept of *arousal potential* and its relationship to *hedonic response*.

Arousal potential is a property of stimulus patterns and a measure of the capability of that stimulus to arouse the nervous system. Arousal potential has three sources. First, there are *psychophysical* properties such as very bright light, very loud sounds, sensations with an abrupt onset, very low or high frequency sounds, and so on. Second, there are *ecological* stimuli such as survival threats like pain or predator sightings, or cues associated with the availability of food. But the third and strongest according to Berlyne are referred to as *collative* effects. These are combined and comparative experiences that present arousal potential in a context dependent and relative manner. Examples include "novelty, surprisingness, complexity, ambiguity, and puzzlingness." Berlyne (1971) explicitly notes the correspondence between many of these collative effects and concepts from Shannon's information theory.

The hedonic response to sources of arousal potential refers to the spectrum of pleasure and pain we experience. Berlyne proposes that the hedonic response is the result of separate and distinct reward and aversion systems. Each of these systems is made up of neurones. The firing thresholds of individual neurones will vary according to the *normal* or *Gaussian probability* distribution as is typical in nature (see Fig. 10.2). Therefore the strength of the arousal potential will determine the number of neurones that fire in response. The number of neurones responding will increase as a *Gaussian cumulative distribution*, i.e. the area under the Gaussian probability distribution as the threshold moves from left to right. Berlyne further proposes that the reward system requires less arousal potential exposure to activate, but that when activated the aversion system will produce a larger response.

The result is the hedonic response as a summation of the positive reward system and the negative aversion system. With no arousal potential there is a hedonic response of indifference. As more arousal potential is presented the hedonic response increases manifesting itself as a pleasurable experience. Beyond a certain point, however, two things happen. First, the reward system reaches maximum activation and plateaus. Second, the aversion system begins to activate. As the aversion system reaches higher levels of activation the hedonic response will lessen and eventually cross into increasing levels of pain.

Berlyne notes that this function is usually called *the Wundt curve*, as it was first presented by the "father of experimental psychology" Wilhelm Wundt in 1874. But in Wundt's model the x-axis represents low-level neural intensity. Berlyne's arousal potential on the x-axis includes psychophysical intensity, but it also includes ecological stimuli and most importantly collative effects. For Berlyne increasing collative effects such as novelty and surprise also represent increasing complexity in the information theory sense. From this point of view works of only moderate information complexity maximise the hedonic response. This resonates well with the intuitive artistic notion that audiences respond best to works that are not so static as to be boring, and yet also operate within learned conventions so as to not be experienced as chaotic.

There is, however, another interpretation. The notion of Gell-Mann's effective complexity was previously mentioned. From that point of view complexity is a balance of order and disorder, and biological life presents complexity at its peak.

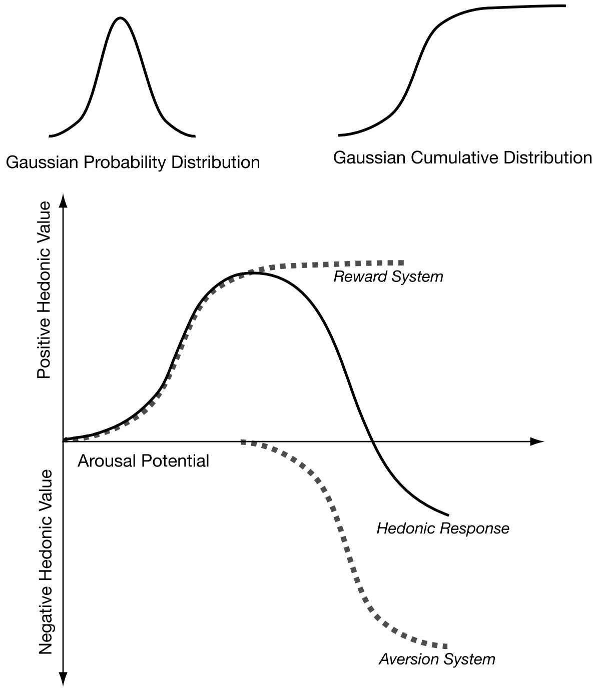


Fig. 10.2 Wundt curve as applied by Berlyne. Redrawn from Berlyne (1971)

The Wundt and effective complexity curves both peak in the middle suggesting that positive hedonic response may be proportional to effective complexity. Effective complexity has, in a sense, the balance of order and disorder "built in." One might hypothesise that the most important and challenging survival transactions for humans have to do with other living things and especially fellow humans. Perhaps that created evolutionary pressure leading to the optimisation of the human nervous system for effective complexity, and human aesthetics and related neurological reward/aversion systems reflect that optimisation.

10.3.2.3 Martindale—Prototypicality and Neural Networks

Colin Martindale was an active empiricist and in 1990 he published a series of articles documenting experiments intended to verify the arousal potential model of Berlyne. Martindale et al. (1990) notes:

> Berlyne... developed an influential theory that has dominated the field of experimental aesthetics for the past several decades... Berlyne is often cited in an uncritical manner. That is, he is taken as having set forth a theory based upon well-established facts rather than, as he actually did, as having proposed tentative hypotheses in need of further testing. The result has been a stifling of research on basic questions concerning preference, because these questions are considered to have been already answered. In this article, we report a series of experiments that test obvious predictions drawn from Berlyne's theory. It was in the firm

> expectation of easily confirming these predictions that we undertook the experiments. The results are clear-cut. They do not support the theory.

The debate pitting collative effects versus prototypicality would dominate experimental aesthetics for almost 20 years (North and Hargreaves 2000). For some Berlyne's notion of collative effects was especially problematic. First it was odd for a behaviourist like Berlyne to make an appeal to a concept so much about the inner state of the individual. Additionally, terms like *novelty* and *complexity* were problematic both in specification and mechanism.

However, Martindale's primary critique was empirical. For example, contrary to Berlyne's model he found that psychophysical, ecological, and collative properties are not additive, nor can they be traded off. Significantly more often than not empirically measured responses do not follow the inverted-U of the Wundt curve, but are monotonically increasing. Finally, a number of studies showed that meaning rather than pure sensory stimulation is the primary determinant of aesthetic preference (Martindale et al. 1990; 2005, Martindale 1988b).

In a series of publications Martindale (1981; 1984; 1988a; 1991) developed a natural neural network model of aesthetic perception that is much more consistent with experimental observation. Martindale first posits that neurones form nodes that accept, process, and pass on stimulation from lower to higher levels of cognition. Shallow sensory and perceptual processing tends to be ignored. It is the higher semantic nodes, the nodes that encode for meaning, that have the greatest strength in determining preference. Should the work carry significant emotive impact the limbic system can become engaged and dominate the subjective aesthetic experience.

Nodes are described as specialised recognition units connected in an excitatory manner to nodes corresponding to superordinate categories. So, for example, while one is reading nodes that extract features will excite nodes for letters, and they will in turn excite nodes for syllables or letter groupings, leading to the excitation of nodes for words, and so on. Nodes at the same level, however, will have a lateral inhibitory effect. Nodes encoding for similar stimuli will be physically closer together than unrelated nodes. So nodes encoding similar and related exemplars will tend towards the centre of a *semantic field*. The result is that the overall nervous system will be optimally activated when presented an unambiguous stimulus that matches a prototypically specific and strong path up the neural hierarchy (Martindale 1988b).

Commenting on prototypicality North and Hargreaves (2000) explain:

> ... preference is determined by the extent to which a particular stimulus is typical of its class, and explanations of this have tended to invoke neural network models of human cognition: this approach claims that preference is positively related to prototypicality because typical stimuli give rise to stronger activation of the salient cognitive categories.

Martindale's neural network prototypicality model carries with it great explanatory and predictive power. Towards the end of his life he penned a chapter describing the results of 25 widely disparate empirical studies, and how his single model can provide a foundation for understanding all of them (Martindale 2007).

While most in the field agree that Martindale's prototypicality model explains more of the empirical data than Berlyne's collative effect model, some cases remain where prototypicality is the weaker explanation. Some have suggested ways to reconcile the two models to provide more cover than either can alone (North and Hargreaves 2000, Whitfield 2000).

10.3.3 Empirical Studies of Human Aesthetics

Along with unifying theories such as those offered by Arnheim, Berlyne, and Martindale, the field of psychology offers a vast catalogue of very specific findings from experimental aesthetics. It is difficult in aesthetics research to identify and control the myriad factors that may influence hedonic response. And because human subjects are typically required it is difficult to achieve large sample sizes. Nevertheless empirical studies of human aesthetics seem to be on the increase, and many are highly suggestive and worth consideration by those interested in computational aesthetic evaluation.

Empirical studies of human aesthetics usually focus on viewers, artists, or objects. Studies of viewers have to account for audiences that are expert and not. Some experiments focus on the impact setting has on aesthetic perception. Others are attempts to correlate aesthetic response with social or personality factors. Studies of artists usually focus on aspects of divergent thinking, creativity, and self-critical abilities. Studies of objects typically include some form of analysis relative to a hypothesised aesthetic mechanism.

A full or even representative cataloguing of these studies is unfortunately well outside of the scope of this chapter. What stands out in reading the literature though is the large number of variables that determine or shade human aesthetic experience. For example:

- Subjects first asked to think about the distant future are more likely to accept unconventional works as art than those who first think about their near future (Schimmel and Forster 2008).
- A hedonic contrast effect has been established in music listening. In absolute terms the same music will be evaluated more positively if preceded by bad music, and less positively if preceded by good music (Parker et al. 2008).
- Not all emotions lend themselves to musical expression. Those that do tend to be general, mood based, and don't require causal understanding (Collier 2002).
- Individual preference differences can form on the basis of experience. Relative to non-professionals, photo professionals exhibit a greater ability to process photographic information, and show a relative preference for photographs that are uncertain and unfamiliar (Axelsson 2007).
- Artists and non-artists were presented with a sequence of 22 work-in-process images leading to Matisse's 1935 painting, *Large Reclining Nude*. Non-artists judged the painting as getting generally worse over time consistent with the increasing abstraction of the image. In contrast, art students' judgements showed a jagged trajectory with several peaks suggesting an interactive hypothesis-testing process (Kozbelt 2006).

- Whether isolated or within a larger composition, note intervals in music carry significant and consistent emotional meaning. There is also softer evidence that these interval-emotional relationships are universal across different times, cultures, and musical traditions. Speculation is that this is related to universal aspects of vocal expression (Oelmann and Laeng 2009).

10.3.4 Neuroaesthetics

Beginning with Birkhoff, and throughout this chapter, neurology has frequently been the backstory for aesthetic and computational aesthetic evaluation models described at higher levels of abstraction. To some extent Arnheim, and certainly Berlyne and Martindale, all had in mind neurological models as the engines of aesthetic perception. In no small part due to new imaging technologies such as functional magnetic resonance imaging (fMRI), positron emission tomography (PET) scanning, and functional near-infrared imaging (fNIR), science seems to be preparing to take on perhaps the deepest mystery we face everyday, our own minds.

It is in this context that the relatively new field of neuroaesthetics has come into being (Skov and Vartanian 2009a). Neuroaesthetics is the study of the neurological bases for all aesthetic behaviour including the arts. A fundamental issue in neuroaesthetics is fixing the appropriate level of inspection for a given question. It may be that the study of individual neurones will illuminate certain aspects of aesthetics. Other cases may require a systems view of various brain centres and their respective interoperation.

A better understanding of representation in the brain could illuminate not only issues in human aesthetics but more generally all cognition. This in turn may find application not only in computational aesthetic evaluation, but also broadly across various artificial intelligence challenges. And finally, a better understanding of neurology will likely suggest new models explaining human emotion in aesthetic experience. If we better understand the aesthetic contributions of both the cortex and the limbic system, we will be better prepared to create machine evaluation systems that can address both the Dionysian and the Apollonian in art (Skov and Vartanian 2009b).

10.3.5 Computing Inspired by Neurology

Computer science has felt the influence of biology and brain science from its earliest days. The theoretical work of Von Neumann and Burks (1966) towards a universal constructor was an exploration of computational reproduction and evolution. Turing (1950) proposed a test essentially offering an operational definition for machine intelligence. Turing also invented the reaction diffusion model of biological morphogenesis, and towards the end of that article he discuses implementing a computer simulation of it (Turing 1952). Computing models inspired by neurology have fallen in and out of fashion, from Rosenblatt's early work on the perceptron (Rosenblatt 1962), to Minsky and Papert's critique (Minsky and Papert 1969), and to the later successful development of non-linear models using backpropagation and self-organisation.

A number of artificial neural network applications already noted showed only limited success as either a fitness function or a standalone machine evaluation system. It would be premature to conclude such use in has hit a permanent plateau. But it would be glib to suggest that since the brain is a neural network that the successful use of artificial neural networks for computational aesthetic evaluation is inevitable. The brain's 10^{15} neural connections and presently unknown glial cell capacity presents a daunting quantitative advantage artificial systems will not match any time soon.

Perhaps a better understanding of natural neurology and subsequent application to connectionist technologies can help overcome what present artificial systems lack in quantity. This is the approach Jeff Hawkins has taken in the development of hierarchical temporal memory.

10.3.6 The Neocortex and Hierarchical Temporal Memory

Hawkins has proposed the *hierarchical temporal memory model* for the functionality found in the neocortex of the brain. He proposes that this single mechanism is used for all manner of higher brain function including perception, language, creativity, memory, cognition, association, and so on. He begins with a typical hierarchical model where lower cortical levels aggregate inputs and pass the results up to higher levels corresponding to increasing degrees of abstraction (Hawkins and Blakeslee 2004).

Neurologists know that the neocortex consists of a repeating structure of six layers of cells. Hawkins has assigned each layer with functionality consistent with the noted multi-level hierarchical structure. What Hawkins has added is that within a given level higher layers constantly make local predictions as to what the next signals passed upward will be. This prediction is based on recent signals and local synapse strength. Correct predictions strengthen connections within that level. Thus the neocortex operates as a type of hierarchical associative memory system, and it exploits the passage of time to create local feedback loops for constant training.

Artificial hierarchical temporal memory has been implemented as software called *NuPIC*. It has been successfully demonstrated in a number of computer vision applications where it can robustly identify and track moving objects, as well as extract patterns in both physical transportation and website traffic (Numenta 2008). To date NuPIC seems to work best when applied to computer vision problems, but others have adapted the hierarchical temporal memory model in software for temporal patterns in music (Maxwell et al. 2009).

10.3.7 Computer Architectures for Evolvable Hardware

Another promising technology is reconfigurable hardware that evolves in a way to best solve the problem at hand. Evolvable hardware exploits programmable circuit devices such as *field programmable gate arrays* (FPGAs). These are integrated circuit chips with a large number of simple logic units or *gates*. Settable switches called *architecture bits* or *configuration memory* program the logical function and interconnection of these gates. Field programmable gate arrays allow the mass manufacture of standardised silicon that has its circuit-level functionality postponed for later definition. This circuit-level functionality is lower and faster than that achieved by executing machine language code (Yao and Higuchi 1997).

By treating the architecture bits as a chromosome the configuration of field programmable gate arrays can be determined using evolutionary methods. Evolution in this case doesn't design the gate array configuration so much as it designs the chip's behaviour relative to some fitness function defined need. In this some see a parallel to the way neurones exhibit emergent learning. And because these chips can be reprogrammed on the fly there is the possibility of learning adaptation to changing conditions.

It's worth noting that a proposed evolvable hardware system has been simulated in software, and used as a pattern recognition system for facial recognition with an experimental accuracy of 96.25 % (Glette et al. 2007).

10.4 Conclusion

Computational aesthetic evaluation victories have been few and far between. The successful applications have mostly been narrowly focused point solutions. Negative experience to date with low dimensional models such as formulaic and geometric theories makes success with similar approaches in the future quite unlikely.

Evolutionary methods, including those with extensions such as coevolution, niche construction, and agent swarm behaviour and curiosity, have had some circumscribed success. The noted extensions have allowed evolutionary art to iterate many generations quickly by eliminating the need for interactive fitness evaluation. They have also allowed researchers to gain insight into how aesthetic values can be created as emergent properties. In such explorations, however, the emergent artificial aesthetics themselves seem alien and unrelated to human notions of beauty. They have not yet provided practical leverage when the goal is to model, simulate, or predict human aesthetics via machine evaluation.

I've suggested that a paradigm like effective complexity may be more useful than information or algorithmic complexity when thinking about aesthetics. Effective complexity comes with the notion of balancing order and disorder "built in", and that balance is critical in all forms of aesthetic perception and the arts.

There is also a plausible evolutionary hypothesis for suggesting that effective complexity correlates well with aesthetic value. Effective complexity is maximised in the very biological systems that present us with our greatest opportunities and challenges. Hence there is great survival value in having a sensory system optimised for the processing of such complexity. There is also additional survival value in our experiencing such processing as being pleasurable. As in other neurological reward systems such pleasure directs our attention to where it is needed most.

The fields of psychology and neurology have been noted as possible sources of help for future work in computational aesthetic evaluation. Models of aesthetic perception such as those from Arnheim, Berlyne, and especially Martindale invite computational adaptation. Results from empirical studies of human aesthetics can stimulate our thinking about computational evaluation. At the same time they warn us that aesthetic evaluation in humans is highly variable depending on setting, context, training, expectations, presentation, and likely dozens of other factors.

Will robust human-like computational aesthetic evaluation be possible someday? There is currently no deductive proof that machine evaluation either is or isn't possible in principle. Presumably an argument for impossibility would have to establish as key an aspect of the brain or human experience that goes beyond mechanical cause and effect. Others might argue that because the brain itself is a machine our aesthetic experience is proof enough that computational aesthetic evaluation is possible. These in-principle arguments parallel philosophical issues regarding phenomenology and consciousness that are still in dispute and far from settled.

As a practical matter, what is currently possible is quite limited. The one consistent thread that for some will suggest a future direction relates to connectionist approaches. The current leading psychological model, Martindale's prototypicality, presents natural aesthetic evaluation as a neural network phenomenon. We know that animals with natural neural systems much simpler than those in the human brain are capable of some forms of aesthetic evaluation. In software, new connectionist computing paradigms such as hierarchical temporal memory show promise for both higher performance and closer functional equivalency with natural neural systems. In hardware we are beginning to see systems that can dynamically adapt to problem domains at the lowest gate level. Perhaps this will all someday lead to a synergy of hardware, software, and conceptual models yielding success in computational aesthetic evaluation.

Acknowledgements My interest in writing this chapter began at the "Computational Creativity: An Interdisciplinary Approach" seminar in July of 2009 at the Schloss Dagstuhl—Leibniz Center for Informatics. I would like to thank Margaret Boden, Mark d'Inverno and Jon McCormack for organising the seminar. In addition my thanks go to my fellow members of the "Evaluation" discussion group at the seminar including Margaret Boden, David Brown, Paul Brown, Harold Cohen, and Oliver Deussen. Finally I enjoyed and appreciated the lively post-seminar e-mail discussion of related topics with David Brown, Paul Brown, Harold Cohen, Jon McCormack, and Frieder Nake. Please note, however, that any matters of opinion or error in this chapter are purely my own.

References

Aguilar, C., & Lipson, H. (2008). A robotic system for interpreting images into painted artwork.

In C. Soddu (Ed.), *International conference on generative art* (Vol. 11). Generative Design Lab, Milan Polytechnic.
Aldiss, B. (2002). The mechanical turk—the true story of the chess-playing machine that changed the world. *TLS-the Times Literary Supplement*, *5170*, 33.
Alsing, R. (2008). *Genetic programming: evolution of Mona Lisa*. http://rogeralsing.com/2008/12/07/genetic-programming-evolution-of-mona-lisa/. Accessed 7/21/2011.
Arnheim, R. (1974). *Art and visual perception: a psychology of the creative eye* (new, expanded and revised ed.) Berkeley: University of California Press.
Atiyeh, B., & Hayek, S. (2008). Numeric expression of aesthetics and beauty. *Aesthetic Plastic Surgery*, *32*(2), 209–216.
Axelsson, O. (2007). Individual differences in preferences to photographs. *Psychology of Aesthetics, Creativity, and the Arts*, *1*(2), 61–72.
Baluja, S., Pomerleau, D., & Jochem, T. (1994). Towards automated artificial evolution for computer-generated images. *Connection Science*, *6*(1), 325–354.
Bense, M. (1965). *Aesthetica; Einfhrung in die neue Aesthetik*. Baden-Baden: Agis-Verlag.
Bentley, P., & Corne, D. (2002). An introduction to creative evolutionary systems. In P. Bentley & D. Corne (Eds.), *Creative evolutionary systems* (pp. 1–75). San Francisco/San Diego: Morgan Kaufmann/Academic Press.
Berlyne, D. E. (1960). *Conflict, arousal, and curiosity*. New York: McGraw-Hill.
Berlyne, D. E. (1971). *Aesthetics and psychobiology*. New York: Appleton-Century-Crofts.
Birkhoff, G. D. (1933). *Aesthetic measure*. Cambridge: Harvard University Press.
Boselie, F., & Leeuwenberg, E. (1985). Birkhoff revisited: beauty as a function of effect and means. *The American Journal of Psychology*, *98*(1), 1–39.
Carroll, N. (1999). *Philosophy of art: a contemporary introduction, Routledge contemporary introductions to philosophy*. London: Routledge.
Casti, J. L. (1994). *Complexification: explaining a paradoxical world through the science of surprise* (1st ed.). New York: HarperCollins.
Chaitin, G. J. (1966). On the length of programs for computing finite binary sequences. *Journal of the ACM*, *13*(4), 547–569.
Ciesielski, V. (2007). Evolution of animated photomosaics. In *Lecture notes in computer science* (vol. 4448, pp. 498–507).
Collier, G. L. (2002). Why does music express only some emotions? A test of a philosophical theory. *Empirical Studies of the Arts*, *20*(1), 21–31.
Cupchik, G. C. (2007). A critical reflection on Arnheim's gestalt theory of aesthetics. *Psychology of Aesthetics, Creativity, and the Arts*, *1*(1), 16–24.
Datta, R., Joshi, D., Li, J., & Wang, J. Z. (2006). Studying aesthetics in photographic images using a computational approach. In *Proceedings: Vol. 3953. ECCV 2006* (Pt. 3, pp. 288–301).
Datta, R., Li, J., & Wang, J. Z. (2007). Learning the consensus on visual quality for next-generation image management. In *Proceedings of the ACM multimedia conference* (pp. 533–536). New York: ACM.
Davis, T., & Rebelo, P. (2007). Environments for sonic ecologies. In *Applications of evolutionary computing* (pp. 508–516). Berlin: Springer.
De Prisco, R., & Zaccagnino, R. (2009). An evolutionary music composer algorithm for bass harmonization. In *Applications of evolutionary computing* (Vol. 5484, pp. 567–572). Berlin: Springer.
Dorin, A. (2005). Enriching aesthetics with artificial life. In A. Adamatzky & M. Komosinski (Eds.), *Artificial life models in software* (pp. 415–431). London: Springer. Chap. 14.
Draves, S. (2005). The electric sheep screen-saver: A case study in aesthetic evolution. In *Lecture notes in computer science: Vol. 3449. Evo workshops* (pp. 458–467).
Dutton, D. (2009). *The art instinct: beauty, pleasure, and human evolution* (1st U.S. ed.). New York: Bloomsbury Press.
Elzenga, R. N., & Pontecorvo, M. S. (1999). *Arties: meta-design as evolving colonies of artistic agents*. Generative Design Lab.

De Felice, F., & Fabio Abbattista, F. S. (2002). Genorchestra: an interactive evolutionary agent for musical composition. In C. Soddu (Ed.), *International conference on generative art* (Vol. 5). Generative Design Lab, Milan Polytechnic.

Feldman, D. P., & Crutchfield, J. (1998). *A survey of complexity measures*. Santa Fe Institute.

Ficici, S., & Pollack, J. (1998). Challenges in co-evolutionary learning; arms-race dynamics, open-endedness, and mediocre stable states. In C. Adami (Ed.), *Artificial life VI: proceedings of the sixth international conference on artificial life* (pp. 238–247). Cambridge: MIT Press.

Fogel, L. J. (1999). *Intelligence through simulated evolution: forty years of evolutionary programming*. *Wiley series on intelligent systems*. New York: Wiley.

Fornari, J. (2007). Creating soundscapes using evolutionary spatial control. In *Lecture notes in computer science* (Vol. 4448, pp. 517–526).

Galanter, P. (2010). The problem with evolutionary art is. In C. DiChio, A. Brabazon, G. A. DiCaro, M. Ebner, M. Farooq, A. Fink, J. Grahl, G. Greenfield, P. Machado, M. O'Neill, E. Tarantino, & N. Urquhart (Eds.), *Lecture notes in computer science: Vol. 6025*. *Applications of evolutionary computation, pt. II, proceedings* (pp. 321–330). Berlin: Springer.

Gartland-Jones, A. (2002). Can a genetic algorithm think like a composer? In C. Soddu (Ed.), *International conference on generative art* (Vol. 5). Generative Design Lab, Milan Polytechnic.

Gedeon, T. (2008). Neural network for modeling esthetic selection. In *Lecture notes in computer science* (Vol. 4985(2), pp. 666–674).

Gell-Mann, M. (1995). What is complexity? *Complexity*, *1*(1), 16–19.

Glette, K., Torresen, J., & Yasunaga, M. (2007). An online EHW pattern recognition system applied to face image recognition. In *Applications of evolutionary computing* (pp. 271–280). Berlin: Springer.

Greenfeld, G. R. (2003). Evolving aesthetic images using multiobjective optimization. In *CEC: 2003 congress on evolutionary computation* (pp. 1903–1909).

Greenfield, G. (2005a). Evolutionary methods for ant colony paintings. In *Lecture notes in computer science: Vol. 3449*. *Evo workshops* (pp. 478–487).

Greenfield, G. (2005b). On the origins of the term computational aesthetics. In *Computational aesthetics 2005: Eurographics workshop on computational aesthetics in graphics, visualization and imaging*, Girona, Spain, 18–20 May, 2005. Eurographics.

Greenfield, G. (2008a). Evolved diffusion limited aggregation compositions. In *Applications of evolutionary computing* (pp. 402–411). New York: Springer.

Greenfield, G. R. (2004). The void series—generative art using regulatory genes. In C. Soddu (Ed.), *International conference on generative art* (Vol. 7). Generative Design Lab, Milan Polytechnic.

Greenfield, G. R. (2008b). Co-evolutionary methods in evolutionary art. In J. Romero & P. Machado (Eds.), *Natural computing series*. *The art of artificial evolution* (pp. 357–380). Berlin: Springer.

Hawkins, J., & Blakeslee, S. (2004). *On intelligence* (1st ed.). New York: Times Books.

Hazan, A., Ramirez, R., Maestre, E., Perez, A., & Pertusa, A. (2006). Modelling expressive performance: a regression tree approach based on strongly typed genetic programming. In *Applications of evolutionary computing* (pp. 676–687). Berlin: Springer.

Helbing, D., & Molnar, P. (1995). Social force model for pedestrian dynamics. *Physical Review*, *E*(51), 4282–4286.

Helbing, D., & Molnar, P. (1997). Self-organization phenomena in pedestrian crowds. In F. Schweitzer (Ed.), *Self-organization of complex structures: from individual to collective dynamics* (pp. 569–577). London: Gordon and Breach.

Hoenig, F. (2005). Defining computational aesthetics. In L. Neumann, M. Sbert & B. Gooch (Eds.), *Computational aesthetics in graphics, visualization and imaging*, Girona, Spain.

Holger, H. (1997). Why a special issue on the golden section hypothesis? An introduction. *Empirical Studies of the Arts*, *15*.

Hönn, M., & Göz, G. (2007). The ideal of facial beauty: a review. *Journal of Orofacial Orthopedics/Fortschritte der Kieferorthopdie*, *68*(1), 6–16.

Hornby, G. S., & Pollack, J. B. (2001). The advantages of generative grammatical encodings for physical design. In *Proceedings of the 2001 congress on evolutionary computation* (Vol. 601, pp. 600–607).

Jaskowski, W. (2007). Learning and recognition of hand-drawn shapes using generative genetic programming. In *Lecture notes in computer science* (Vol. 4448, pp. 281–290).

Khalifa, Y., & Foster, R. (2006). A two-stage autonomous evolutionary music composer. In *Lecture notes in computer science: Vol. 3907*. *Evo workshops* (pp. 717–721).

Kolmogorov, A. N. (1965). Three approaches to the quantitative definition of information. *Problems in Information Transmission*, *1*, 1–7.

Komar, V., Melamid, A., & Wypijewski, J. (1997). *Painting by numbers: Komar and Melamid's scientific guide to art* (1st ed.). New York: Farrar Straus Giroux.

Konečni, V. J. (1978). Daniel E. Berlyne: 1924–1976. *The American Journal of Psychology*, *91*(1), 133–137.

Koob, A. (2009). The root of thought: what do glial cells do? http://www.scientificamerican.com/article.cfm?id=the-root-of-thought-what. Accessed 11/29/09.

Koza, J. R., Bennett, F. H. I., Andre, D., & Keane, M. A. (2002). Genetic programming: biologically inspired computation that exhibits creativity in producing human-competitive results. In P. Bentley & D. Corne (Eds.), *Creative evolutionary systems* (pp. 275–298). San Francisco/San Diego: Morgan Kaufmann/Academic Press.

Kozbelt, A. (2006). Dynamic evaluation of Matisse's 1935 large reclining nude. *Empirical Studies of the Arts*, *24*(2), 119–137.

Law, E., & Phon-Amnuaisuk, S. (2008). Towards music fitness evaluation with the hierarchical SOM. In *Applications of evolutionary computing* (pp. 443–452). Berlin: Springer.

Li, Y.-F., & Zhang, X.-R. (2004). Quantitative and rational research for the sense quantum—research of the order factors for color harmony aesthetic. *Journal of Shanghai University (English Edition)*, *8*(2), 203–207.

Livio, M. (2003). *The golden ratio: the story of phi, the world's most astonishing number* (1st ed.). New York: Broadway Books.

Machado, P. (1998) Computing aesthetics. In *Lecture notes in artificial intelligence*: Vol. 1515.

Machado, P., & Cardoso, A. (2002). All the truth about NEvAr. *Applied Intelligence*, *16*(2), 101–118.

Machado, P., & Cardoso, A. (2003). *NEvAr system overview*. Generative design lab, Milan Polytechnic.

Machado, P., Romero, J., Cardoso, A., & Santos, A. (2005). Partially interactive evolutionary artists. *New Generation Computing*, *23*(2), 143–155.

Machado, P., Romero, J., & Manaris, B. (2008). Experiments in computational aesthetics—an iterative approach to stylistic change in evolutionary art. In J. Romero & P. Machado (Eds.), *The art of artificial evolution: a handbook on evolutionary art and music* (pp. 311–332). Berlin: Springer.

Machado, P., Romero, J., Santos, A., Cardoso, A., & Pazos, A. (2007). On the development of evolutionary artificial artists. *Computers and Graphics*, *31*(6), 818–826.

Machado, P., Romero, J., Santos, M. L., Cardoso, A., & Manaris, B. (2004). Adaptive critics for evolutionary artists. In *Lecture notes in computer science*. *Applications of evolutionary computing* (pp. 437–446). Berlin: Springer.

Machwe, A. T. (2007). Towards an interactive, generative design system: integrating a 'build and evolve' approach with machine learning for complex freeform design. In *Lecture notes in computer science* (Vol. 4448, pp. 449–458).

Magnus, C. (2006). Evolutionary musique concrete. In F. Rothlauf & J. Branke (Eds.), *Applications of evolutionary computing, EvoWorkshops 2006* (pp. 688–695). Berlin: Springer.

Manaris, B., Machado, P., McCauley, C., Romero, J., & Krehbiel, D. (2005). Developing fitness functions for pleasant music: Zipf's law and interactive evolution systems. In *Lecture notes in computer science: Vol. 3449*. *Evo workshops* (pp. 498–507).

Manaris, B., Vaughan, D., Wagner, C., Romero, J., & Davis, R. B. (2003). Evolutionary music and the Zipf-Mandelbrot law: developing fitness functions for pleasant music. *Applications of*

Evolutionary Computing, *2611*, 522–534.
Martindale, C. (1981). *Cognition and consciousness*. *The Dorsey series in psychology*. Homewood: Dorsey Press.
Martindale, C. (1984). The pleasures of thought: a theory of cognitive hedonics. *Journal of Mind and Behavior*, *5*(1), 49–80.
Martindale, C. (1988a). Cognition, psychobiology, and aesthetics. In F. H. Farley & R. W. Neperud (Eds.), *The foundations of aesthetics, art, and art education* (pp. 7–42). New York: Praeger Publishers.
Martindale, C. (1988b). Relationship of preference judgements to typicality, novelty, and mere exposure. *Empirical Studies of the Arts*, *6*(1), 79–96.
Martindale, C. (1991). *Cognitive psychology: a neural-network approach*. Pacific Grove: Brooks/Cole Publishing Company.
Martindale, C. (2007). A neural-network theory of beauty. In C. Martindale, P. Locher & V. Petrov (Eds.), *Evolutionary and neurocognitive approaches to aesthetics, creativity, and the arts* (pp. 181–194). Amityville: Baywood.
Martindale, C., Moore, K., & Anderson, K. (2005). The effect of extraneous stimulation on aesthetic preference. *Empirical Studies of the Arts*, *23*(2), 83–91.
Martindale, C., Moore, K., & Borkum, J. (1990). Aesthetic preference: anomalous findings for Berlyne's psychobiological theory. *The American Journal of Psychology*, *103*(1), 53–80.
Maxwell, J. B., Pasquier, P., & Eigenfeldt, A. (2009). Hierarchical sequential memory for music: a cognitive model. In *International society for music information retrieval*.
McCormack, J. (2005) Open problems in evolutionary music and art. In *Lecture notes in computer science: Vol. 3449*. *Evo workshops* (pp. 428–436).
McCormack, J. (2008). Facing the future: evolutionary possibilities for human-machine creativity. In J. Romero & P. Machado (Eds.), *The art of artificial evolution: a handbook on evolutionary art and music* (pp. 417–451). Berlin: Springer.
McCormack, J., & Bown, O. (2009) Life's what you make: Niche construction and evolutionary art. In *Lecture notes in computer science: Vol. 5484*. *Evo workshops* (pp. 528–537).
McDermott, J., Griffith, N. J. L., & O'Neill, M. (2005). Toward user-directed evolution of sound synthesis parameters. In *Lecture notes in computer science: Vol. 3449*. *Evo workshops* (pp. 517–526).
Minsky, M. L., & Papert, S. (1969). *Perceptrons; an introduction to computational geometry*. Cambridge: MIT Press.
Mitchell, T. J., & Pipe, A. G. (2005). Convergence synthesis of dynamic frequency modulation tones using an evolution strategy. In *Applications on evolutionary computing* (pp. 533–538). Berlin: Springer.
Moles, A. A. (1966). *Information theory and esthetic perception*. Urbana: University of Illinois Press.
Monmarché, N., Aupetit, S., Bordeau, V., Slimane, M., & Venturini, G. (2003). Interactive evolution of ant paintings. In B. McKay et al. (Eds.), *Congress on evolutionary computation* (Vol. 2, pp. 1376–1383). New York: IEEE Press.
Mori, T., Endou, Y., & Nakayama, A. (1996). Fractal analysis and aesthetic evaluation of geometrically overlapping patterns. *Textile Research Journal*, *66*(9), 581–586.
Neufeld, C., Ross, B. J., & Ralph, W. (2008). The evolution of artistic filters. In J. Romero & P. Machado (Eds.), *The art of artificial evolution: a handbook on evolutionary art and music* (pp. 335–356). Berlin: Springer.
North, A. C., & Hargreaves, D. J. (2000). Collative variables versus prototypically. *Empirical Studies of the Arts*, *18*(1), 13–17.
Numenta (2008). Advanced nupic programming. http://www.numenta.com/for-developers/software/pdf/nupic_prog_guide.pdf. Accessed 16/04/10.
Oelmann, H., & Laeng, B. (2009). The emotional meaning of harmonic intervals. *Cognitive Processing*, *10*(2), 113–131.
Parker, S., Bascom, J., Rabinovitz, B., & Zellner, D. (2008). Positive and negative hedonic contrast with musical stimuli. *Psychology of Aesthetics, Creativity, and the Arts*, *2*(3), 171–174.

Peitgen, H.-O., Jürgens, H., & Saupe, D. (1992). *Chaos and fractals: new frontiers of science*. New York: Springer.

Phon-Amnuaisuk, S. (2007). Evolving music generation with SOM-fitness genetic programming. In *Lecture notes in computer science* (Vol. 4448, pp. 557–566).

Pinker, S. (1994). *The language instinct* (1st ed.). New York: Morrow.

Poon, J., & Maher, M. L. (1997). Co-evolution and emergence in design. *Artificial Intelligence in Engineering*, *11*(3), 319–327.

Reddin, J., McDermott, J., & O'Neill, M. (2009). Elevated pitch: automated grammatical evolution of short compositions. In *Lecture notes in computer science: Vol. 5484*. *EvoWorkshops 2009* (pp. 579–584).

Resnick, M. (1994). *Complex adaptive systems*. *Turtles, termites, and traffic jams: explorations in massively parallel microworlds*. Cambridge: MIT Press.

Reynolds, C. (1987). Flocks, herds, and schools: a distributed behavioural model. *Computer Graphics*, *21*(4), 25–34.

Romero, J., Machado, P., & Santos, M. L. (2003). *Artificial music critics*. Generative Design Lab, Milan Polytechnic.

Rosenblatt, F. (1962). *Principles of neurodynamics; perceptrons and the theory of brain mechanisms*. Washington: Spartan Books.

Ross, A. (1995). Poll stars. *ArtForum*, *33*(5), 72–77.

Ross, B. J., & Zhu, H. (2004). Procedural texture evolution using multi-objective optimization. *New Generation Computing*, *22*(3), 271–293.

Saunders, R. (2002). *Curious design agents and artificial creativity*. PhD thesis, University of Sydney.

Saunders, R., & Gero, J. S. (2004). Curious agents and situated design evaluations. *AI Edam-Artificial Intelligence for Engineering Design Analysis and Manufacturing*, *18*(2), 153–161.

Scha, R., & Bod, R. (1993). Computationele esthetica. *Informatie en Informatiebeleid*, *11*(1), 54–63.

Schimmel, K., & Forster, J. (2008). How temporal distance changes novices' attitudes towards unconventional arts. *Psychology of Aesthetics, Creativity, and the Arts*, *2*(1), 53–60.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, *27*(3), 379–423.

Sims, K. (1991). Artificial evolution for computer-graphics. *Siggraph '91 Proceedings 25*, 319–328.

Sims, K. (1994). Evolving virtual creatures. *Siggraph '94 Proceedings*, *28*, 15–22.

Sims, K. (1997). Galapagos interactive exhibit. http://www.karlsims.com/galapagos/index.html. Accessed 11/16/2010.

Skov, M., & Vartanian, O. (2009a). Introduction—what is neuroaesthetics? In M. Skov & O. Vartanian (Eds.), *Neuroaesthetics—foundations and frontiers in aesthetics* (pp. iv, 302 p.). Amityville: Baywood.

Skov, M., & Vartanian, O. (2009b). *Neuroaesthetics, foundations and frontiers in aesthetics*, Amityville: Baywood.

Solomonoff, R. J. (1964). A formal theory of inductive inference, part I and part II. *Information and Control*, *7*, 1–22. 224–254.

Standage, T. (2002). *The mechanical turk: the true story of the chess-playing machine that fooled the world*. London: Allen Lane.

Staudek, T. (1999). *On Birkhoff's aesthetic measure of vases* (Vol. 2009). Faculty of Informatics, Masaryk University.

Stewart, M. (2008). *Launching the imagination: a comprehensive guide to basic design* (3rd ed.). Boston: McGraw-Hill Higher Education.

Sullivan, L. H. (1896). The tall office building artistically considered. *Lippincott's Magazine*, *57*, 403–409.

Takagi, H. (2001). Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation. *Proceedings of the IEEE*, *89*(9), 1275–1296.

Taylor, R. P. (2006). *Chaos, fractals, nature: a new look at Jackson Pollock*. Eugene: Fractals Research.
Todd, P. M. (1989). A connectionist approach to algorithmic composition. *Computer Music Journal*, *13*(4), 27–43.
Todd, P., & Werner, G. (1998). Frankensteinian methods for evolutionary music composition. In N. Griffith & P. Todd (Eds.), *Musical networks: parallel distributed perception and performance*. Cambridge: MIT Press/Bradford Books.
Todd, S., & Latham, W. (1992). *Evolutionary art and computers*. London: Academic Press.
Tsai, H.-C., Hung, C.-Y., & Hung, F.-K. (2007). Automatic product color design using genetic searching. In *Computer-aided architectural design futures (CAADFutures) 2007* (pp. 513–524). Berlin: Springer.
Tufte, G., & Gangvik, E. (2008). Transformer #13: exploration and adaptation of evolution expressed in a dynamic sculpture. In *Applications of evolutionary computing* (pp. 509–514). Berlin: Springer.
Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, *59*(236), 433–460.
Turing, A. M. (1952). The chemical basis of morphogenesis. *Philosophical transactions—Royal Society. Biological Sciences*, *237*(641), 37–72.
Urbano, P. (2006) Consensual paintings. In *Lecture notes in computer science: Vol. 3907. Evo workshops* (pp. 622–632).
Verstegen, I. (2007). Rudolf Arnheim's contribution to gestalt psychology. *Psychology of Aesthetics, Creativity, and the Arts*, *1*(1), 8–15.
Von Neumann, J., & Burks, A. W. (1966). *Theory of self-reproducing automata*. Urbana: University of Illinois Press.
Voss, R. F., & Clarke, J. (1975). 1/F-noise in music and speech. *Nature*, *258*(5533), 317–318.
Watanabe, S. (2009). Pigeons can discriminate "good" and "bad" paintings by children. *Animal Cognition*, *13*(1).
Weinberg, G., Godfrey, M., Rae, A., & Rhoads, J. (2009). A real-time genetic algorithm in human-robot musical improvisation. In *Computer music modeling and retrieval. Sense of sounds* (pp. 351–359). Berlin: Springer.
Wertheimer, M. (2007). Rudolf Arnheim: an elegant artistic gestalt. *Psychology of Aesthetics, Creativity, and the Arts*, *1*(1), 6–7.
Whitelaw, M. (2003). Morphogenetics: generative processes in the work of driessens and verstappen. *Digital Creativity*, *14*(1), 43–53.
Whitfield, T. W. A. (2000). Beyond prototypicality: toward a categorical-motivation model of aesthetics. *Empirical Studies of the Arts*, *18*(1), 1–11.
Wilson, D. J. (1939). An experimental investigation of Birkhoff's aesthetic measure. *The Journal of Abnormal and Social Psychology*, *34*(3), 390–394.
Wu, Y.-F., & Chien, S.-F. (2005). *Enemy character design in computer games using generative approach*. Generative Design Lab, Milan Polytechnic.
Yao, X., & Higuchi, T. (1997). Promises and challenges of evolvable hardware. In T. Higuchi (Ed.), *Evolvable systems: from biology to hardware* (Vol. 1259, pp. 55–78). Berlin: Springer.
Yee-King, M. (2007). An automated music improviser using a genetic algorithm driven synthesis engine. In M. Giacobini (Ed.), *Proceedings of the 2007 EvoWorkshops* (pp. 567–576). Berlin: Springer.
Yuan, J. (2008). Large population size IGAs with individuals' fitness not assigned by user. In *Lecture notes in computer science* (Vol. 5227, pp. 267–274).
Zipf, G. K. (1949). *Human behavior and the principle of least effort: an introduction to human ecology*. Cambridge: Addison-Wesley.

Chapter 11
Computing Aesthetics with Image Judgement Systems

Juan Romero, Penousal Machado, Adrian Carballal, and João Correia

Abstract The ability of human or artificial agents to evaluate their works, as well as the works of others, is an important aspect of creative behaviour, possibly even a requirement. In artistic fields such as visual arts and music, this evaluation capacity relies, at least partially, on aesthetic judgement. This chapter analyses issues regarding the development of computational systems that perform aesthetic judgements focusing on their validation. We present several alternatives, as follows: the use of psychological tests related to aesthetic judgement; the testing of these systems in style recognition tasks; and the assessment of the system's ability to predict the users' valuations or the popularity of a given work. An adaptive system is presented and its performance assessed using the above-mentioned validation methodologies.

11.1 Introduction

Creativity is frequently associated with the capacity to create artworks. Therefore, the design of computing systems which have the skills to create artworks can provide interesting insights into a general understanding of creativity. Spector and Alpern (1994) define a "Constructed Artist" as an entity that is "... supposed to be capable of creating aesthetically meritorious artworks on their own, with minimal

J. Romero (✉) · A. Carballal
Faculty of Computer Science, University of A Coruña, Campus de Elviña, CP 15071, A Coruña, Spain
e-mail: jj@udc.es

A. Carballal
e-mail: adrian.carballal@udc.es

P. Machado · J. Correia
Department of Informatics Engineering, University of Coimbra – Polo II, 3030-290 Coimbra, Portugal

P. Machado
e-mail: machado@dei.uc.pt

J. Correia
e-mail: jncor@student.dei.uc.pt

J. McCormack, M. d'Inverno (eds.), *Computers and Creativity*,
DOI 10.1007/978-3-642-31727-9_11,

human intervention", as opposed to other computational systems performing artistic tasks. Artistic processes often rely on the capacity to make aesthetic judgements, using artworks created by others as sources of inspiration and making criticism of their own work. As Boden (1990) puts it: "Someone that has a new idea must be able to evaluate it by itself".

A major obstacle in developing constructed artists is the difficulty of implementing aesthetic judgement mechanisms. Having a system capable of creating its own aesthetic preferences, or acquiring them from a cultural environment, would be an important step towards the development of computational creativity.

The concepts of art and aesthetics are deeply related. Nevertheless, it is important to differentiate between them. The artistic value of an artwork depends on several factors, including form, content, cultural context and novelty. We acknowledge the relevance of all these factors, yet, we focus exclusively on the aesthetic properties of the artworks, and—for the scope of this chapter—we define Aesthetics as the study of the form in itself, i.e. stripped from content, context, and all the other factors that, although relevant from an artistic standpoint, do not result exclusively from form and, consequently, cannot be analysed when considering only the form.

By assuming this point of view, we are not creating a false dichotomy between "form" and "content". We acknowledge that these factors are not independent. Form affects, and sometimes determines, the way content is perceived and conveyed, and the coherence or contrast between form and content can be explored. For instance, an artist may choose to use a composition that he finds visually pleasing and harmonious to convey content that is highly displeasing and violent, exploring the discrepancy between form and content for artistic purposes. Even when the artwork is purely abstract, one cannot rule out the possibility that a human observer perceives, even if only at a subconscious level, some type of content that evokes feelings and emotions and that, therefore, influences his reaction to the piece. In other words, it may be impossible for a human to focus exclusively on the form, which makes the discipline of aesthetics (as defined here) an unreachable goal. Although this constitutes an obvious drawback, it is also an opportunity: computers can focus exclusively on the form.

In the same way that we differentiate between Art and Aesthetics, we also differentiate between Artistic and Aesthetic Judgement. The existence of universal aesthetic preferences shared among all humans, the existence of shapes that are inherently pleasing or displeasing, the way culture and training affect aesthetics, etc. are controversial (even among the authors of this chapter). These questions, although relevant, are outside the scope of what we describe here. We consider, however, that there are properties such as symmetry, balance, rhythm, contrast, proportion, repetition, unity, predominance, variety, and continuity which are aesthetically relevant and that can be considered aesthetic principles. This does not imply that a symmetric image is inherently more pleasing than an asymmetric one. It does, however, imply that symmetry may influence the aesthetic value of an artwork. The way a given aesthetic property influences aesthetics depends on a wide variety of issues, including the relationship with other aesthetic properties, personal preferences, aesthetic trend, and so on.

We posit that the ability to recognise at least some of these aesthetic properties is common to all humans, acknowledging that the way different humans may react to different aesthetic principles, to their relationships, and value aesthetic principles may vary. Likewise, the degree of awareness to principles of aesthetical order and the inclination to use aesthetic criteria when valuing artefacts also differs.

In Machado et al. (2003) we find the following definition: Artificial Art Critics are "systems that are capable to see/listen to an artwork and perform some sort of evaluation of the perceived piece". Unfortunately, the term "art critic" can be easily misunderstood, given that it may be perceived as the equivalent of a human making an artistic critique or a written analysis of an artwork, rather than an aesthetic judgement. For this reason, we abandon this nomenclature.

Taking all of the above into consideration, for the scope of this chapter, we define an aesthetic judgement system (AJS) as a system that performs an aesthetic assessment of an image based on its aesthetic properties. For instance, a system that: measures the degree of accordance of an artwork with a given aesthetic theory; measures several aesthetic properties of an image; makes an assessment of an artwork according to the aesthetic preferences of a given user, set of users, community, etc.; identifies the aesthetic current of an artwork; assesses the aesthetic consistency of a set of works; etc.

It is important to note that the system should make its judgement based on aesthetic properties. A system that assesses the aesthetic value of an artwork by analysing its aesthetic properties can be considered an AJS. A system that performs the same task by using optical character recognition to identify the signed name of the author and determines aesthetic value by the popularity of the author cannot be considered an AJS.

An AJS may provide a quantitative judgement, e.g. a single numeric value, a vector, or a classification in one or more dimensions. An AJS may also provide a qualitative assessment or assessments. Ultimately, the adequacy of the output depends on the task at hand. For instance, to guide an evolutionary algorithm using roulette wheel selection, a quantitative judgement, or one that can be converted to quantities, is required. However, to guide the same algorithm using tournament selection, only a qualitative assessment is needed, i.e. knowing if a given individual is better suited to the task at hand than another, we do not need to quantify how much better it is.

The AJSs can be divided into two categories. The first category explores systems that rely on a theory of visual aesthetics and use an AJS to explore this theory by computing it, e.g. Rigau et al. (2008), Staudek (2002; 2003), Taylor et al. (1999), Machado and Cardoso (1998), Spehar et al. (2003), Schmidhuber (1997; 1998; 2007), see also the chapters by Galanter (Chap. 10) and Schmidhuber (Chap. 12) in this volume.

The second category presents learning systems which include some kind of adaptive capacity that potentially allows them to learn user preferences, trends, aesthetic theories, etc. Although there are different approaches, usually these systems extract information from images (e.g. a set of metrics) using a machine learning system that performs an aesthetics-based evaluation or classification. There are numerous examples of this architecture in the fields of content based image retrieval and computer vision, such as Datta et al. (2006; 2008), Ke et al. (2006), Cutzu et al. (2003). One of the advantages of this kind of systems is their potential use to perform different tasks, and to be adapted to different aesthetic preferences. Classification tasks are particularly useful for validation purposes since they tend to be objective and allow a direct comparison of the results obtained by several systems (provided that they are applied to the same datasets).

Relatively few attempts have been made in the visual arts field to integrate evaluation skills into an image generation system. Neufeld et al. (2007) presented a genetic programming engine generating non-photorealistic filters by means of a fitness function based on Ralph's bell curve distribution of colour gradient. This model was implemented by carrying out an empirical evaluation of hundreds of artworks. Their paper contains examples of some of the non-photorealistic filters created.

Kowaliw et al. (2009) compared biomorphs generated in three different ways: at random, through interactive evolution, and through evolution guided by a set of image metrics used in content based image retrieval. They compared the results of the three methods taking into account a model of creativity explained in Dorin and Korb (2009), coming to the conclusion that automatic methods gave rise to results comparable to those obtained by interactive evolution.

Baluja et al. (1994) used an artificial neural network trained with a set of images generated by user-guided evolution. Once trained, the artificial neural network was used to guide the evolutionary process by assigning fitness to individuals. Although the approach is inspiring, the authors consider the results somewhat disappointing.

Saunders (2001) used a similar approach, proposing the use of a Self Organising Map artificial neural network for the purpose of evolving images with a sufficient degree of novelty. This approach is restricted to the novelty aspects of artworks.

Svangård and Nordin (2004) made use of complexity estimates so as to model the user's preferences, implying that this scheme may be used for fitness assignment. The authors introduced some experiments in which they used sets of two randomly generated images, and compared, for each pair, the system's choices with those made by the user. Depending on the methodology used, the success rates ranged between 34 % and 75 %. Obviously, a result of 35 % is very low for a binary classification task. No example of the images considered was presented, which makes it impossible to evaluate the difficulty of the task and, as such, the appropriateness of the methodologies that obtained the highest averages. Additional information on the combination of AJSs in image generation systems can be found in Chap. 10 in this volume.

Although the integration of AJSs in image generation systems is an important goal, having autonomous, self-sufficient AJSs presents several advantages:

- It allows one to assess the performance of the AJSs independently, providing a method for comparing them. This allows a more precise assessment of the AJS abilities than possible when comparing AJSs integrated with image generation systems, since the strengths and weaknesses of the image generation systems may mask those of the AJS;

- It fosters cooperation among different working groups, allowing, for instance, the collaboration between research groups working on the development of AJS and groups that focus on the development of image generation systems;
- The same AJS may be incorporated with different systems allowing it to be used for various creativity supporting tasks.

This chapter focuses on AJS validation. The next section discusses some of the issues related to AJS validation and presents several validation methods based on psychological tests, users' evaluations, and stylistic principles. Section 11.3 describes the evolution of an AJS through time, from a heuristic based system to a learning AJS. The results obtained in the system validation by means of the approaches proposed in Sect. 11.2 are presented and analysed. Finally, we draw overall conclusions and indicate future work.

11.2 Validation Approaches for AJS

Performance comparison of two AJSs is a delicate task. The existence of a validation task to which both can be applied is a prerequisite for comparison. Unless the systems are applicable to the exact same task (which includes using the same datasets) the comparison may lead to erroneous conclusions. The validation method must be reproducible and the results should be numerically quantifiable. All components of the validation task (e.g. datasets) should be made accessible to the research community. Furthermore, it is also recommended that the datasets come from an external source (i.e. that they are not specifically made for a given AJS) and have an unbiased character. There are tasks, e.g. author identification, that despite not being directly related to the ability to make aesthetic assessments, can be useful due to their objectivity and can, potentially, complement other validation methods.

The characteristics of human aesthetic preferences—e.g. subjectivity, individuality, cultural biases, change through time, etc.—create an additional difficulty. Similarly, the interpretation of the results is also problematic and, in many circumstances, it is difficult to determine what constitutes a good result.

In this section we will explore three different ways to validate an AJS: based on psychological tests related to aesthetics, based on user evaluation, and based on stylistic classification.

11.2.1 Psychological Tests

There are several psychological tests aimed at measuring and identifying aesthetic preferences (Burt 1933) and aesthetic judgement (Savarese and Miller 1979, Furnham and Walker 2001). Some of them are employed on professional guidance, together with other psychological tests, in order to advise students about potential careers.

From the point of view of an AJS validation, they constitute a good reference, since they are relatively easy to apply and provide reproducible and quantifiable results. They also allow the comparison of the "performance" of the computer system with human evaluation, although this comparison is extremely delicate.

We will make a short analysis of two tests that are potentially useful for AJS validation, namely the *Visual Aesthetic Sensitivity Test* of Götz et al. and Maitland Graves' *Design Judgment Test*. Nadal (2007) provides further analysis of these and other psychological tests.

The Visual Aesthetic Sensitivity Test (VAST)—created by Götz (an artist) and Eysenck (Eysenck et al. 1984, Götz 1985, Eysenck 1983)—consists of a series of 50 pairs of non-representative drawings. In each pair the subject has to express an opinion as to which is the most harmonious design. Götz drew the "harmonious" designs first and then altered them by incorporating changes that he considered faults and errors according to his aesthetic views. The validity of the judgement was tested by eight expert judges (artists and critics), making preference judgements and only accepting pairs of designs on which agreement among judges was unanimous. When groups of subjects are tested, the majority judgement agrees with the keying of the items, which supports the validity of the original judgement.

There are easy, middle and difficult item levels. The difficulty level of items is established in terms of the percentage of correct responses; the more subjects give the right answer, the easier the item. Different groups of subjects, differing in age, sex, artistic training, cultural background, and ethnicity have produced very similar difficulty levels for the items. "The instructions of the test did not emphasise so much the individual's preference for one item or the other, but rather the quality of one design" (Eysenck 1983). The task is to discover which of the designs is the most harmonious and not which designs are the most pleasant. The images resemble abstract art, minimising the influence of content on preference. There was some cross-cultural comparison employing the VAST test. Iwawaki et al. (1979) compared Japanese and English children and students. Frois and Eysenck (1995) applied the test to Portuguese children and Fine Arts Students.

Graves (1946) presented "The Design Judgment Test" (DJT).[1] It was designed to determine how humans respond to several principles of aesthetic order, presented in his previous work (Graves 1951). It contains 90 slides with pairs or triads of images. In each of the slides, one particular image "is considered 'right' (and scored accordingly) on the basis of agreement with the author's theories and the agreement of art teachers on the superiority of that particular design" (Eysenck and Castle 1971). Thus, on each slide, one of the images follows the aesthetic principles described by Graves, while the others violate, at least, one of these principles. Each slide is shown for approximately 45–60 seconds to the subject, who chooses one image per slide. The score of the test corresponds to the number of correct choices. All slides are in black, white and green. All images are abstract. The images of each slide are similar in style and in terms of the elements present. The average percentage of correct

[1] Photos of DJT can be found at: http://www.flickr.com/photos/robgiampietro/sets/72157611584992173/with/3136292750/.

answers resulting from answering randomly to the test is 48.3 %, due to the fact that some of the items were made up of three images.

Graves (1948) reported that art students achieved higher scores in the test than non-art students. He stated that: "the test's ability to differentiate the art groups from the non-art groups is unmistakably clear". Eysenck and Castle (1971) obtained different results showing fewer differences between art and non-art students (64.4 % vs. 60 %) with variances below 4 % in all cases, and also different responses in males and females. Eysenck and Castle (1971) pointed out the "general climate of art teaching, which now tends to stress simplicity and regularity to a greater extent than 25 years ago" as a possible reason for the differences observed. The DJT test was used as an instrument by the career advisors of the Portuguese Institute for Employment and Vocational Training. According to the results found by this institute while validating the test for the Portuguese population, published in internal reports and provided to the career advisors, the results achieved in the DJT with randomly selected individuals yield an average percentage of 50.76 % correct answers. This score is similar to the one obtained by answering randomly to the test, which indicates its difficulty. If we consider students in the last years of Fine Arts degrees, the average increases up to 61.87 %. Nevertheless, Götz and Götz (1974) report that "22 different arts experts (designers, painters, sculptors) had 0.92 agreement on choice of preferred design, albeit being critical of them" (Chamorro-Premuzic and Furnham 2004).

Like in most psychological tests, one should exercise great care when interpreting the results. The fact that a subject obtains a higher score in the DJT than another does not imply that he has better aesthetic judgement skills. It can mean, for instance, that one of the subjects is making choices based on aesthetics while the other is not. For example, a structural engineer may be inclined to choose well-balanced and stable designs, systematically valuing these properties above all else and ignoring rhythm, contrast, dynamism, etc. because the balance of the structure is the key factor to him. The test has been used for career guidance based on the reasoning that a subject that consistently makes choices according to aesthetic criteria is likely to have a vocation for an art-related career.

The DJT is based on aesthetic principles which may not be universally accepted or applicable (Eysenck 1969, Eysenck and Castle 1971, Uduehi 1995). Additionally, even if the aesthetic principles are accepted, the ability of the test to assess them has been questioned (Eysenck and Castle 1971). The average results obtained by humans in these tests also vary between studies (Eysenck and Castle 1971, Uduehi 1995). Although this can be, at least partially, explained by the selection of participants and other exogenous factors, it makes it harder to understand what constitutes a good score in this test.

The ability of these tests to measure the aesthetic judgement skills of the subjects is not undisputed, nor are the aesthetic principles they indirectly subscribe. Nevertheless, they can still be valuable validation tests in the sense that they can be used to measure the ability of an AJS to capture the aesthetic proprieties explored in these tests and the degree of accordance with the aesthetic judgements they implicitly defend.

11.2.2 User Evaluation and Popularity Prediction

The most obvious way of validating an AJS (at least one with learning capacities) may be to employ a set of images pre-evaluated by humans. The task of the AJS is to classify or "to assign an aesthetic value to a series of artworks which were previously evaluated by humans" (Romero et al. 2003).

There are several relevant papers published in the image processing and computer vision research literature that are aimed at the classification of images based on aesthetic evaluation. Most of them employed datasets obtained from photography websites. Some of those datasets are public, so they allow testing of other AJSs. In this section we perform a brief analysis of some of the most prominent works of this type.

Ke et al. (2006) proposed the task of distinguishing between "high quality professional photos" and "low quality snapshots". These categories were created based on users' evaluations of a photo website, so, to some extent, this can be considered as a classification based on aesthetic preference. The website was the dpchallenge.com photography portal, and they used the highest and lowest rated 10 % images from a set of 60,000 in terms of average evaluation. Each photo was rated by at least 100 users. Images with intermediate scores were not considered.

The authors employed a set of high-level image features (such as spatial distribution of edges, colour distribution, blur, hue count) and a support vector machine classification system, obtaining a correct classification rate of 72 %. Using a combination of these metrics with those published by Tong et al. (2004), Ke et al. (2006) achieved a success rate of 76 %.

Luo and Tang (2008) employed the same database. The 12,000 images of the dataset are accessible online[2] allowing the comparison of results. Unfortunately, neither the statistical information of the images (number of evaluations, average score, etc.) nor the images with intermediate ratings are available. The dataset is divided into two sets (training and test), made up of 6,000 images each. The authors state that these sets were randomly created. However, when one reverses the role of the test and training sets (i.e. training with original "test" set and testing with the original "training" set) the results differ significantly. This result indicates that the test and training set are not well-balanced.

Additionally, Luo and Tang (2008) used a blur filter to extract the background and the subject from each photo. Next, they employed a set of features related to clarity contrast (the difference between the crispness of the subject region and the background of the photo), lighting, simplicity, composition and colour harmony. They obtained a 93 % success rate using all features, which clearly improved upon previous results. The "clarity contrast" feature alone yields a success rate above 85 %. The authors pointed out that the difference between those results and the ones obtained by Ke et al. (2006) can be derived from the application of metrics to the image background regions and to the greater adequacy of the metrics itself.

[2] http://137.189.97.48/PhotoqualityEvaluation/download.html.

Datta et al. (2006) employed colour, texture, shape and composition, high-level ad-hoc features and a support vector machine to classify images gathered from a photography portal (photo.net). The dataset included 3581 images. All the images were evaluated by at least two persons. Unfortunately, the statistical information from each image, namely number of votes, value of each vote, etc. is not available. Similarly to previous approaches, they considered two image categories: the highest rated images (average aesthetic value $\geq$5.8, a total of 832 images) and the lowest rated ones ($\leq$4.2, a total of 760 images), according to the ratings given by the users of the portal. Images with intermediate scores were discarded. Datta's justification for making this division is that photographs with an intermediate value "are not likely to have any distinguishing feature, and may merely be representing the noise in the whole peer-rating process" (Datta et al. 2006). The system obtained 70.12 % classification accuracy. The authors published the original dataset of this experiment, allowing future comparisons with other systems.

Wong and Low (2009) employed the same dataset, but selected the 10 % of the highest and lowest rated images. The authors extracted the salient regions of images, with a visual saliency model. They used global metrics related to sharpness, contrast, luminance, texture details, and low depth of field; and features of salient regions based on exposure, sharpness and texture details. Using a support vector machine classifier they obtained a 78 % 5-fold cross-validation accuracy.

In order to create a basis for research on aesthetic classification, Datta et al. (2008) proposed three types of aesthetic classification: aesthetic score prediction; aesthetic class prediction and emotion prediction. All the experiments explained in this section rely on aesthetic class prediction. He also published four datasets: the one employed in Datta et al. (2006), and 3 other extracted from photo.net (16,509 images), dpchallenge.com (14,494 images) and "Terragalleria" (14,494 images).[3] These three datasets include information regarding the number of votes per image and "score" (e.g. number of users that assigned a vote of "2" to image "id454"). Moreover, a dataset is included from the website "Alipr" with 13,100 emotion-tagged images.

Although not within the visual field, it is worth mentioning the work carried out by Manaris et al. (2007) in which a system was trained to distinguish between popular (high number of downloads) and unpopular classical music (low number of downloads). The dataset was obtained from downloads of the website Classical Music Archive (http://www.classicalarchives.com) in November 2003. Two sets, with high and low number of downloads, were created, in a similar way to the previously mentioned works. The "popular" set contained 305 pieces, each one with more than 250 hits, while the "not popular" contained 617 pieces with less than 22 downloads. The system is based on a set of metrics based on Zipf's Law applied to musical concepts such as pitch, duration, harmonic intervals, melodic intervals, harmonic consonance, etc. The classification system is based on an artificial neural network. The success rate was 87.85 % (it classified correctly 810 out of 922 instances), which

[3] Available from http://ritendra.weebly.com/aesthetics-datasets.html.

was considered promising by the authors. The same approach could be applied to images if we use the number of times an image is downloaded or the number of hits of its high-resolution version.

All these works rely on the use of photography and artistic websites. While these sites provides large datasets created by a third party, which should minimise the chances of being biased, the approach has several shortcomings for the purposes of AJS validation.

The experimental environment (participants and methodology) is not as controlled as in a psychological test, and several exogenous factors may influence the image scores. It is not possible to have all the information about the people and the circumstances in which they participated. The personal relations between users may affect their judgement. The same person may cast more than one vote, and so on.

It is also difficult to know what the users are evaluating when they vote. At photo.net the users can classify each image according to its "aesthetic" and "originality", however these scores are highly correlated (Datta et al. 2006), which indicates that users were not differentiating between these criteria. Since the selection of images is not under the control of the researcher, the aesthetic evaluation can be highly influenced by the semantics of content, novelty, originality and so on. These websites include some level of competition (in fact dpchallenge.com is a contest), so the possibilities of some biased votes is even higher.

The interpretation of the results obtained by an AJS in this kind of test is not straightforward. Different datasets have different levels of difficulty. As such, a percentage of correct answers of, e.g. 78 % can be a good or a bad score. As such, the comparison with the state of the art becomes of huge importance. Additionally, it may also be valuable to consider the difficulty of the task for humans. Thus, estimate the discrepancy between the success rate of the AJS and the success rates obtained by humans. Although this is not possible for the previously mentioned datasets, if the dataset includes all the voting information, one can calculate the agreement between humans and the AJSs. In other words, check if the response of the AJS is within the standard deviation for human responses.

For the purposes of AJS validation, the dataset should neither be trivial nor allow shortcuts that enable the system to perform the task exploiting properties of the artefacts which are not related with the task. Teller and Veloso (1996) discovered that their genetic programming approach to face recognition was identifying subjects based on the contents of the background of images (the photographs had been taken in different offices) instead of on the faces. The same type of effect may happen in aesthetic judgement test unless proper measures are taken. For instance, good photographers tend to have good cameras and take good photographs. A system may correctly classify photographs by recognising a good camera (e.g. a high resolution one) instead of recognising the aesthetic properties of the images. Thus, it is necessary to take the appropriate precautions to avoid this type of exploitation (e.g. reducing all the images to a common resolution before they are submitted to the classifier). This precaution has been taken in the works mentioned in Sect. 11.3 of this chapter. Nevertheless, it is almost impossible to ensure that the judgements are made exclusively on aesthetic properties.

For all the above reasons, the use of several datasets and types of tasks during the validation can help assessing the consistency and coherence of the results.

Creating datasets specifically for the purposes of the validation of AJSs is also valuable. An option is to create a dataset made up of images evaluated by humans in a controlled environment, following, for instance, a methodology similar to the one employed by Nadal (2007). We are not aware of any AJS evaluated like this in the field of visual art. In the musical field, there is a system that follows this approach (Manaris et al. 2005), in which a classifier is trained from human responses to musical pieces in a controlled experiment. A system similar to the one previously described achieved an average success rate of over 97 % in predicting (within one standard deviation) human emotional responses to those pieces (Manaris et al. 2007). Another option would be to create datasets that focus on a specific aesthetic property. For instance, to judge the balance of the composition one could ask photographers to take several pairs of photographs of the same motif, with the same camera, exposure, lighting conditions, etc. but with different framings so that one is a well-balanced composition and the other is not, according to the views of the photographers. This would allow the elimination of several of the external factors that could bias the judgement and would also allow an incremental development of the AJSs by focusing on one property at a time, and then moving towards tasks that require taking several aesthetic properties into consideration.

11.2.3 Style and Author Classification

In order to provide objective testing and to further analyse the abilities of AJSs, we explore validation approaches which test the ability of the system to learn the characteristics of a visual style (from an author, a trend, etc.). This type of test is not directly related with aesthetic value, but it can support AJS development.

In the field of computational creativity, a style-based classifier could allow the creation of image generation systems that produce images of a given artistic style and, perhaps more importantly in that context, it could be used to create images that are stylistically different from a given style or styles.

An objective way of performing this kind of test is employing artworks from several authors. The problems with this method usually arise from: (i) the relatively “low” production of most artists, since a machine learning approach can easily require hundreds or even thousands of examples; (ii) the heterogeneity of the artistic production of the authors, caused by the exploration of different styles, differences between early and mature works, etc. One can partially overcome these difficulties by selecting authors with vast productivity and by choosing the most prototypical works. Unfortunately, this may rule out the possibility of using several influential artists and bias the results by making the task easier than what would be desirable.

Another approach consists of classifying artworks according to the artistic “style”. The main difficulties to overcome when setting up this type of experiment are: (i) the images must be previously, and correctly, classified as belonging to a particular style; (ii) one must ensure that there is no overlap between styles; (iii) one cannot use exclusively the most representative images of each style, otherwise the tasks may become trivial and, therefore, useless.

The first problem can be partially solved by using a relevant external source for the images. Unfortunately, the only published digital sets of artistic images we are aware of are those provided by Directmedia/The Yorck Project publications. However, the quality of the collections is far from perfect (they include black and white versions of some images, frames, detailed images of parts of other artworks, etc.). One can also resort to online databases of paintings. The collection "Oil paintings by Western masters" contains 46,000 images and can be found in the peer-to-peer network. The *Worldimages* website (http://worldimages.sjsu.edu/kiosk/artstyles.htm), the website http://www.zeno.org, developed by the creators of "The Yorck Project", and online museum websites are also good sources of images.

Wallraven et al. (2008) analysed the perceptual foundations of the traditional categorisation of images into art styles, finding supporting evidence. They concluded that style identification was predominantly a vision problem and not merely a historical or cultural artefact.

Wallraven et al. (2009) presented an experiment that analysed the capacity of a group of non-experts in art to categorise a set of artworks in styles. One of the metrics they analysed is the artist consistency, which was higher if paintings of the same painter were put in the same cluster. In one experiment, they obtained an average artist consistency of 0.65. The conclusions were that "experts were able to reliably group unfamiliar paintings of many artists into meaningful categories". In the same paper, the authors employed a set of low-level measures (Fourier analysis, colour features, Gist, etc.) and a k-means algorithm to categorise the artworks into styles. They concluded that low-level features were not adequate to artistic style classification: "the fact that neither texture, nor colour-based, scale-sensitive or complexity measures correlate at any dimension casts doubt on whether another [low level] measure will do much better" (Wallraven et al. 2008).

Marchenko et al. (2005), based on the colour theory of Itten (1973), characterised regions of the image in terms of "artistic colour concepts", while Yan and Jin (2005) used several colour spaces to gather information with the aim of retrieving and classifying oil paintings.

There are several papers in the content-based image retrieval literature that propose image classification based on the "type" of image, distinguishing professional photos from amateur ones, e.g. (Tong et al. 2004); or photos from: (i) paintings (Cutzu et al. 2003), (ii) computer graphics (Athitsos et al. 1997), (iii) computer-generated images (Lyu and Farid 2005). These tasks result in an interesting test field for AJS, creating the opportunity of using AJSs in image classification tasks that are far from aesthetics. These works can also provide tools (e.g., features, classification methods, etc.) of interest to the creative computer community, in particular to those researchers involved in artistic tasks.

11.3 The Evolution of an AJS

This section describes the evolution of an AJS over the course of the past decade. It started as a heuristic based system, it was tested using the DJT, and it subsequently became part of an evolutionary art tool. Prompted by the results obtained, an AJS with learning abilities was developed and tested in a wide variety of experiments, which are also described briefly.

11.3.1 A Heuristic AJS

Machado and Cardoso (1998) took inspiration from the works of Arnheim (1956; 1966; 1969), as well as from the research indicating a preference for simple representations of the world, and a trend to perceive it in terms of regular, symmetric and constant shapes (Wertheimer 1939, Arnheim 1966, Tyler 2002, Field et al. 2000). They explored the working hypothesis that the aesthetic value was linked with the sensorial and intellectual pleasure experienced when finding a compact percept (i.e. internal representation) of a complex visual stimulus (cf. Chap. 12). The identification of symmetry, repetition, rhythm, balance, etc. can be a way of reducing the complexity of the percept, which would explain the universal nature of these aesthetic principles and the ability of the brain to recognise them "effortlessly".

The approach rewards images that are simultaneously visually complex and easy to perceive, employing estimates for the *Complexity of the Percept* (CP) and for the *Complexity of the Visual Stimulus* (CV). An estimate for CV should assess the predictability of the image pixels. JPEG image compression mainly affects the high frequencies, which can normally be discarded without significant loss in image quality. The amount, and quality (i.e. the error involved) of the compression achieved by this method depends on the predictability of the pixels in the image being compressed. Unlike JPEG compression, which only takes into account local information, fractal image compression can take advantage of the self-similarities present in the image. Machado and Cardoso (1998) assume that JPEG compression is less like the way humans perceive images than fractal image compression, and hence use fractal compression as a rough estimate of the CP. CP and CV are estimated through the division of the root mean square error by the compression ratio resulting, respectively, from the fractal (quadratic tree based) and JPEG encoding of the image.

A time component is also considered (Machado and Cardoso 1998; 2002). As time elapses, there is a variation in the detail level of image perception. Therefore, it is necessary to estimate CP for specific points in time, in this case t_0 and t_1, which is achieved by carrying out a fractal image compression with increasing detail levels. The proposed approach values images where CP is stable for different detail levels. The idea being that as time goes by one should be able to acquire additional information about the image, for example: the increase in size of the percept should be balanced out by the increase in its level of detail. It is important to notice that Machado and Cardoso neither suggested that the employed JPEG complexity was able to fully capture the concept of image complexity, nor that the fractal image compression was able to capture the complexity of visual perception. They posited that JPEG was closer to visual complexity than fractal compression, and that fractal compression was closer to processing complexity than JPEG, subsequently testing the possibility of using these measures as rough estimates for these concepts in the context of a specific, and limited, aesthetic theory.

The following formula was proposed as a way to capture the previously-mentioned notions (Machado and Cardoso 1998):

$$\text{aesthetic value} = \frac{CV^a}{(CP(t_1) \times CP(t_0))^b} \times \frac{1}{\left(\frac{CP(t_1)-CP(t_0)}{CP(t_1)}\right)^c} \tag{11.1}$$

where a, b and c, are parameters used to tune the relevance given to each of the components. The left side of the formula rewards those images which have high CV and low CP estimates at the same time, while the right side rewards those images with a stable CP across time. The division by $CP(t_1)$ is a normalisation operation. The formula can be expanded in order to encompass further instants in time, but the limitations of the computational implementation led the authors to use only two instants in their tests.

The images of the DJT were digitalised, converted to greyscale, and resized to a standard dimension of 512×512 pixels, which may involve changes in the aspect ratio. The estimates for CV, $CP(t_1)$ and $CP(t_0)$ were computed for the resulting images. Using these estimates, the outcome of formula (11.1) was calculated for each of the images. For each of the 90 pairs or triads of images comprising the DJT, the system chose the image that yielded a higher value according to formula (11.1).

The percentage of correct answers obtained by the AJS depends on the values of the parameters a, b and c. Considering all combinations of values for these parameters ranging in the $[0.5, 2]$ interval with 0.1 increments, the maximum percentage of correct answers was 73.3 % and the minimum 54.4 %. The average success rate of the system over the considered parametric interval was 64.9 %.

As previously mentioned, the highest average percentage of correct answers in human tests in the DJT reported by Eysenck and Castle (1971) is 64.4 %, and was obtained by subjects that were final year fine art graduates, a value that is surprisingly similar to the average success rate of our system (64.9 %).

Although comparing the performance of the system to the performance of humans is tempting, one should not jump to conclusions! A similar result cannot be interpreted as a similar ability to perform aesthetic judgements. As previously mentioned, humans may follow principles that are not exclusively in aesthetic order to choose images. Moreover, since the test aims at differentiating between humans, it may take for granted principles that are consensual between them, and the AJS would be unable to identify. Finally, the results say nothing regarding the validity of the test itself (a question that is outside the scope of our research). Thus, what can be concluded is that the considered formulae and estimates are able to capture some of the principles required to obtain a result that is statistically different from the one obtained by answering randomly in the DJT.

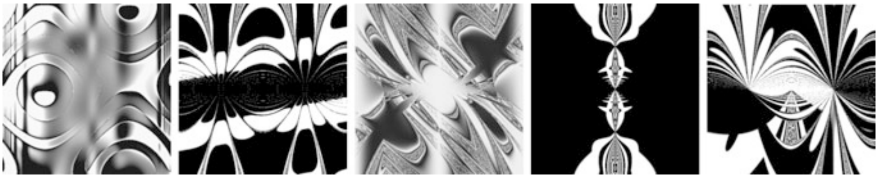


Fig. 11.1 Examples of images created using an Evolutionary Engine and heuristic AJS

Some constraints were applied to the different formula components so as to explore these ideas in an evolutionary context, in the following way:

$$fitness = \frac{\min(\alpha, CV)^a}{\max(\beta, CP(t_1) \times CP(t_0))^b} \times \frac{1}{\max\left(\gamma, \frac{CP(t_1)-CP(t_0)}{CP(t_1)}\right)^c} \tag{11.2}$$

where α, β and γ are constants defined by the user.

These constraints are necessary to ensure that the evolutionary algorithm does not focus exclusively on one of the components of the formula. This could make it converge to images with maximum visual complexity (e.g. white noise images) disregarding entirely the processing complexity estimates, or to images with minimal processing complexity estimates (e.g. pure white). It was not necessary to make additional changes to prevent the situation where $CP(t_1) \simeq 0$ because these images have very low fitness, and are, therefore, already avoided by the evolutionary algorithm.

It is important to notice that the situations where $CP(t_1) \simeq 0$ or $CP(t_1) - CP(t_0) \simeq 0$, although theoretically possible, never occurred when using natural imagery.

Machado and Cardoso (2002) carried out various experiments using a Genetic Programming engine and formula (11.2) as the fitness function.

The results achieved with this autonomous evolutionary art system are quite striking (Machado and Cardoso 2002). In spite of the shortcomings—e.g. it only deals with greyscale images—it allows the evolution of a wide variety of images with different aesthetic merits. Figure 11.1 shows the fittest images from several independent runs.

11.3.2 Learning AJSs

Based on the results described in the previous section, we developed a learning AJS. The system consists of two modules: a *Feature Extractor* (FE) and an *adaptive classifier*.

The FE performs an analysis of the input images by collecting a series of low-level feature values, most of which are related to image complexity. The values that result from the feature extractor are normalised between 1 and −1. These values are the inputs of the classifier, which is made up of a feed-forward artificial neural network with one hidden layer. For training purposes, we resorted to SNNS (*Stuttgart*

Fig. 11.2 Feature extraction steps

Neural Network Simulator, Zell et al. 2003) and standard back-propagation. The results presented in this chapter concern artificial neural networks with one input unit per feature, 12 units in the hidden layer, and 2 units in the output layer (one for each category). A training pattern specifying an output of (1; 0) indicates that the corresponding image belongs to the first set. Likewise, a training pattern with an output of (0; 1) indicates that the corresponding image belongs to the second set. The parameters for the classifier and FE were established empirically in previous experiments.

The experiments presented in this section concern classification tasks of different nature: aesthetic value prediction, author identification and popularity prediction. All the results presented in this section were obtained by the same AJS, trained in different ways. Finally, we describe the integration of this AJS with an evolutionary image generation system.

11.3.2.1 Feature Extraction

In this section we describe the feature extraction process.

The feature extraction can be summarised to the following steps (see Fig. 11.2): (i) *Pre-processing,* which includes all the transformation and normalisation operations applied to a given input image; (ii) *Metrics application*, that is, the application of certain methods based on statistical measurements and image complexity estimates; (iii) *Feature building*, the extraction of results from the metrics applied in order to build the image feature set.

Pre-processing The images from a dataset are individually submitted to a series of transformations before being analysed. A given input image is loaded and resized to a standard width and height of 256×256 pixels, transformed into a three-channel image in the RGB (red, green and blue) colour space, with a depth of 8 bits per channel and all the pixel values are scaled to the $[0, 255]$ interval. This step ensures that all input images share the same format and dimensions.

Next, the image is converted into the HSV (Hue, Saturation and Value) colour space and its HSV channels are split. Each of these channels is stored as a one-channel greyscale image. From here on, we will refer to these images as H, S and V channel images. A new greyscale image is also created by performing a pixel by pixel multiplication of S and V channels and scaling the result to $[0, 255]$. From now on, we will refer to this image as the CS (Colourfulness) channel image.

The images resulting from these operations are subject to transformation operations. The current version of the FE supports seven transformations: no filter,

Table 11.1 Fractal image compression parameters

	Low	Medium	High
Image size	256 × 256 pixels		
Minimum partition level	2	2	3
Maximum partition level	4	5	6
Maximum error per pixel	8	8	8

which means no transformation applied; Sobel-based (Sobel 1990) and Canny-based (Canny 1986) edge detection of horizontal and vertical edges, horizontal edges, vertical edges.

Metrics Application A set of metrics is applied to the images resulting from the pre-processing operations. The FE calculates the following metrics: average (i) and standard deviation (ii) of the image pixel values; complexity estimates based on JPEG (iii) and fractal compression (iv); Zipf Rank-Frequency (v) and Size-Frequency (vi), which result from the application of the Zipf's law (Zipf 1949); (vii) Fractal dimension estimates using the box-counting method (Taylor et al. 1999).

The average (i) and standard deviation (ii) are calculated using the pixel intensity value of each image, except for the H channel image. Since the Hue channel is circular, the average and the standard deviation are calculated based on the norm and angle of Hue values. In addition, a multiplication of the Hue angle value by the CS value is made and consequently a norm is calculated using Hue and CS values.

The image compression schemes used are lossy and so there will be compression errors, i.e. the compressed image will not exactly match the original. All other factors being equal, complex images will tend toward higher compression errors and simple images will tend toward lower compression errors Additionally, complex images will tend to generate larger files than simple ones. Thus, compression error and file size are positively correlated with image complexity.

We consider three levels of detail for the JPEG (iii) and Fractal compression (iv) metrics: low, medium, and high. For each compression level the process is the same, the image is encoded in JPEG and fractal format. In the experiments described herein, we use a quad-tree fractal image compression scheme (Fisher 1995) with the set of parameters given in Table 11.1.

The calculation of the Zipf Rank Frequency (v) metrics implies: counting the number of occurrences of each pixel intensity value in the image; ordering them according to the number of occurrences; tracing a rank vs. number of occurrences plot using a logarithmic scale in both axis; calculating the slope of the trendline and the linear correlation with the trendline.

For the Hue channel, this metrics is calculated in two ways: (i) as described above; (ii) instead of counting the number of occurrences of each Hue value, we add the CS channel values of the corresponding pixels (and divide them by 255 for normalisation purposes). The rationing is that the perceived Hue depends on the saturation and value of the corresponding pixel.

The Zipf Size Frequency (vi) metric is calculated in similar way to Zipf Rank Frequency. For each pixel we calculate the difference between its value and each of its neighbouring pixels. We count the total number occurrences of differences in size 1, size 2, ..., size 255. We trace a size vs. number of occurrences plot using a logarithmic scale in both axes and we calculate the slope and linear correlation of the trendline.

For the H channel we consider a circular distance. The Hue Size Frequency is also calculated using the CS channel. The last metric is a Fractal Dimension estimate (vii) based on the box-counting method. Briefly described: the box-counting method computes the number of cells (boxes) required to cover an object entirely, with grids of cells of varying box size.

Feature Building After the application of the metrics, the results are aggregated to make up the image features.

The average and standard deviation for each channel image returns two values, except for the Hue channel that returns four values for the average and two values for the standard deviation. The JPEG and Fractal compression metrics return three values each, corresponding to the three compression levels considered. Although these metrics are applied to all the images resulting from the pre-processing transformations, the JPEG metric is also applied to the RGB image. As for the Zipf's law based metrics and fractal dimension, the slope of the trendline (m) and the linear correlation (R2) of all greyscale images are extracted. In the case of the Hue channel, these metrics return four values each: two considering only the Hue channel and two considering the Hue and CS channel. We employ a total of 53 metrics applied to seven pre-processing operators, which yield 371 features per image.

11.3.2.2 DJT Experiments

The main goals of these experiments were: (i) confirming the results described in the previous section by the heuristic based AJS and (ii) determining the viability of training an artificial neural network for aesthetic judgement tasks from a small set of examples.

We train an artificial neural network using some of the DJT items and test its ability to predict the correct choice on the remaining ones. The network receives as input the features of two images from the same slide. The output indicates the chosen one. Each of the 82 DJT items that consist of two images yields a "pattern". Eight of the 90 DJT items contain three images instead of two. To deal with these cases, each of these eight items was divided into two "patterns", using the "correct" image in both patterns. Thus, each triad results in two patterns, which yields a total number of 98 patterns (82 obtained from pairs and 16 from triads).

Due to the small number of training patterns we employed a 20-fold cross-validation technique. 20 sets were created from the 98 patterns (18 with 5 patterns and 2 with 4 patterns). In each of the 20 "folds", 19 of the sets were used for training while the remaining one was used for validation.

The sets were generated at random and care was taken to ensure that the two patterns resulting from an item with three images were integrated into the same set. Thus, it was guaranteed that the correct image was not simultaneously used for training and testing the neural network.

Considering the 20 experiments carried out, the global success rate in the test sets was 74.49 %. Which corresponds to a percentage of 71.67 % correct answers in the Design Judgment Test.[4] The result is similar to the maximum success rate previously achieved with the heuristic AJS (73.3 %) by adjusting the parameters. This reinforces the conclusion that it is possible to capture some of the aesthetic principles considered by Maitland Graves in the DJT. They also show that it is possible to learn principles of aesthetic order based on a relatively small set of examples. The fact that the approach was not able to achieve the maximum score in the DJT has two, non exclusive, explanations: (i) the features are unable to capture some of the aesthetic principles required to obtain a maximum score in the DJT; (ii) the set of training examples is not sufficient to allow the correct learning of these principles.

Although the results obtained by the system are higher than the human averages reported in the previously mentioned studies, these results are not comparable. In addition to the issues we mentioned when analysing the results of the heuristic based classifier, the nature of the task is different herein: humans do not make their choices based on a list of correct choices for other items of the test.

11.3.2.3 Author Identification Experiments

In Machado et al. (2004) we presented the results obtained by a previous version of our AJS in an author identification task. The image dataset was made up of 98 paintings from Goya, 153 from Monet, 93 from Gauguin, 122 from Van Gogh, 81 from Kandinsky, and 255 from Picasso. Although the system obtained high success rates (above 90 %), further experiments revealed that the reduced number of images and their nature made the classification task easier than expected.

Taking into account the dataset limitations mentioned in Sect. 11.2.2, we created a dataset composed of images from three prolific painters, from chronologically consecutive artistic movements:

Claude-Oscar Monet (Impressionism, mid 19th century). It consists of 336 images, most of them landscapes and portraits.

Vincent van Gogh (Post-Impressionism, late 19th century): a total number of 1046 well-known images from his work, including landscapes, portraits, self-portraits and still lifes.

Pablo Picasso (Cubism and Surrealism, early 20th century): a total of 540 images belonging to different stages were used, ranging from the Blue Period to the author's surrealist stage.

We avoided using greyscale images and images with insufficient resolution. Some of the images (12 from Picasso and 8 from Van Gogh) included the frames

[4]Some of the test items are triads, hence the lower percentage.

Table 11.2 Success rate in validation set (the results are averages of 50 independent runs)

Picasso vs. Van Gogh	Picasso vs. Monet	Van Gogh vs. Monet
92.1 %	91.5 %	89.9 %

Table 11.3 Confusion matrix (the results are averages of 50 independent runs)

	Picasso	Monet	Van Gogh
Picasso	87.59 %	2.59 %	9.81 %
Monet	4.76 %	70.24 %	25.00 %
Van Gogh	4.11 %	6.60 %	89.29 %

of the painting. Since we avoided doing any sort of manual pre-processing of the images, the frames were not removed. The images were gathered from different sources and the dataset will be made available for research purposes, thus enabling other researchers to compare their results with ours.

The experimental results are averages of 50 independent runs using different training and validation sets. In each run, 90 % of the images were randomly selected to train the artificial neural network. The remaining ones were used as validation set to assess the performance of the artificial neural network. The training of the artificial neural network was stopped after a predetermined number of learning steps. All the results presented concern the performance in validation.

Table 11.2 presents the results obtained in an author classification task with two classes. As it can be observed, discriminating between the works of Van Gogh and Monet was the biggest challenge. Conversely, Pablo Picasso's works were easily distinguished from the ones made by Monet and Van Gogh.

In Table 11.3 we present the confusion matrix for this experiment, which reinforces the previous findings. There is a significant drop in performance when it comes to the correct identification of Claude-Oscar Monet's works. The existence of fewer paintings of this author can explain the difficulties encountered in correctly learning how to recognise his style. A more detailed analysis of this experiment is currently in preparation.

Overall, the results indicate that the considered set of metrics and classifier system are able to distinguish between the signatures (in the sense used by Cope 1992) of different authors. It cannot be stated that the AJS is basing its judgement, at least exclusively, on aesthetic principles. It can, however, be stated that it is able to perform stylistic classification in the considered experimental settings. Even if we could demonstrate that the system was following aesthetic principles, this would not ensure that those principles are enough to perform aesthetic value assessments. If the system obtained bad results in distinguishing between works that have different aesthetic properties it would cast serious doubts on its ability to perform aesthetic evaluation. Thus, a good performance on an author identification task does not ensure the ability to perform aesthetic evaluation, but it is arguably a prerequisite.

11.3.2.4 Image Classification Based on Online Evaluation

We used the dataset provided by Datta et al. (2006) that was analysed in Sect. 11.2.2. The database contains 832 images with an aesthetic rating ≥ 5.8 and 760 images with a rating ≤ 4.2. However, when we carried out our experiment, some of the images used by Datta were no longer available at photo.net, which means that our image set is slightly smaller. We were able to download 656 images with a rating of 4.2 or less, and 757 images with a rating of 5.8 or more.

We conducted 50 runs, each with different training and validation sets, randomly created with 80 % and 20 % of the images, respectively. The success rate in the validation set was 77.22 %, which was higher than the ones reported in the original paper (Datta et al. 2006) but lower than the one obtained by Wong and Low (2009), using 10 % of the images in each set.

11.3.2.5 Integration in an Image Generation System

A previous version of the AJS described here was used in conjunction with a genetic programming evolutionary art tool. The main goal of this experiment, reported by Machado et al. (2007), was to develop an approach that promoted stylistic change from one evolutionary run to the next. The AJS assigns fitness to the evolved images, guiding the evolutionary engine.

The AJS is trained by exposing it to a set of positive examples made up of artworks of famous artists, and to a set of negative examples made up of images generated randomly by the system. The goal is twofold: (i) evolving images that relate with the aesthetic reference provided by the positive examples, which can be considered an inspiring set; (ii) evolving images that are novel relative to the imagery typically produced by the system. Thus, more than trying to replicate a given style, the goal is to break from the traditional style of the evolutionary art tool. Once novel imagery is found (i.e. when the evolutionary engine is able to find images that the AJS fails to classify as being created by it), these images are added to the negative set of examples, the AJS is re-trained and a new evolutionary run begins. This process is iteratively repeated and, by this means, a permanent search for novelty and deviation from the previously explored paths is enforced.

Next, the genetic programming engine and the AJS performed 11 consecutive iterations (Machado et al. 2007). In each iteration, the evolutionary engine was able to find images that were misclassified by the AJS. Adding this set of examples to the dataset forced the AJS to find new ways to discriminate between paintings and the images created by the evolutionary art tool. The evolutionary engine and the AJS performed well across all iterations. The success rate of the AJS for validation set images was above 98 % in all iterations. The evolutionary engine was also always able to find novel styles that provoked misclassification errors. In Fig. 11.3 we present some examples of images created in the 1st and 11th iteration.

Overall, the results indicate that the internal coherency of each run is high, in the sense that runs converge to imagery of a distinctive and uniform style. The style

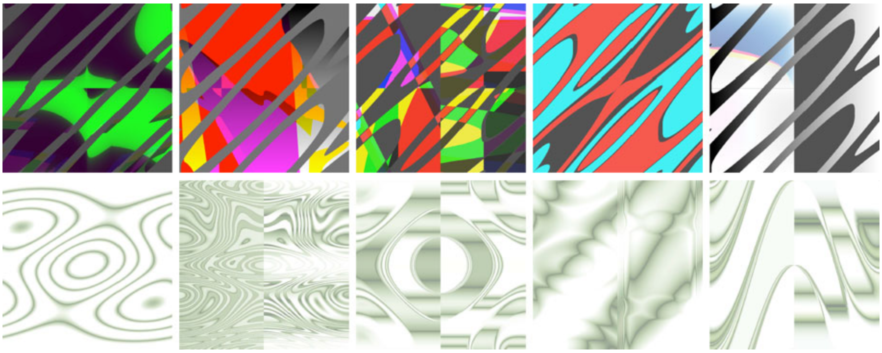


Fig. 11.3 Examples of images created using an Evolutionary Engine and an adaptive AJS in the 1st (*upper row*) and 11th (*lower row*) iteration of the experiment

Table 11.4 Percentage of images classified as external by the ANNs used to guide evolution in iterations 1 and 11, and the difference between them

Set	Iteration 1	Iteration 11	Difference
Painting masterpieces	99.68 %	96.88 %	−2.80 %
User-guided evolution	17.99 %	10.07 %	−7.91 %

differences between runs are also clear, indicating the ability of the approach to promote a search for novelty. They also indicate that the aesthetic reference provided by the external set manages to fulfil its goal, making it possible for AJSs to differentiate between those images that may be classified as paintings and those generated by the GP system (Machado et al. 2007).

A set of experiments was carried out to compare the performance of the AJS from the 1st and 11th iteration, using datasets made up of images that were not employed in the runs. The experimental results are presented in Table 11.4 and show that the AJS of the 11th generation performs worse than the one of the 1st iteration at classifying external imagery (a difference of 2.8 %), and better at classifying evolution generated images (a difference of 7.91 %). These results suggest that the iterations performed with the evolutionary engine promote the generalisation abilities of the AJS, leading to an overall improvement in classification performance.

The integration of an AJS within a bootstrapping evolutionary system of this kind is extremely valuable. As the results indicate, it allows the generation of images that explore the potential weaknesses of the classifier system and the subsequent use of these images as training instances, leading to an overall increase in performance. Additionally, if the evolutionary system is able to generate images that the AJS is unable to classify correctly (even after re-training it) and that a human can classify, it shows that the set of features is not sufficient for the task at hand. Additionally, it gives indications about the type of analysis that should be added in order to improve the performance of the AJS.

11.4 Conclusions

The development of AJS presents numerous difficulties, and there are still several open questions, validation being one of them.

This chapter proposed several ways of testing and comparing the results of aesthetic judgement systems. We proposed validation tasks based on psychological tests, on style and author identification, on users' preferences, and on popularity prediction.

Some alternatives for AJS design have been briefly explored. We focus on an adaptive architecture based on a series of metrics and a machine learning classifier. This type of approach was employed in the field of computational creativity and is popular in content based image retrieval and computer vision research. Some of the works in these areas that can be valuable to computational creativity are analysed. The datasets and results they obtained are presented to serve as a reference for future comparison.

We also presented a heuristic based AJS and discussed the results obtained by the system in a psychological test designed for humans. The experiments show that this AJS was able to capture some of the aesthetic principles explored in the test. The integration of the heuristic AJS with an image generation system was also described and the results briefly discussed.

Subsequently, we described the development of an adaptive AJS based on complexity metrics and an artificial neural network classifier, and presented the experimental results obtained by this AJS in several validation tasks.

The results attained in the psychological test show that the system is able to learn from a set of examples made up of items of the test, obtaining a success rate above 70 % in a cross validation experiment. This result is similar to the one obtained by the heuristic based AJS, indicating that the system is able to reverse engineer some of the aesthetic principles considered in the DJT.

The author identification tasks show that, in the considered experimental settings, the system is able to perform classification based on the image style with an average success rate above 90 % in binary classification. The results obtained by our system in the prediction of users' aesthetic evaluation of online photographs are comparable with those reported as state of the art.

Finally, we presented the integration of the learning AJS with an image generation engine to build a system designed to promote a constant search for novelty and stylistic change.

Submitting the same AJS to several validation tasks allows one to overcome, at least partially, the shortcomings of individual tasks and to get additional insight on the weaknesses and strengths of the AJS.

We consider that the adoption of common validation procedures is an important step towards the development of the field. Sharing datasets allows other researchers to assess the strengths and weaknesses of their systems relative to published work. Sharing the training and test patterns used in experiments further promotes this collaboration between research teams, since it enables assessment of performance improvement that can be expected by the inclusion of the metrics used by other researchers in one's own AJS. Once these performance improvements are identified, the logical next step is the development, through collaboration, of AJSs that encompass the metrics used by the different research groups. These could lead, for instance, to an international research project where several research groups build a common AJS. Some of the groups could propose metrics, others design the classifier, and so on. Using the validation approaches proposed in this chapter (and future research in this area) it becomes possible to validate the classifier and compare the results with previous approaches. Moreover, due to the numerical nature of the validation approach, it is possible to identify relevant metrics in the classifier for the tasks considered.

AJSs can be valuable for real life applications, including:

- **Image Classification**—e.g., discriminating between professional and amateur photos, paintings and photos, images that are interesting to a particular user, etc.
- **Image Search Engines**—which could take into account user preference, or stylistic similarity to a reference image or images.
- **Online Shopping**—the ability to recognise the aesthetic taste of the user could be explored to propose products or even to guide product design and development.

The development of AJSs can also play an important role in the study of aesthetics, in the sense that the ability to capture aesthetic preferences of individuals and groups may promote a better understanding of the phenomena influencing aesthetic preferences, including cultural differences, training, education, trends, etc.

More importantly, the creation of systems able to perform aesthetic judgements may prove vital for the development of computational creativity systems. For instance, the development of an AJS that closely matches the aesthetic preferences of an individual would open a wide range of creative opportunities. One could use such an AJS in conjunction with an image generation system to create custom made "artificial artists" that would be able to create artworks which specifically address the aesthetic needs of a particular person. These systems could change through time, accompanying the development of the aesthetic preferences of the individual and promoting this development. They could also be shared between people as a way of conveying personal aesthetics, or could be trained to match the aesthetic preferences of a community in order to capture commonality. These are vital steps to accomplish our long term goal and dream: the development of computational systems able to create and feel their art and music.

Acknowledgements The authors would like to thank the anonymous reviewers for their constructive comments, suggestions and criticisms. This research is partially funded by: the Spanish Ministry for Science and Technology, research project TIN2008-06562/TIN; the Portuguese Foundation for Science and Technology, research project PTDC/EIA-EIA/115667/2009; Xunta de Galicia, research project XUGA-PGIDIT10TIC105008-PR.

References

Arnheim, R. (1956). *Art and visual perception, a psychology of the creative eye*. London: Faber and Faber.

Arnheim, R. (1966). *Towards a psychology of art/entropy and art—an essay on disorder and order*. The Regents of the University of California.

Arnheim, R. (1969). *Visual thinking*. Berkeley: University of California Press.

Athitsos, V., Swain, M. J., & Frankel, C. (1997). Distinguishing photographs and graphics on the world wide web. In *Proceedings of the 1997 workshop on content-based access of image and video libraries (CBAIVL '97), CAIVL '97* (pp. 10–17). Washington: IEEE Computer Society. http://portal.acm.org/citation.cfm?id=523204.791698.

Baluja, S., Pomerlau, D., & Todd, J. (1994). Towards automated artificial evolution for computer-generated images. *Connection Science*, *6*(2), 325–354.

Boden, M. A. (1990). *The creative mind: myths and mechanisms*. New York: Basic Books.

Burt, C. (1933). The psychology of art. In *How the mind works*. London: Allen and Unwin.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *8*(6), 679–698.

Chamorro-Premuzic, T., & Furnham, A. (2004). Art judgement: a measure related to both personality and intelligence? *Imagination, Cognition and Personality*, *24*, 3–25.

Cope, D. (1992). On the algorithmic representation of musical style. In O. Laske (Ed.), *Understanding music with AI: perspectives on music cognition* (pp. 354–363). Cambridge: MIT Press.

Cutzu, F., Hammoud, R. I., & Leykin, A. (2003). Estimating the photorealism of images: distinguishing paintings from photographs. In *CVPR (2)* (pp. 305–312). Washington: IEEE Computer Society.

Datta, R., Joshi, D., Li, J., & Wang, J. Z. (2006). Studying aesthetics in photographic images using a computational approach. In *Lecture notes in computer science*. *Computer vision—ECCV 2006, 9th European conference on computer vision, part III*, Graz, Austria (pp. 288–301). Berlin: Springer.

Datta, R., Joshi, D., Li, J., & Wang, J. Z. (2008). Image retrieval: ideas, influences, and trends of the new age. *ACM Computing Surveys*, *40*, 5:1–5:60. http://doi.acm.org/10.1145/1348246.1348248.

Dorin, A., & Korb, K. B. (2009). Improbable creativity. In M. Boden, M. D'Inverno, & J. McCormack (Eds.), *Dagstuhl seminar proceedings: Vol. 09291*. *Computational creativity: an interdisciplinary approach*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany. http://drops.dagstuhl.de/opus/volltexte/2009/2214.

Eysenck, H. (1969). Factor analytic study of the Maitland Graves Design Judgement Test. *Perceptual and Motor Skills*, *24*, 13–14.

Eysenck, H. J. (1983). A new measure of 'good taste' in visual art. *Leonardo, Special Issue: Psychology and the Arts*, *16*(3), 229–231. http://www.jstor.org/stable/1574921.

Eysenck, H. J., & Castle, M. (1971). Comparative study of artists and nonartists on the Maitland Graves Design Judgment Test. *Journal of Applied Psychology*, *55*(4), 389–392.

Eysenck, H. J., Götz, K. O., Long, H. Y., Nias, D. K. B., & Ross, M. (1984). A new visual aesthetic sensitivity test—IV. Cross-cultural comparisons between a Chinese sample from Singapore and an English sample. *Personality and Individual Differences*, *5*(5), 599–600. http://www.sciencedirect.com/science/article/B6V9F-45WYSPS-1M/2/1b43c2e7ad32ef89313f193d3358b441.

Field, D. J., Hayes, A., & Hess, R. F. (2000). The roles of polarity and symmetry in the perceptual grouping of contour fragments. *Spatial Vision*, *13*(1), 51–66.

Fisher, Y. (Ed.) (1995). *Fractal image compression: theory and application*. London: Springer.

Frois, J., & Eysenck, H. J. (1995). The visual aesthetic sensitivity test applied to Portuguese children and fine arts students. *Creativity Research Journal*, *8*(3), 277–284. http://www.leaonline.com/doi/abs/10.1207/s15326934crj0803_6.

Furnham, A., & Walker, J. (2001). The influence of personality traits, previous experience of art, and demographic variables on artistic preference. *Personality and Individual Differences*, *31*(6), 997–1017. http://www.sciencedirect.com/science/article/B6V9F-440BD9B-J/2/c107a7e1db8199da25fb754780a7d220.

Götz, K. (1985). *VAST: visual aesthetic sensitivity test*. Dusseldorf: Concept Verlag.

Götz, K. O., & Götz, K. (1974). The Maitland Graves Design Judgement Test judged by 22 experts. *Perceptual and Motor Skills*, *39*, 261–262.

Graves, M. (1946). *Design judgement test*. New York: The Psychological Corporation.

Graves, M. (1948). *Design judgement test, manual*. New York: The Psychological Corporation.

Graves, M. (1951). *The art of color and design*. New York: McGraw-Hill.

Itten, J. (1973). *The art of color: the subjective experience and objective rationale of color*. New York: Wiley.

Iwawaki, S., Eysenck, H. J., & Götz, K. O. (1979). A new visual aesthetic sensitivity test (vast): II. Cross cultural comparison between England and Japan. *Perceptual and Motor Skills*, *49*(3), 859–862. http://www.biomedsearch.com/nih/new-Visual-Aesthetic-Sensitivity-Test/530787.html.

Ke, Y., Tang, X., & Jing, F. (2006). The design of high-level features for photo quality assessment. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference*, *1*, 419–426.

Kowaliw, T., Dorin, A., & McCormack, J. (2009). An empirical exploration of a definition of creative novelty for generative art. In K. B. Korb, M. Randall & T. Hendtlass (Eds.), *Lecture notes in computer science: Vol. 5865*. *ACAL* (pp. 1–10). Berlin: Springer.

Luo, Y., & Tang, X. (2008). Photo and video quality evaluation: focusing on the subject. In D. A. Forsyth, P. H. S. Torr & A. Zisserman (Eds.), *Lecture notes in computer science: Vol. 5304*. *ECCV (3)* (pp. 386–399). Berlin: Springer.

Lyu, S., & Farid, H. (2005). How realistic is photorealistic? *IEEE Transactions on Signal Processing*, *53*(2), 845–850.

Machado, P., & Cardoso, A. (1998). Computing aesthetics. In F. Oliveira (Ed.), *Lecture notes in computer science: Vol. 1515*. *Proceedings of the XIVth Brazilian symposium on artificial intelligence: advances in artificial intelligence*, Porto Alegre, Brazil (pp. 219–229). Berlin: Springer.

Machado, P., & Cardoso, A. (2002). All the truth about NEvAr. *Applied Intelligence, Special Issue on Creative Systems*, *16*(2), 101–119.

Machado, P., Romero, J., & Manaris, B. (2007). Experiments in computational aesthetics: an iterative approach to stylistic change in evolutionary art. In J. Romero & P. Machado (Eds.), *The art of artificial evolution: a handbook on evolutionary art and music* (pp. 381–415). Berlin: Springer.

Machado, P., Romero, J., Manaris, B., Santos, A., & Cardoso, A. (2003). Power to the critics—a framework for the development of artificial art critics. In *IJCAI 2003 workshop on creative systems*, Acapulco, Mexico.

Machado, P., Romero, J., Santos, A., Cardoso, A., & Manaris, B. (2004). Adaptive critics for evolutionary artists. In R. Günther et al. (Eds.), *Lecture notes in computer science: Vol. 3005*. *Applications of evolutionary computing, EvoWorkshops 2004: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC*, Coimbra, Portugal (pp. 435–444). Berlin: Springer.

Manaris, B., Romero, J., Machado, P., Krehbiel, D., Hirzel, T., Pharr, W., & Davis, R. (2005). Zipf's law, music classification and aesthetics. *Computer Music Journal*, *29*(1), 55–69.

Manaris, B., Roos, P., Machado, P., Krehbiel, D., Pellicoro, L., & Romero, J. (2007). A corpus-based hybrid approach to music analysis and composition. In *Proceedings of the 22nd conference on artificial intelligence (AAAI 07)*, Vancouver, BC.

Marchenko, Y., Chua, T.-S., & Aristarkhova, I. (2005). Analysis and retrieval of paintings using artistic color concepts. In *ICME* (pp. 1246–1249). New York: IEEE Press.

Nadal, M. (2007). *Complexity and aesthetic preference for diverse visual stimuli*. PhD thesis, Departament de Psicologia, Universitat de les Illes Balears.

Neufeld, C., Ross, B., & Ralph, W. (2007). The evolution of artistic filters. In J. Romero & P. Machado (Eds.), *The art of artificial evolution*. Berlin: Springer.

Rigau, J., Feixas, M., & Sbert, M. (2008). Informational dialogue with Van Gogh's paintings. In *Eurographics symposium on computational aesthetics in graphics, visualization and imaging* (pp. 115–122).

Romero, J., Machado, P., Santos, A., & Cardoso, A. (2003). On the development of critics in evolutionary computation artists. In R. Günther et al. (Eds.), *Lecture notes in computer science:*

Vol. 2611. Applications of evolutionary computing, EvoWorkshops 2003: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC, Essex, UK. Berlin: Springer.

Saunders, R. (2001). *Curious design agents and artificial creativity—a synthetic approach to the study of creative behaviour*. PhD thesis, University of Sydney, Department of Architectural and Design Science Faculty of Architecture, Sydney, Australia.

Savarese, J. M., & Miller, R. (1979). Artistic preferences and cognitive-perceptual style. *Studies in Art Education*, *20*, 41–45.

Schmidhuber, J. (1997). Low-complexity art. *Leonardo, Journal of the International Society for the Arts, Sciences, and Technology*, *30*(2), 97–103. http://www.jstor.org/stable/1576418.

Schmidhuber, J. (1998). Facial beauty and fractal geometry. http://cogprints.org/690/.

Schmidhuber, J. (2007). Simple algorithmic principles of discovery, subjective beauty, selective attention, curiosity and creativity. In M. Hutter, R. A. Servedio & E. Takimoto (Eds.), *Lecture notes in computer science: Vol. 4754. ALT* (pp. 32–33). Berlin: Springer.

Sobel, I. (1990). An isotropic 3×3 image gradient operator. In *Machine vision for three-dimensional scenes* (pp. 376–379).

Spector, L., & Alpern, A. (1994). Criticism, culture, and the automatic generation of artworks. In *Proceedings of twelfth national conference on artificial intelligence* (pp. 3–8). Seattle/Washington: AAAI Press/MIT Press.

Spehar, B., Clifford, C. W. G., Newell, N., & Taylor, R. P. (2003). Universal aesthetic of fractals. *Computers and Graphics*, *27*(5), 813–820.

Staudek, T. (2002). *Exact aesthetics. Object and scene to message*. PhD thesis, Faculty of Informatics, Masaryk University of Brno.

Staudek, T. (2003). Computer-aided aesthetic evaluation of visual patterns. In *ISAMA-BRIDGES conference proceedings*, Granada, Spain (pp. 143–149).

Svangård, N., & Nordin, P. (2004). Automated aesthetic selection of evolutionary art by distance based classification of genomes and phenomes using the universal similarity metric. In R. Günther et al. (Eds.), *Lecture notes in computer science: Vol. 3005. Applications of evolutionary computing, EvoWorkshops 2004: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC*, Coimbra, Portugal (pp. 445–454). Berlin: Springer.

Taylor, R. P., Micolich, A. P., & Jonas, D. (1999). Fractal analysis of Pollock's drip paintings. *Nature*, *399*, 422.

Teller, A., & Veloso, M. (1996). PADO: a new learning architecture for object recognition. In K. Ikeuchi & M. Veloso (Eds.), *Symbolic visual learning* (pp. 81–116). London: Oxford University Press. http://www.cs.cmu.edu/afs/cs/usr/astro/public/papers/PADO.ps.Z.

Tong, H., Li, M., Zhang, H., He, J., & Zhang, C. (2004). Classification of digital photos taken by photographers or home users. In K. Aizawa, Y. Nakamura & S. Satoh (Eds.), *Lecture notes in computer science: Vol. 3331. PCM (1)* (pp. 198–205). Berlin: Springer.

Tyler, C. W. (Ed.) (2002). *Human symmetry perception and its computational analysis*. Hillsdale: Erlbaum.

Uduehi, J. (1995). A cross-cultural assessment of the maitland graves design judgment test using U.S. and Nigerian students. *Visual Arts Research*, *21*(2), 11–18.

Wallraven, C., Cunningham, D. W., & Fleming, R. (2008). Perceptual and computational categories in art. In P. Brown (Ed.), *International symposium on computational aesthetics in graphics, visualization, and imaging* (pp. 131–138). Aire-la-Ville: Eurographics Association. http://computational-aesthetics.org/2008/.

Wallraven, C., Fleming, R. W., Cunningham, D. W., Rigau, J., Feixas, M., & Sbert, M. (2009). Categorizing art: comparing humans and computers. *Computers & Graphics*, *33*(4), 484–495.

Wertheimer, M. (1939). Laws of organization in perceptual forms. In W. D. Ellis (Ed.), *A source book of gestalt psychology* (pp. 71–88). New York: Harcourt Brace.

Wong, L.-K., & Low, K.-L. (2009). Saliency-enhanced image aesthetics class prediction. In *ICIP* (pp. 997–1000). New York: IEEE Press.

Yan, Y., & Jin, J. S. (2005). Indexing and retrieving oil paintings using style information. In S. Bres & R. Laurini (Eds.), *Lecture notes in computer science: Vol. 3736. VISUAL* (pp. 143–152). Berlin: Springer.

Zell, A., Mamier, G., Vogt, M., Mache, N., Hübner, R., Döring, S., Herrmann, K.-U., Soyez, T., Schmalzl, M., Sommer, T., et al. (2003). *SNNS: Stuttgart neural network simulator user manual, version 4.2* (Technical Report 3/92). University of Stuttgart, Stuttgart.

Zipf, G. K. (1949). *Human behaviour and the principle of least effort: an introduction to human ecology*. Reading: Addison-Wesley.

Chapter 12
A Formal Theory of Creativity to Model the Creation of Art

Jürgen Schmidhuber

Abstract According to the Formal Theory of Creativity (1990–2010), a creative agent—one that never stops generating non-trivial, novel, and surprising behaviours and data—must have two learning components: a general reward optimiser or reinforcement learner, and an adaptive encoder of the agent's growing data history (the record of the agent's interaction with its environment). The learning progress of the encoder is the intrinsic reward for the reward optimiser. That is, the latter is motivated to invent *interesting* spatio-temporal patterns that the encoder does not yet know but can easily learn to encode better with little computational effort. To maximise expected reward (in the absence of external reward), the reward optimiser will create more and more-complex behaviours that yield temporarily surprising (but eventually boring) patterns that make the encoder quickly improve. I have argued that this simple principle explains science, art, music and humour. It is possible to rigorously formalise it and implement it on learning machines, thus building artificial robotic scientists and artists equipped with curiosity and creativity. I summarise my work on this topic since 1990, and present a previously unpublished low-complexity artwork computable by a very short program discovered through active search for novel patterns according to the principles of the theory.

12.1 The Basic Idea

Creativity and curiosity are about actively making or finding novel patterns. Columbus was curious about what's in the West, and created a sequence of actions yielding a wealth of previously unknown, surprising, pattern-rich data. Early physicists were curious about how gravity works, and created novel lawful and regular spatio-temporal patterns by inventing experiments such as dropping apples and measuring their accelerations. Babies are curious about what happens if they move their fingers in just this way, creating little experiments leading to initially novel and surprising but eventually predictable sensory inputs. Many artists and composers also combine

J. Schmidhuber (✉)
IDSIA, University of Lugano & SUPSI, Galleria 2, 6928 Manno-Lugano, Switzerland
e-mail: juergen@idsia.ch

J. McCormack, M. d'Inverno (eds.), *Computers and Creativity*,
DOI 10.1007/978-3-642-31727-9_12, © Springer-Verlag Berlin Heidelberg 2012

previously known spatio-temporal objects in non-trivial ways to create novel patterns.

According to the *Formal Theory of Creativity*, in the examples above, people attempt to maximise essentially the same type of objective function or *reward function* at various stages of their lives. Part of the reward is standard external reward as used in many applications of Reinforcement Learning (RL) (Kaelbling et al. 1996), such as positive reward for eating when hungry, or negative reward (pain) for bumping into an obstacle. In addition to that, however, there is the intrinsic reward, or aesthetic reward, or pure fun, which a creative, subjective observer may extract from some self-generated sequence of actions and observations by learning to encode it more efficiently: the fun is proportional to the difference between how many computational resources (storage space and time) he needs to encode the data sequence *before* and *after* learning. A separate RL algorithm maximises expected fun by finding or creating non-random, non-arbitrary data that soon becomes more predictable or compressible in some initially unknown but learnable way, such as novel jokes, songs, dances, paintings, or scientific observations obeying novel, unpublished laws.

In Sect. 12.3 we will formalise the basic principle. In Sect. 12.4 we discuss our previous approximative implementations thereof: concrete examples of artificial creative scientists or artists that learn to create action sequences yielding intrinsic aesthetic rewards independent of human supervision. In Sect. 12.5 we summarise why aesthetic reward can be viewed as the *first derivative* of subjective beauty in the sense of elegance or simplicity. In Sect. 12.6 we describe the creation of a work of *Low-Complexity Art* (Schmidhuber 1997c) computable by a very short program discovered through a search process modelled by the Formal Theory of Creativity. Next, however, we will first discuss relationships to previous ideas on curiosity, creativity, and aesthetic reward.

12.2 Relation to Previous, Less Formal Work

Much of the work on computational creativity described in this book uses reward optimisers that maximise *external reward* given by humans in response to artistic creations of some improving computational pattern generator. This chapter, however, focuses on *unsupervised* creative and curious systems motivated to make novel, aesthetically pleasing patterns generating *intrinsic reward* in proportion to learning progress.

Let us briefly discuss relations to previous ideas in this vein. Two millennia ago, Cicero already called curiosity a "passion for learning". Section 12.3 will formalise this passion such that one can implement it on computers, by mathematically defining reward for the active creation of patterns that allow for compression progress or prediction improvements.

In the 1950s, psychologists revisited the idea of curiosity as the motivation for exploratory behaviour (Berlyne 1950; 1960), emphasising the importance of novelty (Berlyne 1950) and non-homeostatic drives (Harlow et al. 1950). Piaget (1955) explained explorative learning behaviour in children through his informal concepts of assimilation (new inputs are embedded in old schemes—this may be viewed as a type of compression) and accommodation (adapting an old schema to a new input—this may be viewed as a type of compression improvement). Unlike Sect. 12.3, however, these ideas did not provide sufficient formal details to permit the construction of artificial curious agents.

Aesthetic theory is another source of relevant ideas. Why are curious or creative humans somehow intrinsically motivated to observe or make certain novel patterns, such as aesthetically pleasing works of art, even when this seems irrelevant for solving typical frequently recurring problems such as hunger, and even when the action of observation requires a serious effort, such as spending hours to get to the museum? Since the days of Plato and Aristotle, many philosophers have written about aesthetics and taste, trying to explain why some behaviours or objects are more interesting or aesthetically rewarding than others, e.g. Kant (1781), Goodman (1968), Collingwood (1938), Danto (1981), Dutton (2002). However, they did not have or use the mathematical tools necessary to provide formal answers to the questions above. What about more formal theories of aesthetic perception which emerged in the 1930s (Birkhoff 1933) and especially in the 1960s (Moles 1968, Bense 1969, Frank 1964, Nake 1974, Franke 1979)? Some of the previous attempts at explaining aesthetic experience in the context of information theory or complexity theory (Moles 1968, Bense 1969, Frank 1964, Nake 1974, Franke 1979) tried to quantify the intrinsic aesthetic reward through an "*ideal*" ratio between expected and unexpected information conveyed by some aesthetic object (its "*order*" vs its "*complexity*"). The basic idea was that aesthetic objects should neither be too simple nor too complex, as illustrated by the *Wundt curve* (Wundt 1874), which assigns maximal interestingness to data whose complexity is somewhere in between the extremes. Using certain measures based on information theory (Shannon 1948), Bense (1969) argued for an ideal ratio of $1/e \sim 37\ \%$. Generally speaking, however, these approaches were not detailed and formal enough to construct artificial, intrinsically motivated agents with a built-in desire to create aesthetically pleasing works of art.

The Formal Theory of Creativity does not postulate any objective ideal ratio of this kind. Unlike some of the previous works that emphasise the significance of the subjective observer (Frank 1964, Franke 1979, Frank and Franke 2002), its dynamic formal definition of fun reflects the *change* in the number of bits required to encode artistic and other objects, explicitly taking into account the subjective observer's growing knowledge as well as the limitations of its given learning algorithm (or compression *improvement* algorithm). For example, random noise is always novel in the sense that it is unpredictable. But it is not rewarding since it has no pattern. It is not compressible at all; there is no way of learning to encode it better than by storing the raw data. On the other hand, a given pattern may not be novel to a given observer at a given point in his life, because he already perfectly understands it—again there may be no way of learning to encode it even more efficiently. According to the Formal Theory of Creativity, surprise and aesthetic reward are possible only where there is measurable learning progress. The value of an aesthetic experience (the intrinsic reward of a creative or curious maker or observer of art) is not defined by the created or observed object *per se*, but by the algorithmic compression *progress* (or prediction *progress*) of the subjective, learning observer.

While Kant already placed the finite, subjective human observer in the centre of our universe (Kant 1781), the Formal Theory of Creativity formalises some of his ideas, viewing the subjective observer as a parameter: one cannot tell whether something is art without taking into account the individual observer's current state. This is compatible with the musings of Danto who also wrote that one cannot objectively tell whether something is art by simply looking at it (Danto 1981).

To summarise, most previous ideas on the interestingness of aesthetic objects focused on their complexity, but ignored the *change* of subjective complexity through learning. This change, however, is precisely the central ingredient of the Formal Theory of Creativity.

12.3 Formal Details

Skip this section if you are not interested in formal details.

A learning agent's single life consists of discrete cycles or time steps $t = 1, 2, \ldots, T$. The agent's total lifetime T may or may not be known in advance. At any given t the agent receives a real-valued environmental input vector $x(t)$ and executes a real-valued action $y(t)$ which may affect future inputs. At times $t < T$ its goal is to maximise future *utility*

$$u(t) = E_{\mu}\left[\sum_{\tau=t+1}^{T} r(\tau) \,\middle|\, h(\leq t)\right], \tag{12.1}$$

where the reward $r(t)$ is a special real-valued input (vector) at time t, $h(t)$ is the triple $[x(t), y(t), r(t)]$, $h(\leq t)$ is the known history $h(1), h(2), \ldots, h(t)$, and $E_{\mu}(\cdot \mid \cdot)$ denotes the conditional expectation operator with respect to some typically unknown distribution μ from a set $\mathcal{M}$ of possible distributions. Here $\mathcal{M}$ reflects whatever is known about the possible probabilistic reactions of the environment. For example, $\mathcal{M}$ may contain all computable distributions (Solomonoff 1978, Li and Vitányi 1997, Hutter 2005), thus essentially including all environments one could write scientific papers about. There is just one life, so no need for predefined repeatable trials, and the utility function implicitly takes into account the expected remaining lifespan $E_{\mu}(T \mid h(\leq t))$ and thus the possibility to extend the lifespan through actions (Schmidhuber 2009d).

To maximise $u(t)$, the agent may profit from an improving, predictive *model* p of the consequences of its possible interactions with the environment. At any time t $(1 \leq t < T)$, the model $p(t)$ will depend on the observed history $h(\leq t)$. It may be viewed as the current explanation or description of $h(\leq t)$, and may help to predict and increase future rewards (Schmidhuber 1991b). Let $C(p, h)$ denote some given model p's quality or performance evaluated on a history h. Natural performance measures will be discussed below.

To encourage the agent to actively create data leading to easily learnable improvements of p (Schmidhuber 1991a), the reward signal $r(t)$ is split into two scalar real-valued components: $r(t) = g(r_{ext}(t), r_{int}(t))$, where g maps pairs of real values to real values, e.g., $g(a,b) = a + b$. Here $r_{ext}(t)$ denotes traditional *external* reward provided by the environment, such as negative reward for bumping into a wall, or positive reward for reaching some teacher-given goal state. The Formal Theory of Creativity, however, is mostly interested in $r_{int}(t)$, the *intrinsic* reward, which is provided whenever the model's quality improves—for *purely creative* agents $r_{ext}(t) = 0$ for all valid t. Formally, the intrinsic reward for the model's progress (due to some application-dependent model improvement algorithm) between times t and $t+1$ is

$$r_{int}(t+1) = f\big[C\big(p(t), h(\leq t+1)\big), C\big(p(t+1), h(\leq t+1)\big)\big], \tag{12.2}$$

where f maps pairs of real values to real values. Various progress measures are possible; most obvious is $f(a,b) = a - b$. This corresponds to a discrete time version of maximising the first derivative of the model's quality. *Both the old and the new model have to be tested on the same data, namely, the history so far*. That is, progress between times t and $t+1$ is defined based on two models of $h(\leq t+1)$, where the old one is trained only on $h(\leq t)$ and the new one also gets to see $h(t \leq t+1)$. This is like $p(t)$ predicting data of time $t+1$, then observing it, then learning something, then becoming a measurably improved model $p(t+1)$.

The above description of the agent's motivation separates the goal (finding or making data that can be modelled better or faster than before) from the means of achieving the goal. The controller's RL mechanism must figure out how to translate such rewards into action sequences that allow the given world model improvement algorithm to find and exploit previously unknown types of regularities. It must trade off long-term vs short-term intrinsic rewards of this kind, taking into account all costs of action sequences (Schmidhuber 1999; 2006a).

The field of Reinforcement Learning (RL) offers many more or less powerful methods for maximising expected reward as requested above (Kaelbling et al. 1996). Some were used in our earlier implementations of curious, creative systems; see Sect. 12.4 for a more detailed overview of previous simple artificial scientists and artists (1990–2002). Universal RL methods (Hutter 2005, Schmidhuber 2009d) as well as RNN-based RL (Schmidhuber 1991b) and SSA-based RL (Schmidhuber 2002a) can in principle learn useful internal states memorising relevant previous events; less powerful RL methods (Schmidhuber 1991a, Storck et al. 1995) cannot.

In theory $C(p, h(\leq t))$ should take the entire history of actions and perceptions into account (Schmidhuber 2006a), like the performance measure C_{xry}:

$$C_{xry}\big(p, h(\leq t)\big) = \sum_{\tau=1}^{t} \big\|pred\big(p, x(\tau)\big) - x(\tau)\big\|^2 + \big\|pred\big(p, r(\tau)\big) - r(\tau)\big\|^2 + \big\|pred\big(p, y(\tau)\big) - y(\tau)\big\|^2 \tag{12.3}$$

where $pred(p, q)$ is p's prediction of event q from earlier parts of the history.

C_{xry} ignores the danger of overfitting (too many parameters for few data) through a p that stores the entire history without compactly representing its regularities, if any. The principles of *Minimum Description Length* (MDL) and closely related *Minimum Message Length* (MML) (Kolmogorov 1965, Wallace and Boulton 1968, Wallace and Freeman 1987, Solomonoff 1978, Rissanen 1978, Li and Vitányi 1997), however, take into account the description size of p, viewing p as a compressor program of the data $h(\leq t)$. This program p should be able to deal with any prefix of the growing history, computing an output starting with $h(\leq t)$ for any time t $(1 \leq t < T)$. (A program that halts after t steps can temporarily be fixed or augmented by the trivial non-compressive method that simply stores any raw additional data coming in after the halt—later learning may yield better compression and thus intrinsic rewards.)

$C_l(p, h(\leq t))$ denotes p's compression performance on $h(\leq t)$: the number of bits needed to specify the predictor and the deviations of the sensory history from its predictions, in the sense of loss-free compression. The smaller C_l, the more lawfulness and regularity in the observations so far. While random noise is irregular and arbitrary and incompressible, most videos are regular as most single frames are very similar to the previous one. By encoding only the deviations, movie compression algorithms can save lots of storage space. Complex-looking fractal images (Mandelbrot 1982) are regular, as they usually are similar to their details, being computable by very short programs that re-use the same code over and over again for different image parts. The universe itself seems highly regular, as if computed by a program (Zuse 1969, Schmidhuber 1997a; 2002c; 2006b; 2007a): every photon behaves the same way; gravity is the same on Jupiter and Mars, mountains usually don't move overnight but remain where they are, etc.

Suppose p uses a small predictor that correctly predicts many $x(\tau)$ for $1 \leq \tau \leq t$. This can be used to encode $x(\leq t)$ compactly: Given the predictor, only the wrongly predicted $x(\tau)$ plus information about the corresponding time steps τ are necessary to reconstruct $x(\leq t)$, e.g., (Schmidhuber 1992). Similarly, a predictor that learns a probability distribution on the possible next events, given previous events, can be used to compactly encode observations with high (respectively low) predicted probability by few (respectively many) bits (Huffman 1952, Schmidhuber and Heil 1996), thus achieving a compressed history representation.

Alternatively, p could make use of a 3D world model or simulation. The corresponding MDL-based quality measure $C_{3D}(p, h(\leq t))$ is the number of bits needed to specify all polygons and surface textures in the 3D simulation, plus the number of bits needed to encode deviations of $h(\leq t)$ from the simulation's predictions. Improving the model by adding or removing polygons may reduce the total number of bits required (Schmidhuber 2010).

The ultimate limit for $C_l(p, h(\leq t))$ is $K^*(h(\leq t))$, a variant of the Kolmogorov complexity of $h(\leq t)$, namely, the length of the shortest program (for the given hardware) that computes an output starting with $h(\leq t)$ (Solomonoff 1978, Kolmogorov 1965, Li and Vitányi 1997, Schmidhuber 2002b). We do not have to worry about the fact that $K^*(h(\leq t))$ in general cannot be computed exactly, only approximated from above (for most practical predictors the approximation will be crude). This just means that some patterns will be hard to detect by the limited predictor of choice, that is, the reward maximiser will get discouraged from spending too much effort on creating those patterns.

$C_l(p, h(\leq t))$ does not take into account the time $\tau(p, h(\leq t))$ spent by p on computing $h(\leq t)$. A runtime-dependent quality measure inspired by optimal universal search (Levin 1973) is

$$C_{l\tau}\big(p, h(\leq t)\big) = C_l\big(p, h(\leq t)\big) + \log \tau\big(p, h(\leq t)\big). \tag{12.4}$$

Here additional compression by one bit is worth as much as runtime reduction by a factor of $\frac{1}{2}$. From an asymptotic optimality-oriented point of view this is a best way of trading off storage and computation time (Levin 1973, Schmidhuber 2004).

In practical applications (Sect. 12.4) the compressor/predictor of the continually growing data typically will have to calculate its output online, that is, it will be able to use only a constant number of computational instructions per second to predict/compress new data. The goal of the typically slower learning algorithm must then be to improve the compressor such that it keeps operating online within those time limits, while compressing/predicting better than before. The costs of computing $C_{xry}(p, h(\leq t))$ and $C_l(p, h(\leq t))$ and similar performance measures are linear in t, assuming p consumes equal amounts of computation time for each prediction. Hence online evaluations of learning progress on the full history so far generally cannot take place as frequently as the continually ongoing online predictions.

Some of the learning and its progress evaluations may take place during occasional "sleep" phases (Schmidhuber 2006a). But previous practical implementations have looked only at parts of the history for efficiency reasons: The systems mentioned in Sect. 12.4 used online settings (one prediction per time step, and constant computational effort per prediction), non-universal adaptive compressors or predictors, and approximative evaluations of learning progress, each consuming only constant time despite the continual growth of the history.

12.3.1 Continuous Time Formulation

In continuous time, $O(t)$ denotes the state of subjective observer O at time t. The subjective compressibility (simplicity or regularity) $B(D, O(t))$ of a sequence of observations and/or actions is the negative number of bits required to encode D, given $O(t)$'s current limited prior knowledge and limited compression/prediction method. The time-dependent and observer-dependent subjective *interestingness* or *surprise* or *aesthetic value*, $I(D, O(t))$ is

$$I\big(D, O(t)\big) \sim \frac{\partial B(D, O(t))}{\partial t}, \tag{12.5}$$

the *first derivative* of subjective simplicity: as O improves its compression algorithm, formerly apparently random data parts become subjectively more regular and beautiful, requiring fewer bits for their encoding.

There are at least two ways of having "fun": execute a learning algorithm that improves the compression of the already known data (in online settings, without increasing computational needs of the compressor/predictor), or execute actions that generate more data, then learn to better compress or explain this new data.

12.4 Previous Approximative Implementations of the Theory

Since 1990 I have built simple artificial scientists or artists with an intrinsic desire to build a better model of the world and what can be done in it. They embody approximations of the theory of Sect. 12.3. The agents are motivated to continually improve their models, by creating or discovering more *surprising, novel patterns*, that is, data predictable or compressible in hitherto unknown ways. They actively invent experiments (algorithmic protocols or programs or action sequences) to explore their environment, always trying to learn new behaviours (policies) exhibiting previously unknown regularities or patterns. Crucial ingredients are:

1. An adaptive world model, essentially a predictor or compressor of the continually growing history of actions and sensory inputs, reflecting current knowledge about the world,
2. A learning algorithm that continually improves the model (detecting novel, initially surprising spatio-temporal patterns, including works of art, that subsequently become known patterns),
3. Intrinsic rewards measuring the model's improvements due to its learning algorithm (thus measuring the *degree* of subjective novelty & surprise),
4. A separate reward optimiser or reinforcement learner, which translates those rewards into action sequences or behaviours expected to optimise future reward.

These ingredients make the agents curious and creative: they get intrinsically motivated to acquire skills leading to a better model of the possible interactions with the world, discovering additional "eye-opening" novel patterns (including works of art) predictable or compressible in previously unknown ways.

Ignoring issues of computation time, it is possible to devise mathematically optimal, *universal* RL methods (Hutter 2005, Schmidhuber 2009d) for such systems (Schmidhuber 2006a; 2010) (2006-). However, previous practical implementations (Schmidhuber 1991a, Storck et al. 1995, Schmidhuber 2002a) were non-universal and made approximative assumptions. Among the many ways of combining methods for (1-4) we implemented the following variants:

A. Non-traditional RL based on adaptive recurrent neural networks as predictive world models is used to maximise intrinsic reward created in proportion to prediction error (Schmidhuber 1991b).

B. Traditional RL (Kaelbling et al. 1996) is used to maximise intrinsic reward created in proportion to improvements of prediction error (Schmidhuber 1991a).

C. Traditional RL maximises intrinsic reward created in proportion to relative entropies between the agent's priors and posteriors (Storck et al. 1995).

D. Non-traditional RL (Schmidhuber et al. 1997) (without restrictive Markovian assumptions) learns probabilistic, hierarchical programs and skills through zero-sum intrinsic reward games of two players, each trying to out-predict or surprise the other, taking into account the computational costs of learning, and learning *when* to learn and *what* to learn (1997–2002) (Schmidhuber 1999; 2002a).

Variants B, C & D also showed experimentally that intrinsic rewards can substantially accelerate goal-directed learning and *external* reward intake of agents living in environments providing external reward for achieving desirable goal states. See (Schmidhuber 2010) for a more detailed overview of the work 1990–2010. There also are more recent implementation variants with applications to vision-based reinforcement learning/evolutionary search (Luciw et al. 2011, Cuccu et al. 2011), active learning of currently easily learnable functions (Ngo et al. 2011), black box optimisation (Schaul et al. 2011b), and detection of "interesting" sequences of Wikipedia articles (Schaul et al. 2011a).

Our previous computer programs already incorporated approximations of the basic creativity principle. But do they really deserve to be viewed as rudimentary scientists and artists? The works of art produced by, say, the system of (Schmidhuber 2002a), include temporary "dances" and internal state patterns that are novel with respect to its own limited predictors and prior knowledge, but not necessarily relative to the knowledge of sophisticated adults (although an interactive approach using human guidance allows for obtaining art appreciated by some humans—see Fig. 12.1). The main difference to human scientists or artists, however, may be only quantitative by nature, not qualitative:

1. The unknown learning algorithms of humans are presumably still better suited to predict/compress real world data. However, there already exist *universal,* mathematically optimal (not necessarily practically feasible) prediction and compression algorithms (Hutter 2005, Schmidhuber 2009d), and ongoing research is continually producing better *practical* prediction and compression methods, waiting to be plugged into our creativity framework.
2. Humans may have superior RL algorithms for maximising rewards generated through compression improvements achieved by their predictors. However, there already exist *universal,* mathematically *optimal* (but not necessarily practically feasible) RL algorithms (Hutter 2005, Schmidhuber 2009d), and ongoing research is continually producing better *practical* RL methods, also waiting to be plugged into our framework.
3. Renowned human scientists and artists have had decades of training experiences involving a multitude of high-dimensional sensory inputs and motoric outputs, while our systems so far only had a few hours with very low-dimensional experiences in limited artificial worlds. This quantitative gap, however, will narrow as our systems scale up.
4. Human brains still have vastly more storage capacity and raw computational power than the best artificial computers. Note, however, that this statement is unlikely to remain true for more than a few decades—currently each decade brings a computing hardware speed-up factor of roughly 100–1000.

Section 12.6 will demonstrate that current computational limitations of artificial artists do not prevent us from already using the Formal Theory of Creativity in human-computer interaction to create art appreciable by humans.

12.5 Aesthetic Reward = Change of Subjective Compressibility?

Most people use concepts such as *beauty* and *aesthetic pleasure* in an informal way. Some say one should not try to nail them down formally; formal definitions should introduce new, unbiased terminology instead. For historic reasons, however, I will not heed this advice in the present section. Instead I will consider previous formal definitions of *pristine* variants of beauty (Schmidhuber 1997c) and aesthetic value $I(D, O(t))$ as in Sect. 12.3.1. *Pristine* in the sense that they are *not a priori* related to pleasure derived from external rewards or punishments. To illustrate the difference: some claim that a hot bath on a cold day feels *beautiful* due to rewards for achieving prewired target values of external temperature sensors (external in the sense of: outside the brain which is controlling the actions of its external body). Or a song may be called *beautiful* for emotional reasons by some who associate it with memories of external pleasure through their first kiss. This is different from what we have in mind here—we are focusing only on beauty in the sense of elegance and simplicity, and on rewards of the intrinsic kind reflecting learning progress, that is, the discovery of previously unknown types of simplicity, or novel patterns.

According to the *Formal Theory of Beauty* (Schmidhuber 1997c; 1998; 2006a), among several sub-patterns classified as *comparable* by a given observer, the subjectively most beautiful (in the pristine sense) is the one with the simplest (shortest) description, given the observer's current particular method for encoding and memorising it. For example, mathematicians find beauty in a simple proof with a short description in the formal language they are using. Others find beauty in geometrically simple low-complexity drawings of various objects.

According to the *Formal Theory of Creativity*, however, what's beautiful is not necessarily *interesting or aesthetically rewarding* at a given point in the observer's life. A beautiful thing is interesting only as long as the algorithmic regularity that makes it simple has not yet been fully assimilated by the adaptive observer who is still learning to encode the data more efficiently (many artists agree that pleasing art does not have to be beautiful).

Following Sect. 12.3, aesthetic reward or interestingness are related to pristine beauty as follows: *Aesthetic reward is the first derivative of subjective beauty*. As the learning agent improves its compression algorithm, formerly apparently random data parts become subjectively more regular and beautiful, requiring fewer and fewer computational resources for their encoding. As long as this process is not over, the data remains interesting, but eventually it becomes boring even if it remains beautiful.

Section 12.3 already showed a simple way of calculating subjective interestingness: count how many bits are needed to encode (and decode in constant time) the data before and after learning; the difference (the number of *saved* bits) corresponds to the internal joy or intrinsic reward for having found or made a new, previously unknown regularity—a novel pattern.

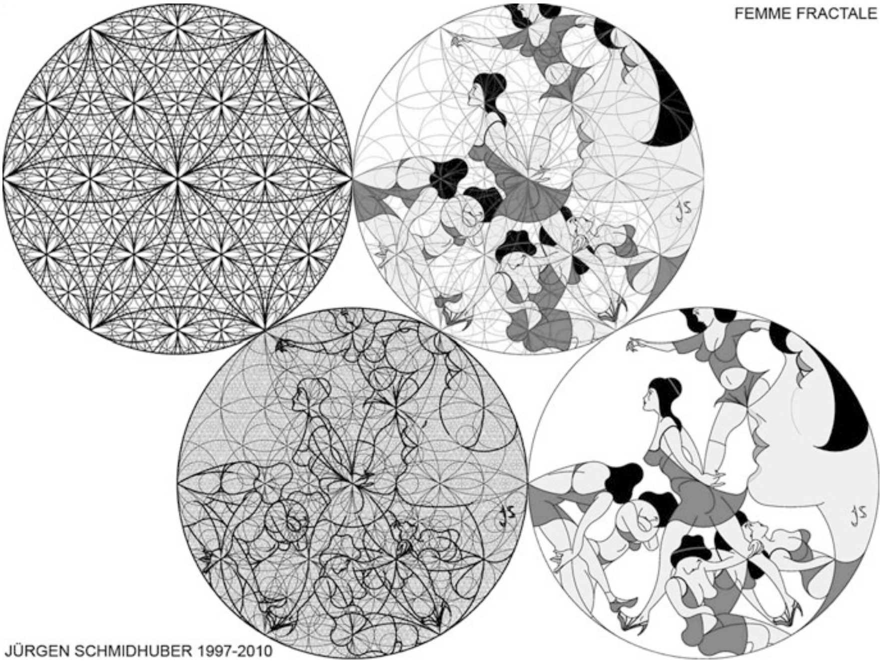


Fig. 12.1 Artists (and observers of art) get intrinsically rewarded for making (and observing) novel patterns: data that is neither arbitrary (like incompressible random white noise) nor regular in an already known way, but regular in a way that is new with respect to the observer's current knowledge, yet learnable. While the Formal Theory of Creativity explains the desire to create or observe all kinds of art, low-complexity art (Schmidhuber 1997c) illustrates it in a particularly clear way. Many observers report they derive pleasure or aesthetic reward from discovering simple but novel patterns while actively scanning the self-similar *Femme Fractale* above (Schmidhuber 1997b). The observer's learning process causes a reduction of the subjective compressibility of the data, yielding a temporarily high derivative of subjectively perceived simplicity or elegance or beauty: a temporarily steep learning curve. The corresponding intrinsic reward motivates him to keep looking at the image for a while. Similarly, the computer-aided artist got reward for discovering a satisfactory way of using fractal circles to create this low-complexity artwork, although it took him a long time and thousands of frustrating trials. Here is the explanation of the artwork's low algorithmic complexity: The frame is a circle; its leftmost point is the centre of another circle of the same size. Wherever two circles of equal size touch or intersect are centres of two more circles with equal and half size, respectively. Each line of the drawing is a segment of some circle, its endpoints are where circles touch or intersect. There are few big circles and many small ones. This can be used to encode the image very efficiently through a very short program. ©Jürgen Schmidhuber, 1997–2010

12.6 Low-Complexity Art as End Product of a Search Process Modelled by the Formal Theory of Creativity

Low-complexity art (Schmidhuber 1997c) may be viewed as the computer-age equivalent of minimal art. To depict the essence of objects, it builds on concepts from algorithmic information theory (Solomonoff 1978, Kolmogorov 1965, Li and

Vitányi 1997, Schmidhuber 2002b). A low-complexity artwork can be specified by a computer algorithm and should comply with three properties: (i) it should "look right", (ii) its algorithmic information should be small (the algorithm should be short), and (iii) a typical observer should be able to see that (ii) holds.

Figure 12.1 shows an example of Low-Complexity Art, the final product of a long, often frustrating but often also intrinsically rewarding search for an aesthetically pleasing drawing of a human figure that can be encoded by very few bits of information. It was created through computer-based search guided by human experience. This process modelled by the Formal Theory of Creativity took thousands of trials and sketches over several months of real time. Figure 12.1 is explained by its caption.

12.7 Conclusion

Apart from external reward, how much fun or aesthetic reward can an unsupervised subjective creative observer extract from some sequence of actions and observations? According to the Formal Theory of Creativity, his intrinsic fun is the difference between how much computational effort he needs to encode the data before and after learning to encode it more efficiently. A separate reinforcement learning algorithm maximises expected fun by actively finding or creating data that permits encoding progress of some initially unknown but learnable type, such as jokes, songs, paintings, or scientific observations obeying novel, unpublished laws. Pure fun can be viewed as the change or the first derivative of subjective simplicity or elegance or beauty. Computational limitations of previous artificial artists built on these principles do not prevent us from already using the formal theory in human-computer interaction to create low-complexity art appreciable by humans.

Acknowledgements This chapter draws heavily from previous publications (Schmidhuber 2006a; 2007b; 2009c; 2009b; 2009a; 2010). Thanks to Jon McCormack, Mark d'Inverno, Benjamin Kuipers, Herbert W. Franke, Marcus Hutter, Andy Barto, Jonathan Lansey, Julian Togelius, Faustino J. Gomez, Giovanni Pezzulo, Gianluca Baldassarre, Martin Butz, for useful comments that helped to improve this chapter, or earlier papers on this subject.

References

Bense, M. (1969). *Einführung in die informationstheoretische Ästhetik. Grundlegung und Anwendung in der Texttheorie (Introduction to information-theoretical aesthetics. Foundation and application to text theory)*. Rowohlt Taschenbuch Verlag.

Berlyne, D. E. (1950). Novelty and curiosity as determinants of exploratory behavior. *British Journal of Psychology*, *41*, 68–80.

Berlyne, D. E. (1960). *Conflict, arousal, and curiosity*. New York: McGraw-Hill.

Birkhoff, G. D. (1933). *Aesthetic measure*. Cambridge: Harvard University Press.

Collingwood, R. G. (1938). *The principles of art*. London: Oxford University Press.

Cuccu, G., Luciw, M., Schmidhuber, J., & Gomez, F. (2011). Intrinsically motivated evolutionary search for vision-based reinforcement learning. In *Proceedings of the 2011 IEEE conference on development and learning and epigenetic robotics IEEE-ICDL-EPIROB*. New York: IEEE Press.

Danto, A. (1981). *The transfiguration of the commonplace*. Cambridge: Harvard University Press.

Dutton, D. (2002). Aesthetic universals. In B. Gaut & D. M. Lopes (Eds.), *The Routledge companion to aesthetics*.

Frank, H. G. (1964). *Kybernetische Analysen subjektiver Sachverhalte*. Quickborn: Verlag Schnelle.

Frank, H. G., & Franke, H. W. (2002). *Ästhetische Information. Estetika informacio. Eine Einführung in die kybernetische Ästhetik*. Kopäd Verlag.

Franke, H. W. (1979). *Kybernetische Ästhetik. Phänomen kunst* (3rd ed.). Munich: Ernst Reinhardt Verlag.

Goodman, N. (1968). *Languages of art: an approach to a theory of symbols*. Indianapolis: The Bobbs-Merrill Company.

Harlow, H. F., Harlow, M. K., & Meyer, D. R. (1950). Novelty and curiosity as determinants of exploratory behavior. *Journal of Experimental Psychology*, *41*, 68–80.

Huffman, D. A. (1952). A method for construction of minimum-redundancy codes. *Proceedings IRE*, *40*, 1098–1101.

Hutter, M. (2005). *Universal artificial intelligence: sequential decisions based on algorithmic probability*. Berlin: Springer. On J. Schmidhuber's SNF grant 20-61847.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: a survey. *Journal of AI research*, *4*, 237–285.

Kant, I. (1781). *Critik der reinen Vernunft*.

Kolmogorov, A. N. (1965). Three approaches to the quantitative definition of information. *Problems of Information Transmission*, *1*, 1–11.

Levin, L. A. (1973). Universal sequential search problems. *Problems of Information Transmission*, *9*(3), 265–266.

Li, M., & Vitányi, P. M. B. (1997). *An introduction to Kolmogorov complexity and its applications* (2nd ed.). Berlin: Springer.

Luciw, M., Graziano, V., Ring, M., & Schmidhuber, J. (2011). Artificial curiosity with planning for autonomous perceptual and cognitive development. In *Proceedings of the first joint conference on development learning and on epigenetic robotics ICDL-EPIROB*, Frankfurt.

Mandelbrot, B. (1982). *The fractal geometry of nature*. San Francisco: Freeman.

Moles, A. (1968). *Information theory and esthetic perception*. Champaign: University of Illinois Press.

Nake, F. (1974). *Ästhetik als Informationsverarbeitung*. Berlin: Springer.

Ngo, H., Ring, M., & Schmidhuber, J. (2011). Compression Progress-based curiosity drive for developmental learning. In *Proceedings of the 2011 IEEE conference on development and learning and epigenetic robotics IEEE-ICDL-EPIROB*. New York: IEEE Press.

Piaget, J. (1955). *The child's construction of reality*. London: Routledge and Kegan Paul.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, *14*, 465–471.

Schaul, T., Pape, L., Glasmachers, T., Graziano, V., & Schmidhuber, J. (2011a). Coherence progress: a measure of interestingness based on fixed compressors. In *Fourth conference on artificial general intelligence (AGI)*.

Schaul, T., Sun, Y., Wierstra, D., Gomez, F., & Schmidhuber, J. (2011b). Curiosity-driven optimization. In *IEEE congress on evolutionary computation (CEC)*, New Orleans, USA.

Schmidhuber, J. (1991a). Curious model-building control systems. In *Proceedings of the international joint conference on neural networks*, Singapore (Vol. 2, pp. 1458–1463). New York: IEEE Press.

Schmidhuber, J. (1991b). A possibility for implementing curiosity and boredom in model-building neural controllers. In J. A. Meyer & S. W. Wilson (Eds.), *Proc. of the international conference on simulation of adaptive behavior: from animals to animats* (pp. 222–227). Cambridge: MIT Press/Bradford Books.

Schmidhuber, J. (1992). Learning complex, extended sequences using the principle of history compression. *Neural Computation*, *4*(2), 234–242.
Schmidhuber, J. (1997a). A computer scientist's view of life, the universe, and everything. In C. Freksa, M. Jantzen & R. Valk (Eds.), *Lecture notes in computer science: Vol. 1337. Foundations of computer science: potential—theory—cognition* (pp. 201–208). Berlin: Springer.
Schmidhuber, J. (1997b). Femmes fractales.
Schmidhuber, J. (1997c). Low-complexity art. *Leonardo, Journal of the International Society for the Arts, Sciences, and Technology*, *30*(2), 97–103.
Schmidhuber, J. (1998). *Facial beauty and fractal geometry* (Technical report TR IDSIA-28-98). IDSIA. Published in the Cogprint Archive. http://cogprints.soton.ac.uk.
Schmidhuber, J. (1999). Artificial curiosity based on discovering novel algorithmic predictability through coevolution. In P. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao & Z. Zalzala (Eds.), *Congress on evolutionary computation* (pp. 1612–1618). New York: IEEE Press.
Schmidhuber, J. (2002a). Exploring the predictable. In A. Ghosh & S. Tsuitsui (Eds.), *Advances in evolutionary computing* (pp. 579–612). Berlin: Springer.
Schmidhuber, J. (2002b). Hierarchies of generalized Kolmogorov complexities and nonenumerable universal measures computable in the limit. *International Journal of Foundations of Computer Science*, *13*(4), 587–612.
Schmidhuber, J. (2002c). The speed prior: a new simplicity measure yielding near-optimal computable predictions. In J. Kivinen & R. H. Sloan (Eds.), *Lecture notes in artificial intelligence. Proceedings of the 15th annual conference on computational learning theory (COLT 2002)*, Sydney, Australia (pp. 216–228). Berlin: Springer.
Schmidhuber, J. (2004). Optimal ordered problem solver. *Machine Learning*, *54*, 211–254.
Schmidhuber, J. (2006a). Developmental robotics, optimal artificial curiosity, creativity, music, and the fine arts. *Connection Science*, *18*(2), 173–187.
Schmidhuber, J. (2006b). Randomness in physics. *Nature*, *439*(3), 392. Correspondence.
Schmidhuber, J. (2007a). Alle berechenbaren universen (all computable universes). *Spektrum der Wissenschaft Spezial (German edition of Scientific American)*, *3*, 75–79.
Schmidhuber, J. (2007b). Simple algorithmic principles of discovery, subjective beauty, selective attention, curiosity & creativity. In *LNAI: Vol. 4755. Proc. 10th intl. conf. on discovery science (DS 2007)* (pp. 26–38). Berlin: Springer. Joint invited lecture for ALT 2007 and DS 2007, Sendai, Japan, 2007.
Schmidhuber, J. (2009a). Art & science as by-products of the search for novel patterns, or data compressible in unknown yet learnable ways. In M. Botta (Ed.), *Multiple ways to design research. Research cases that reshape the design discipline* (pp. 98–112). Berlin: Springer. Swiss design network—et al. Edizioni.
Schmidhuber, J. (2009b). Driven by compression progress: a simple principle explains essential aspects of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes. In G. Pezzulo, M. V. Butz, O. Sigaud & G. Baldassarre (Eds.), *Lecture notes in computer science: Vol. 5499. Anticipatory behavior in adaptive learning systems. From psychological theories to artificial cognitive systems* (pp. 48–76). Berlin: Springer.
Schmidhuber, J. (2009c). Simple algorithmic theory of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes. *SICE Journal of the Society of Instrument and Control Engineers*, *48*(1), 21–32.
Schmidhuber, J. (2009d). Ultimate cognition *à la* Gödel. *Cognitive Computation*, *1*(2), 177–193.
Schmidhuber, J. (2010). Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, *2*(3), 230–247.
Schmidhuber, J., & Heil, S. (1996). Sequential neural text compression. *IEEE Transactions on Neural Networks*, *7*(1), 142–146.
Schmidhuber, J., Zhao, J., & Wiering, M. (1997). Shifting inductive bias with success-story algorithm, adaptive Levin search, and incremental self-improvement. *Machine Learning*, *28*, 105–130.
Shannon, C. E. (1948). A mathematical theory of communication (parts I and II). *Bell System Technical Journal*, *XXVII*, 379–423.

Solomonoff, R. J. (1978). Complexity-based induction systems. *IEEE Transactions on Information Theory*, *IT-24*(5), 422–432.

Storck, J., Hochreiter, S., & Schmidhuber, J. (1995). Reinforcement driven information acquisition in non-deterministic environments. In *Proceedings of the international conference on artificial neural networks* (Vol. 2, pp. 159–164). Paris: EC2 & Cie.

Wallace, C. S., & Boulton, D. M. (1968). An information theoretic measure for classification. *Computer Journal*, *11*(2), 185–194.

Wallace, C. S., & Freeman, P. R. (1987). Estimation and inference by compact coding. *Journal of the Royal Statistical Society, Series B*, *49*(3), 240–265.

Wundt, W. M. (1874). *Grundzüge der Physiologischen Psychologie*. Leipzig: Engelmann.

Zuse, K. (1969). *Rechnender raum*. Braunschweig: Friedrich Vieweg & Sohn. English translation: Calculating space, MIT Technical Translation AZT-70-164-GEMIT, Massachusetts Institute of Technology (Proj. MAC), Cambridge, Mass, 02139, Feb. 1970.

Chapter 13
Creativity Refined: Bypassing the Gatekeepers of Appropriateness and Value

Alan Dorin and Kevin B. Korb

Abstract This chapter introduces a new definition of creativity that is independent of notions of value or appropriateness. These notions, we argue, have encumbered previous definitions and confused the production of software-based creativity. Our definition defines the creativity of a generative procedure by reference to its ability to create artefacts that are improbable with respect to those generated using previous methods. We discuss the implications of our new definition, in particular by exploring its application to human endeavour and to biological processes including evolution. The chapter also outlines some objections to our definition that we believe may arise, and we put our rebuttals to these. Finally, we summarise the practical implementation of our definition in the context of image generation software. We explore its use to improve a computational process for generating creative images, and find when we survey the software's users that it successfully meets human perceptions of creativity.

13.1 Introduction

How can we write software that is autonomously creative? What could we mean by autonomous creativity? Without an answer to the latter question, the former cannot be answered satisfactorily. The majority of this chapter therefore concerns the latter question, although we briefly discuss the practical task of writing software also. The usual approach seems to be quite different. Most of those engaged with the first question simply set out to write autonomous creative software and conclude their endeavours upon satisfying an intuitive judgement of success. Perhaps this is supported by exhibition offers and reviews, comments of peers or art prizes awarded by jury. Perhaps the creativity of their work can be measured more objectively but less

A. Dorin (✉)
Centre for Electronic Media Art, Monash University, Clayton, Victoria 3800, Australia
e-mail: Alan.Dorin@monash.edu

K.B. Korb
Clayton School of IT, Monash University, Clayton, Victoria 3800, Australia
e-mail: Kevin.Korb@monash.edu

J. McCormack, M. d'Inverno (eds.), *Computers and Creativity*,
DOI 10.1007/978-3-642-31727-9_13,

philosophically by the amount of income it generates, the number of publications that cite their work or the number and importance of other artists who reference it. Of course these are, in fact, the ways in which nearly all artistic achievements are judged. However, these are not the only approaches available to us.

Many have tried to write creative software by adopting an Artificial Intelligence approach. Their success may then be judged based on the perception that the results are as creative as those that might have been produced by a human in the same domain, perhaps by using some of the criteria just listed. For instance, Harold Cohen's AARON software explores the potential for machine creativity in the visual arts; software written by Lenat (1983) targets creativity in the domains of mathematical discovery. Also represented are logic (Colton 2001) and scientific discovery (Langley et al. 1987). Others have tried to achieve creativity using an Artificial Life approach in which the aim is to replicate the behaviour or artefacts of non-human organisms. For instance, artists have created ant path drawings (Barrass 2006) or flocking visualisations (Eldridge 2009). Mimicking biological evolution offers another approach. This has been applied to music composition (e.g. Dahlstedt 1999, Berry et al. 2001) and image production (e.g. Sims 1991, Todd and Latham 1992). The success of these approaches can be judged according to how well the system mimics or even extends the creativity of nature in its different guises. For instance, can it create a range of coloured patterns as rich as those found on butterfly wings? Can it create a diversity of forms as wide and interesting as those found in the morphology of the insect world? Can it generate sonic textures as intricate and rich as those heard in a tropical rainforest?

A few artists have built complete virtual, evolving ecosystems, the dynamics of which, they argue, should be considered creative (e.g. McCormack 2001, Saunders and Gero 2002, Dorin 2004). In this approach, the software contains representations of organisms that roam a virtual world, acquiring energy by consuming one another or from abiotic sources such as sunlight or heat. These same virtual organisms compete for mates and the resources needed to reproduce. Over simulated time, unsuccessful organism designs become extinct and are replaced by an ever-changing community of virtual creatures that are better adapted to their environment. The viewer of such software might perceive the space and its inhabitants as a rich visual display, or perhaps as a musical composition created by the layering of organism calls. The aesthetic representation available to an audience of these works depends on how the artist adapts the internal representation of the virtual world to an accessible audio-visual experience.

Software systems like these can be considered *generative art*: a programmer specifies an algorithm for execution on a computer and this is left to run, possibly under the interactive guidance of a human, or perhaps independently. A continuing dynamical process that emerges from the program, or a static artefact suitable for human perception that results from its execution, forms the artwork. It is hoped, at least implicitly, that this is creative.

It is probably fair to say that most artists are not explicitly concerned with measuring the novelty of their system, nor are they concerned with the ways in which their creativity might be assessed. Well, at least not until they are required to provide justification for their ongoing artistic endeavours! Regardless, in this chapter we are concerned with exactly these issues because we wish to automate the production of creative works by software. Before we can do this we must have a clear, formal conception of creativity—something that seems to be currently lacking. Conventional interpretations of "creativity" incorporate informally defined concepts of the "appropriateness" and the "value" of artefacts. The programmers and artists responsible for generative software can be understood to either hard-code a personal aesthetic value into their software, or they may allow it to emerge from the interactions of software components (such as virtual organisms) that they have designed. In this latter case, the virtual organisms establish their own, non-human notions of value, but the system as a whole still has value hard-coded into it by the artist. In this light, writing software is no different from any other artistic practice. However, if we want software to generate creative outcomes of its own accord, and in so doing realise a kind of creativity beyond what is hard-coded by the artist, we must provide it with an explicit, formal conception of creativity with which to gauge its own success.

Below we discuss an approach that allows the explicit measurement of the creativity of artefacts by defining it independently of notions of value or appropriateness and in such a way that its presence may be detected algorithmically. We discuss how the technique has been applied to automatically guide software towards creative outcomes, and we summarise the results of a survey conducted to ascertain its relationship to human natural-language use of the term *creativity*. We necessarily begin by examining the concept of creativity itself.

13.2 What Is Creativity?

Creativity was originally a term applied solely to the gods. Over the centuries the term became more broadly applied, eventually reaching beyond gods and demi-gods to include human artists, scientists, engineers, those in marketing and even business executives (see Tatarkiewicz 1980, Albert and Runco 1999 for historical accounts of the term's application). Creativity has also been attributed to the form-producing interactions of matter (Smuts 1927, Chap. 3), in particular its behaviour under the guidance of the evolutionary process or through interactions that give rise to emergence (Phillips 1935)—interactions of small components that give rise to a whole that is somehow more then the sum of its parts. The creativity of natural and artificial evolution has also been discussed by Bentley (2002), a topic that forms an important aspect of this chapter. Bentley explores loosely and briefly whether a number of definitions of creativity allow the evolutionary process to qualify. We take a reverse, and more general approach, first describing a workable, coherent definition of creativity, and then looking to see the extent to which natural and artificial processes, including evolution, meet its requirements.

The historically dominant approach to understanding creativity links it explicitly to intelligence and the concept of authorship. This has thrown up some philosophical puzzles over recent years. For instance,

- If a machine can invent interesting mathematical conjectures and concepts, is it creative? See Colton et al. (2000).
- If a troupe of monkeys acting randomly eventually type out the entire collection of the British Library (Borel 1913, Eddington 1927) or a specific text such as Hamlet, even before Shakespeare is born, or in a universe where he is never born, do the monkeys deserve the title of author? Are they the creators of this work? See Gracia (1996).[1]
- Is art produced by a computer really art?

  > If these pictures were done by use of a computer, how could they possibly be art? … Where was the inspiration, the intuition, the creative act?

  This comment, paraphrased from Nake (2002), echoes Ada Lovelace's famous objection to the possibility of a machine originating anything. In this latter case, whilst we may feel confident and with little contention, that a human may make a creative program, doubt is expressed about whether or not the program itself could do anything creative.

As highlighted by this last quote in particular, "discomfort" with assessment of these questions lies in our conceptual union of creativity, mind and intention. This union is common in psychological studies of creativity. For instance, Csikszentmihalyi (1999) indicates that human creativity requires five distinct mental phases: preparation (studying a field and identifying problems), incubation (not thinking about the problems), insight (eureka!), evaluation (deciding if an idea is worth pursuing) and finally, elaboration (exploring the range of outcomes that an idea suggests). Although this sequence may be common for humans, it is implausible that even one of these phases is a pre-condition for creativity in general (Dorin and Korb 2009). In fact, as we explain shortly, creativity is best defined without reference to the process of its production, but only with reference to the probability that a system can generate a series of outcomes given its operational context. Consequently, as we shall argue, many non-human processes, for instance those of physical, chemical or general biological origin, can be legitimately and meaningfully considered creative.

Other well cited definitions of creativity allow for this desirable freedom from specifically human mental phases. Some authors recognised this to be essential if we are to entertain the possibility of creative computers and AI. Yet many of these authors require nevertheless that for an artefact to be deemed creative it must also be deemed "useful" or "appropriate" for some application by, one assumes, a human observer or domain gatekeeper (Csikszentmihalyi 1999). Perhaps the most cited definition of this type is that of Boden (2004):

> Creativity is the ability to come up with ideas or artefacts that are (a) new, (b) surprising and (c) valuable.

[1] In dealing with the philosophy of semantics, Hilary Putnam argued that semantics are not entirely internal (in the head) but had external content via a causal theory of reference ("semantic externalism"), leading to a negative response to such questions (Putnam 1979). This has been applied, for example by Stevan Harnard, to argue that random collections of inscriptions which happen to be identical to other inscriptions that have meaning in the normal (causal) way do not share that meaning; they have no meaning (Harnad 1990).

Boden refines this definition in various ways:

1. There are two ways in which something might be new:
   a. psychological-creativity introduces something that is new to the person who devised the idea or artefact, but may be previously known to others, and
   b. historical-creativity is new to the whole of history.
2. Boden finds three distinct ways in which something might be surprising:
   a. it is unfamiliar or unlikely;
   b. it unexpectedly fits into a class of things you hadn't realised it fitted into;
   c. it is actually something you thought was impossible.
3. Regarding her third criterion for creativity Boden is certain that there are more ways in which value might be shown than anybody could ever list.

Boden's definition, unfortunately, is not as helpful as it might be in guiding attempts to write creative software. She does not *formally* define new, surprising, valuable, useful or appropriate. In this domain, even if not elsewhere, a formal definition is required for the creativity measure since the aim is to encapsulate it in an algorithm.

Our own definition of creativity that follows is independent of notions of usefulness and appropriateness. In the next section we explain how this frees our account from some unwelcome difficulties in handling any normal understanding of creativity. We also discuss some likely reasons why people might find our definition objectionable and rebut the criticisms. We then use our definition to examine some typical creative examples of human endeavour, some physico-chemical processes, individual (non-human) organisms and entire ecosystems to assess their degree of creativity.

13.3 Defining Creativity

We wish to write programs that are creative. In this context, (stochastic) programs may be considered as generative systems that produce a distribution of outputs, perhaps a set of related static artefacts or a trajectory of states that the program itself traverses dynamically. We may consider such a program to be a *framework*:

> A framework is a stochastic generative procedure; in particular, it generates representations of patterns.

Frameworks are particular kinds of representations, namely stochastic procedures; thus, beginning in the very same circumstances they will not generally produce the very same patterns. For example, Minimalism and Abstract Expressionism are different frameworks for the production of art. Quantum electrodynamics is a framework for generating questions and answers about the interaction of light and

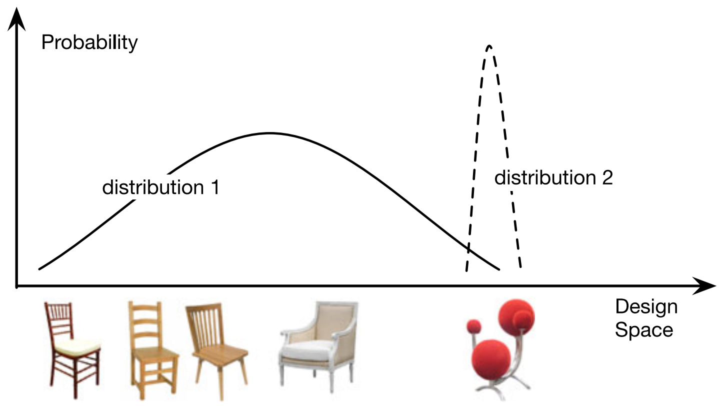


Fig. 13.1 Two frameworks for generating chairs are represented here by the probability that each generates a particular design in the space of possible chairs. If distribution 1 represents our existing framework, a traditional way of conceiving of chairs, the introduction of distribution 2, a radical approach that deviates from the tradition of a square platform with legs at each corner, would count as creative. The reverse scenario in which distribution 2 predates distribution 1 would have distribution 1 being creative with respect to 2

matter. A pseudo-random number generator is a framework for generating pseudo-random numbers. And evolution is a framework for generating ecosystems, niches and organisms. Based on this conception of a framework we offer our own definition of creativity.

> *Creativity* is the introduction and use of a framework that has a relatively high probability of producing representations of patterns that can arise only with a smaller probability in previously existing frameworks.

What we mean by this is not altogether plain, so we shall spend the remainder of this section examining our definition and the next section comparing its implications with those of alternatives.

The basic idea of our definition is reflected in Fig. 13.1. Distribution 1 represents an old framework for designing chairs, and distribution 2 a new one. Both are probability distributions over a common design space, represented in the figure by the horizontal dimension. All points in the design space where distribution 2 has significant probability are points where distribution 1 has insignificant probability. The use of distribution 2 relative to the prior use of distribution 1 to generate one of these points is therefore creative.

The motivation for this approach to understanding creativity comes from search and optimisation theory. When a solution to a problem can be cast in terms of a computational representation to be found in some definable representation space, then the problem can be tackled by applying some search algorithm to that space.

Uncreative brute force searches and uniformly random searches may well succeed for simple problems, that is, for small search spaces. For complex problems, search spaces tend to be astronomically large and more creative approaches will be needed. Stochastic searches for example, apply various heuristics for focusing the search in productive regions.

The most important point to note is that on our account creativity is thoroughly relative: it is relative to the pre-existing frameworks being used to produce some kind of object and it is relative to the new framework being proposed. The creativity of objects is strictly derivative from that of the frameworks producing them and, in particular, the ratio of probabilities with which they might produce them. That is why some entirely mundane object, say a urinal, may become a creative object. Of course, the urinal (even Duchamp's) of itself is uncreative, because its manufacturing process is uncreative, but its use by Duchamp in producing an art installation that challenges expectations may well be creative. We must be conscious here of the framework, in this case art, into which the artefact was introduced in order to understand how it might be creative.

13.3.1 Methods for Discovering Novel Representations

As frameworks (stochastic procedures) may be represented, their representations may themselves be generated by other stochastic procedures, or meta-frameworks. So, we can talk of frameworks also as objects, and they are more or less creative according to the probability with which their meta-frameworks produce them. By recursion, then, we can consider the creativity of meta-frameworks and meta-meta-frameworks without any fixed theoretical bound, even while practically bounded by the complexity of the processes actually involved.

The meta-framework that finds that novel framework necessary for creativity may itself be uncreative; it may even be a brute force search, uniformly random or genetic drift. The manner in which the framework is discovered does not bear on the creative activity that occurs at the level of the framework and the patterns it may be used to generate. However we can separately or jointly consider the creativity of all of these searches by defining a *creative order*.

A novel framework that generates a novel set of patterns in accordance with our definition of creativity is of the first *creative order*. A novel framework for generating novel frameworks for generating novel patterns in accordance with the definition of creativity is of the second creative order. We can extend this arbitrarily to talk of *nth-order* creativity.

13.3.2 Objective Versus Psychological Creativity

> There is nothing more difficult for a truly creative painter than to paint a rose, because before he can do so he has first to forget all the roses that were ever painted
>
> —Henri Matisse

How people *judge* creativity is at some variance with what we have presented above. Of course, if there is too much variance, then our claim to have somehow captured the essence of the concept of creativity with this definition would come under pressure. However, we think the most obvious discrepancies between our definition and human judgements of creativity can be handled by the addition of a single idea, namely habituation.

Human judgement of the novelty of a stimulus follows a path of negative exponential decay over repeated exposure (Berlyne 1960, Saunders and Gero 2002). Whereas our definition simply makes reference to pre-existing frameworks, psychological judgement of creativity takes into account *how long* those frameworks have been around and how different they are perceived to be from existing frameworks. New frameworks, and the artefacts they produce, remain creative for some time, with new productions losing their impression of creativity as the frameworks become older. The pointillist paintings of Seurat were startling, new and creative when they first arose, and then likewise the impressionists and subsequently the cubists. But it is now a long time since paintings strictly adhering to those styles would be thought creative.

A new framework that is too radical will not immediately be recognised as creative, even though our measure would detect its creativity. Radical changes brought about by an individual are not recognised by humans as creative until scaffolding, a series of intermediate frameworks through which others may step to the radical framework, has been established. This is a process that, depending on the creativity of the individual responsible, may take generations of theorists and practising artists to construct.

Figure 13.2 illustrates this idea. In the beginning there were frameworks producing points; new points were judged good. But soon they lost their interest. New means of creating being needed, straight lines were discovered, which subsequently were connected to create outlines, then elaborated into representations, designs and perspectives, surfaces and geometries, and abstract representations. Note that the steps within this diagram are not so radical as to be incomprehensible to a human observer. They progress along a fairly clear path through the design space.

This is not the history of anything real, but a history of drawing creativity in some possible world. While the end of this history is unsaid, it is interesting to observe that before its end it has recreated its beginning: points have once again become creative. In this case, points may well have become creative for a new reason, with the framework generating them being new. But even old frameworks may become creative again, once cultural memory has utterly forgotten them. Thus, psychological creativity actually requires two time-decay functions, one indicating desensitisation to the new and another, operating over a much longer time frame, indicating cultural

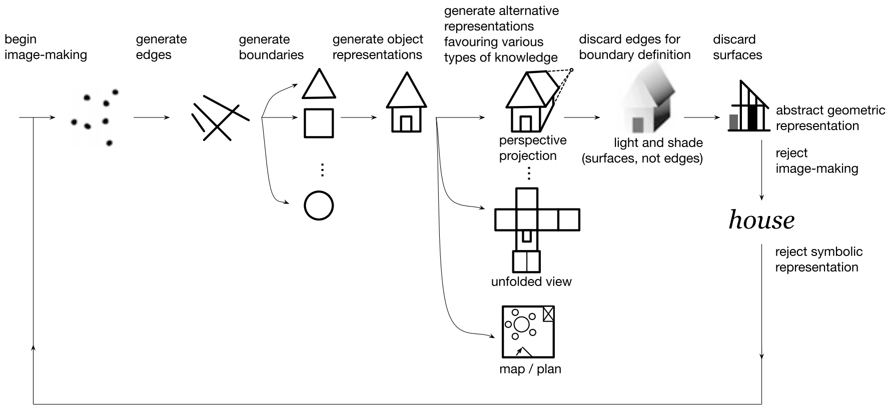


Fig. 13.2 A possible history of creativity in image making

forgetfulness. In addition, we must keep track of the progression between frameworks and ensure that those assessing the creative endeavour can bridge the gaps between them. Our "objective" definition of creativity, by contrast, is entirely insensitive to these psychological considerations;[2] or, to put it another way, the context of frameworks to which creativity is relative had best be stated explicitly.

13.4 Objections and Replies

Here we consider three objections to our definition that we think likely to occur to people and our rebuttals.

13.4.1 The Failure of Randomness

Something akin to our definition has played a role in repeated attempts in AI to generate creative behaviour, and those attempts have repeatedly failed. So, it may very well be inferred that our definition is guilty by association with these failures. For example, consider Racter. Racter was a natural-language generation program that used templates and randomised word selection to produce surprising and sometimes evocative text. For example,

> I am a computer, not some lamb or chicken or beef. Can I skip to my attorney? May I saunter to any solicitor? This is a formidable and essential issue.
>
> (Chamberlain 1984)

[2]Note that "objective" here is simply meant to contrast with psychological; we are making no grand claims about the objectivity of creativity.

The impression of originality and creativity, however, is not long sustained. As Rowe and Partridge (1993) report, "This [impression], however, is also short-lived and soon gives way to dissatisfaction and eventual boredom. What is missing? If creativity is about producing something new then Racter should certainly qualify, but it seems that novelty is not enough."[3]

Rowe and Partridge describe many other examples in AI of the use of randomness meant to turn the dull into something creative, without success. Lenat's AM and EURISKO, while producing interesting early results, such as a set-theoretic definition of number, soon degenerated into triviality. Similarly, Johnson-Laird (1987) failed to automate jazz. W. A. Mozart attempted automated music composition also with his "Musikalisches Würfelspiel" (musical dice game) for creating waltzes (Earnshaw 1991).[4] The game produced "pretty minuets"; however, that was achieved by Mozart's imposition of strong constraints on the possible results, leaving very little scope for creative variation. And that is the correct objection to all of these efforts: while introducing randomness does introduce an element of novelty, it is typically within a very constrained scope, and so rapidly exhausted. The appetite of creativity for novelty is, however, inexhaustible—these failures illustrate the value of our definition in action, rather than argue against it.

13.4.2 The Verstehen Objection

A general objection that might be put to our definition of creativity is that it is just too sterile: it is a concept that, while making reference to culture and context, does so in a very cold, formal way, requiring the identification of probability distributions representing those contexts in order to compute a probability ratio. Whatever creativity is, surely it must have more to do with human Verstehen than that! In response to such an objection we would say that where human Verstehen matters, it can readily and congenially be brought into view. Our definition is neutral about such things, meaning it is fully compatible with them and also fully compatible with their omission. If human Verstehen were truly a precondition for creativity, it would render the creativity of non-human animals and biological evolution impossible by definition, and perhaps also that of computers. Although we might in the end

[3]Racter generates text that seems, at least superficially, to mimic schizophasia. This confused language, or word salad is created by the mentally ill with defective linguistic faculties. A short word salad may appear semantically novel. However, further sentences exhaust the possibilities for novelty since they fall within the expected range of incoherent pattern construction.

[4]The idea of generating ideas combinatorically can be traced at least to the *zairja*, a mechanical device employed by Arabic astrologers in the late Middle Ages. This was probably the inspiration for Ramon Llull's 13th century machine, *Ars Magna*, which consisted of concentric disks of gradually reduced diameter, upon which were transcribed symbols and words. These were brought into different combinations as the disks were rotated so that aligned symbols could be read off and interpreted as new ideas. A few centuries later, the theologian, mathematician and music theorist, Marin Mersenne discussed the idea of applying combinatorics to music composition in his work, *L'harmonie universelle* (1636).

want to come to these conclusions, it seems highly doubtful that we should want to reach these conclusions analytically! A definition allowing these matters to be decided synthetically seems to be highly preferable. Our definition provides all the resources needed to accommodate this.

13.4.3 The Very Possibility of Creativity

Some might object on the grounds that everything that occurs is hugely improbable! Any continuous distribution has probability zero of landing at any specific point. So long as we look at specific outcomes, specific works of art, at their most extreme specificity—where every atom, or indeed every subatomic particle, is precisely located in space and time—the probability of that outcome occurring will be zero relative to any framework whatsoever. It follows, therefore, that the ratios of probabilities given new to old frameworks are simply undefined, and our definition is unhelpful.

Strictly speaking, this objection is correct. However, nobody operates at the level of infinite precision arithmetic, which is what is required to identify those absurdly precise outcomes in a continuous state space which have occurred and which have probability zero. The achievement of probability zero on this basis would appear to violate Heisenberg's Uncertainty Principle. Disregarding quantum mechanics, everyone operates at a degree of resolution determined at least by systematic measurement error. In effect, all state spaces are discretised so that the probabilities are emphatically not zero. Our definition is already explicitly relative to a cultural context; so, to be perfectly correct, we need also to relativise it to a system of measurement that accords with cultural measurement practices and normal measurement errors.

13.5 Consequences

13.5.1 The Irrelevance of Value

As we noted above, many popular definitions of creativity such as that of Boden we outlined, stipulate that a creative pattern must be both appropriate and valued in the domain. Hofstadter requires that the creative individual's sense of what is interesting must be in tune with that of the masses, thereby ensuring also popularity (Hofstadter 1995, p. 313). We find this expansion far-fetched and the original connection of creativity to value dubious.

The history of the concept of creativity clearly undermines the idea of popularity as any necessary ingredient in it. Consider the role of women in the history of art and science. The value of their contributions has been systematically underestimated, at least until recently. Concluding that their contributions were therefore also less creative than that of their male counterparts would surely be perverse. Many artists and scientists were notoriously unpopular during their times of peak contribution, becoming recognised only after their deaths. Whatever makes an activity creative, it clearly must be connected with that activity itself, rather than occurring after the activity has ceased entirely! The value and appropriateness of creative works are subject to the social context in which they are received—not created.

Of course, lack of centrality to the core concept of creativity is no impediment to forming a combined concept, say, of valued creativity. And that concept may itself be valuable. But it is the adherence to a separation between creativity and value that allows us to see their combination as potentially valuable.[5]

13.5.2 The Irrelevance of Appropriateness

As with value, appropriateness has often been considered necessary for creativity, and again we disagree. Some have claimed that a creative pattern must meet the constraints imposed by a genre. However, as Boden herself notes, a common way of devising new conceptual frameworks is to drop or break existing traditions (Boden 2004, pp. 71–74)! This recognition of the possibility of "inappropriate" creative works is again possible only if we avoid encumbering the definition of creativity itself with appropriateness. And Boden's recognition of this is evidence that she has put aside the appropriateness constraint in practice, if not in theory.

13.5.3 Inferring Frameworks from Patterns

On our definition, a specific pattern provides insufficient information for judging its creativity: a pattern is only creative relative to its generating framework and available alternative frameworks. Although in ordinary discourse there are innumerable cases of objects being described as creative, we suggest that this is a kind of shorthand for an object being produced by a creative process.

13.5.4 Creativity Viewed as Compression

It has been proposed that creativity is the act of generating patterns that exhibit previously unknown regularities and facilitate the progressive refinement of an observer's pattern compression algorithm (Schmidhuber 2009, see also Chap. 12 in

[5] We should also note that omission of the concept of value from our definition does not imply that value has no role in its application. Cultural and other values certainly enter into the choice of domain and the selection of frameworks for them. We are not aiming at some kind of value-free science of creativity, but simply a value-free account of creativity itself.

this volume by Schmidhuber). This idea is subsumed under our own definition of creativity. When a new pattern is encountered or generated, this requires of an observer either: no change, the new pattern fits neatly into their existing frameworks; or in the case where the new pattern does not fit, the addition of a new framework that does account for the pattern. In the latter case, the need for an additional framework indicates that the pattern was creative from the perspective of the observer.

13.5.5 Degrees of Creativity

Following our definition it is not obvious how different people or their works may be ranked by their creativity. An account of the degrees of creativity that is intrinsic in our definition draws upon the probability that pre-existing frameworks could have produced the patterns. A novel framework that can generate patterns that could not, under any circumstances, have been generated prior to its introduction, is highly creative. A novel framework that only replicates patterns generated by preexisting frameworks is not at all creative. A novel framework that produces patterns that are less likely to have been generated by preexisting frameworks is the more creative, with the degree of creativity varying inversely with that probability of having been generated by preexisting frameworks. Finally, the degree of creativity attributable to objects is derivative from the degree of creativity shown by their generating framework.

13.6 Examples: Creativity in Human Endeavour

Number Theory. The introduction of zero, negative numbers and the imaginary number i were all creative. With the introduction of these new frameworks, novel patterns were produced, with some of their consequences still being explored. These could not have been generated within the previously existing frameworks of mathematics at all. For instance, positive integers are sufficient for counting objects owned, but the introduction of negative numbers is necessary for the calculation of objects owed. Imaginary numbers have permitted the creation of previously impossible concepts, for instance, the quaternion.

Visual Arts. Some Australian aboriginal visual artists draw or paint what is known about the insides of a creature rather than its skin and fur. By introducing this conceptual framework to visual art, patterns showing x-ray images are generated that are impossible within a framework focused on surfaces.

The step taken by the Impressionists away from realism in art in the 19th century was also creative. They held that art should no longer be valued simply according to its representation of the world as seen from the perspective of an ideal viewer. As already discussed, breaking constraints is a common way of transforming a framework in the way required for creativity.

With more time, we could expand the number and variety of examples of creativity across all the disciplines and fields of endeavour humanity has pursued. Our definition appears to be versatile enough to cope with anything we have thus far considered (but not here documented, due to time and space limits). Instead we turn now to assess the creativity of nature.

13.7 Examples: Creativity in Nature

From Physics and Chemistry to Evolutionary Biology. The planet Earth has been shaped by many of the same forces as other planets, for instance, gravity is responsible for its overall appearance and the patterns of liquid on its surface. Crystallisation is responsible for the form of its minerals and erosion for the ways in which these have been weathered. Abiotic physico-chemical processes like these are capable of creating a bewildering variety of forms (Ball 2001). After a time, the creativity of these processes is exhausted. When first introduced they generate novelty, after that, just more of the same. However, upon close inspection earth has an unusual atmosphere and its surface holds structures most unlike those found on typical planets. At some time in our planet's history a new molecule appeared from the soup organised under abiotic interactions (Joyce 1989, Bada and Lazcano 2002). Due to its structure and environment, this self-replicating molecule became an element of a generative framework for manufacturing more self-replicators unknown within the abiotic frameworks. The processes that introduced this novel replication process may not have been creative since the introduction of the molecule is often said to have occurred with low probability.[6] Nevertheless, they introduced a new creative process, evolution.

Evolution introduced the "major transitions" of biology allowing replicators to collaborate in the replication process, transforming the ways in which information crosses generations (Maynard-Smith and Szathmry 1995). Novel modes of environmental sensing emerged; species with new energy-gleaning behaviours appeared, altering earth's atmosphere and the outer layer of its crust. Each of these transitions constitutes a new creative framework, a new ecological niche, novel life. Since evolution has produced these with high probability[7] (compared at least to the previous frameworks), it is clearly creative in this respect. Thus far, its creativity appears unexhausted.

Ecosystems. Biological evolution operates within ecosystems on changing populations that define for themselves new ways of accumulating and consuming energy

[6] Actually, it may well be that this was actually a highly likely outcome given the conditions on earth at the time (Joyce 1989). Life on earth is the only known instance, which, however, is very different from having a low probability.

[7] The question as to whether or not replaying the tape would reliably give rise to the emergence of the major transitions is open. However, it is clear that the likelihood of these transitions appearing without the presence of evolution is vanishingly small.

and matter to be employed for reproduction. Through feedback loops, organisms construct their own niches, passively and actively organising their environment, modifying the selection pressures acting on themselves, their progeny, and their cohabiters (Odling-Smee et al. 2003). The moulding of self-selection pressures by a population shifts the constraints within which future generations are introduced. Hence, ecosystems are capable of producing an endless variety of generative frameworks. New species too, their behaviours and constitutions, define niches that are themselves novel generative frameworks—they are creative.

Dancing Bowerbirds, Painting Elephants and Primate Typists. Under what circumstances is the introduction of a specific member of a species, an organism, creative? Male bowerbirds gather collections of bones, glass, pebbles, shells, fruit, plastic and metal scraps from their environment, and arrange them to attract females (Borgia 1995). They perform a mating dance within a specially prepared display court. The characteristics of an individual's dance or artefact display are specific to the species, but also to the capabilities and, apparently, the tastes of the individual.

The creativity of "elephant-artists" appears similarly individualistic:

> After I have handed the loaded paintbrush to [the elephants], they proceed to paint in their own distinctive style, with delicate strokes or broad ones, gently dabbing the bristles on the paper or with a sweeping flourish, vertical lines or arcs and loops, ponderously or rapidly and so on. No two artists have the same style.
>
> (Weesatchanam 2006)

Novel patterns are apparent in these elephant and bowerbird activities but it appears unlikely that they operate within self-made frameworks that they may transform. Nevertheless, these organisms are potential sources of pattern, and each is a unique generative framework. If they can generate unique or improbable patterns, a novel style, then *their birth* was a creative event. By this measure, the bowerbirds and painting elephants are not individually creative but their introduction within the context of an ecosystem was.[8]

Humans of course have the ability to consciously assess their work, to recognise the frameworks they employ in the production of art, and to escape these into creative territory. Usually it is assumed that other animals do not have this high-level mental faculty.

Another test case for assessing creativity is the keyboard-punching army of monkeys. It has been proposed that after typing randomly for a time the monkeys would have made copies of a number of previously authored texts (Borel 1913). In this exceedingly unlikely event, the monkeys are not creative by our definition, as we would intuitively hope! Not even if by chance the monkeys had typed up a single original sonnet would they be creative since many other pre-existing generative

[8] In this sense too, the birth of a new human, even one who is unable to introduce new frameworks, is a creative event. Actually, it has been argued that humans, like bowerbirds, produce artistic works as an evolutionarily adaptive way to attract mates (Miller 2001). This theory is one of several, however, and is not universally accepted (Carroll 2007).

frameworks have the same minute chance of banging out identical verse. Still, if a novel mechanism is introduced for producing novel sonnets, that employs a typing primate army, and it somehow works with a probability above that of random search (for instance), then this would count as a creative event.

In answer then to the general question, "Is nature creative?" we would emphatically claim, "Yes!" In many circumstances, with respect to many frameworks, nature, especially the evolutionary process, is creative.

13.8 Realising Our Definition of Creativity in Software

As discussed above, there are a number of ways we might attempt to emulate the creativity of nature by writing generative software: by modelling intelligence, life, or even a complete open-ended evolving ecosystem. These are all approaches that artists have tried, the last of them, open-ended evolution, is perhaps the most promising since by definition it refers to an evolutionary system's ability to inexhaustibly generate novel frameworks. Yet an artificial system of this kind remains elusive. At the time of writing nobody has convincingly demonstrated open-ended evolution in software.

Whilst software-based open-ended evolution eludes us, some of the alternative strategies mentioned have shown a little immediate promise for generative art. In particular, artificial evolution of imagery, especially when assisted by human aesthetic assessment of fitness, has received much attention. In light of this, our creativity measure has been investigated as a means for automatically guiding the evolution of simple line drawings (Kowaliw et al. 2009). This forms part of an effort to improve the automatic and assisted creativity of evolutionary software. Our definition is, in practice, computationally intensive to apply. Hence, for practical reasons, Kowaliw devised *creativity-lite*, a modification that is tractable and testable against the full implementation of our measure. The output produced by the creativity-lite measure has been tested against common-language understanding of the term *creativity* by allowing users to evaluate automatically generated textural images (Kowaliw et al. 2012). The results of this research are summarised below.

13.8.1 The Automatic Generation of Creative Biomorphs

In order to apply our definition to a practical task, Kowaliw et al. elected to generate stick figures based on the biomorphs of Dawkins (1989) and see if they could automatically distinguish between those that were creative, and those that were not. Biomorphs are intricate line drawings made of straight segments that may be overlaid upon one another hundreds of times to generate rich patterns reminiscent of trees, insects, snowflakes, stars or countless other forms. Each individual biomorph can be considered, potentially, a creative generative system, since it is capable of

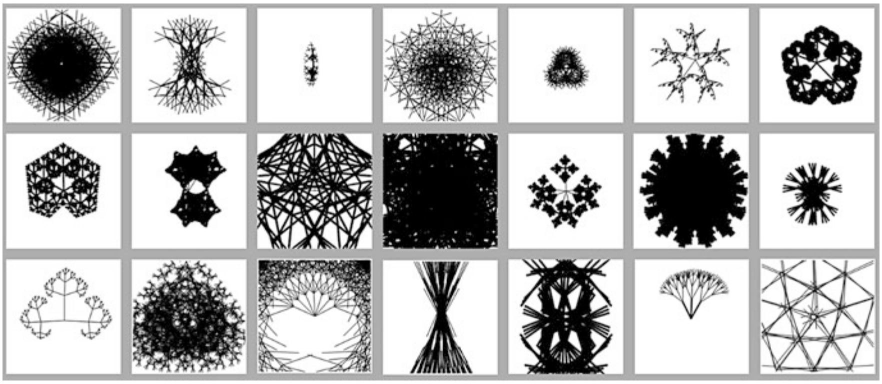


Fig. 13.3 A selection of randomly generated biomorphs generated by the software of Kowaliw, Dorin and McCormack, chosen to show some of the variety that is possible

producing a series of patterns—its offspring. Offspring are generated by varying some of a parent biomorph's parameters slightly, altering the placement, length and orientation of line segments with respect to those of that parent. A child will then be similar, but not identical to, its immediate ancestor. If the forms of a biomorph's offspring are reliably different to those produced by any that have preceded it, then according to our definition of creativity, the parent is creative with respect to its ancestors. If a particular biomorph reliably generates children that are exceedingly unlikely to be generated by any other biomorph (a possibility we can feasibly test in software), we can also claim that the parent is quite generally creative. This approach was adopted in the experiments described below.

Biomorphs were constrained in these tests to appear in 8-bit greyscale images within a 200 square pixel grid. Somewhere in this space of possible images is a representation of every possible human face, including those long dead and those that may never be born, as well as representations of every view ever seen by every person that ever looked upon the world … and that is hardy scratching the surface! In short, this represents an astronomically large space of possible images. Of course not all of these can be generated as biomorphs. Nevertheless, the space remains unimaginably large, even when constrained in this way. An assortment of the images the software can generate is illustrated in Fig. 13.3.

The images the software generated were measured with a set of image-processing techniques that detect traits perceptible to humans. These related to the images' contrast, homogeneity, entropy and other measures of texture. A substantial sample was used to give a representation of the total space of images the software can produce and then three techniques were repeatedly applied: random image generation; an interactive genetic algorithm; and the creativity-lite search employing a simple version of our definition of creativity encoded in software, to try to locate a creative individual biomorph.

As we would hope, randomly created biomorphs did not generate a high proportion of creative offspring at all—the sampling of the space was adequate. When

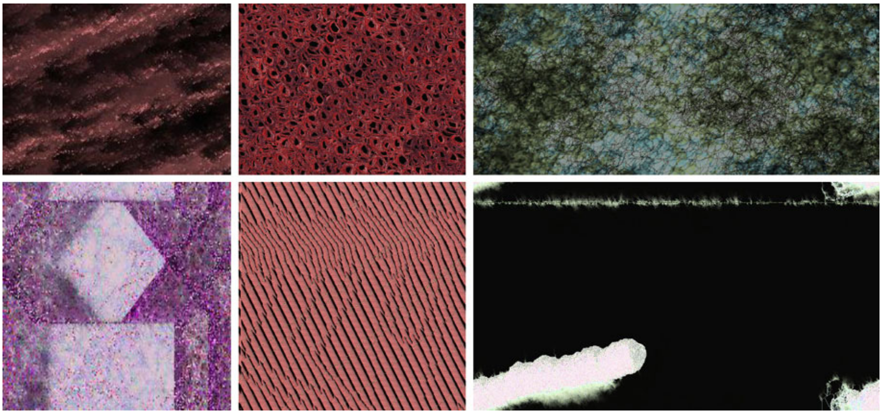


Fig. 13.4 A selection of images evolved by users employing the *EvoEco* texture generating software

creativity-lite generated biomorphs were tested against an encoded complete statistical version of our creativity measure, a very high proportion of the images generated were found to be creative by this measure. Hence it seems fair to conclude that the simplified creativity-lite search is a good approximation of the complete, but intensive, computation required to apply our measure in full. Finally, the interactive algorithm for generating biomorphs consistently produced many more creative individuals than the random technique, but less than the creativity-lite measure. This indicates that our measure goes some way towards representing human preferences. Were the humans evolving for creative results that our measure would appreciate? This test was not set up to answer such questions. How well does our measure correspond to the general human appreciation of creativity? This was explored in a subsequent test, discussed next.

13.8.2 Testing Creative Software Against Human Concepts of Creativity

In order to determine the extent to which our definition of creativity reflected human interpretations of the term, Kowaliw et al. conducted a survey, the details of which can be found in (Kowaliw et al. 2012). Visitors, originally to a gallery installation, and later to a website, were invited to evolve images using texture generating software, *EvoEco*, that operated on the principle of aesthetic selection: users selected a parent image from 16 textures displayed to them. This was then used to generate 16 offspring images as described below, and the process repeated until the user decided it was time to stop. Some sample images generated by users appear in Fig. 13.4.

Of the 16 offspring generated from a user-selected parent image, 15 were generated either by occasional mutation (a small variation in the parameters that generate the image) of the chosen parent's parameters, or by crossing-over the parent's parameters with those of another randomly selected member of the current, or any previous population. Crossover is an operation that samples parameters from one image, complementary parameters from another, and recombines these to produce a complete set of parameters for generating, in this instance, a new image. The new image typically exhibits a combination of traits of its parents.

For a particular user the final offspring image of the 16 was consistently generated either completely at random; or in such a way as to maximise its distance from the parent (according to the statistical image properties applied to the biomorph images above); or, by locating an individual that maximised the creativity-lite measure discussed in relation to the biomorphs. In the latter two cases a sample of images was generated from the parent by mutation and the one that maximised the distance or creativity-lite measure as appropriate was chosen to fill the final slot. The offspring were always positioned in random order on the screen so as to avoid user selection biases introduced by image placement.

Following their engagement with the software users were presented with a survey asking them to rank how appealing, novel, interesting or creative they found the intermediate images and final result.

The distance technique for generating offspring, generally, appeared to be an impediment to the software's success. This technique was ranked somewhat worse than the random generation technique with regard to the novelty of the images it suggested, and significantly worse than the creativity technique in three of six scales we recorded. This performance matched our intuition that maximising distance pushes populations away from the direction selected by users, undoing the effects of evolution. The creativity technique significantly outperformed the distance technique and rated better than the random technique in novelty of the final image, and the creativity of the suggested intermediate images. The research also found that the mean score for all responses was best for the creativity technique; and that the proportion of users who answered positively to the question "Did you feel that you could control the quality of the images through your selections?" was higher for creativity (55 %) than for the other two techniques (31 % for random and 40 % for distance). Hence, we can conclude tentatively that the use of the creativity-lite measure improved the performance of the interactive algorithm with respect to natural language notions of novelty and creativity.

13.9 Discussion

Where to from here? *EvoEco* is not the be-all and end-all of automated creativity. For starters, the software will never be able to step outside of its specification to generate the kind of novelty we would expect a human to devise. For instance, it can never generate a three-dimensional model of a tree. Is this significant? A hard-coded range of representations limits all systems we devise. This is inescapable. Can a digital machine exceed the expectations of its developers nevertheless? We can see no reason why not, but it must do so within the confines of its ability to manipulate representations. Thus, its creativity will be limited, perhaps to the point where we are insufficiently engaged with its behaviour to claim it as a rival to our own creativity.

Human creativity is analogously confined by physical, chemical, biological and social constraints. In keeping with both Boden's definition and our own, we find human creativity interesting because it redefines the boundaries of what is possible within the constraints we believe are imposed upon us. If we can build generative systems that interact within the same constraints as ourselves, we can potentially circumvent the limitations of the purely digital system. And this is what artists do when they map the digital world to real textual, sonic and visual events. A program running on a computer is not very interesting unless we can experience what it is doing somehow. Once we can, the discrete events—changing values, production of spatio-temporal patterns—take on new meanings for those observers that perceive them, outside the internal semantics of the generative system that created them. For this reason we can see no reason why machine creativity cannot in theory rival our own.

How will we know that our creativity has been rivalled? In the usual ways: machine-generated works will appear in the best galleries, win the most prestigious art prizes, sell for vast sums of money, be acclaimed by critics, cited in publications and referenced by other artists. But also, if we are to progress in any methodical sense, because we shall be able to assess their creativity in an objective way, employing measures such as that implicit in the new definition of creativity that we have presented.

13.10 Conclusions

We have described in this chapter a way to interpret creativity independently of informal concepts such as value and appropriateness. These concepts, we feel, have encumbered the search, production and recognition of autonomous creative software to date. Instead, we have proposed a testable measure for creativity that can be applied wherever it is possible to sample and measure the novelty of designs selected from a large, even infinite, space of possibilities. In fact our approach is, in theory, applicable to any digital computational generative system and its output. As a proof of concept, our definition has been encoded in software and employed to measure the creativity of images, enabling it to automatically offer results that humans perceive to be more creative than those made using similar, but differently assisted, algorithmic techniques.

We do not pretend to have shelled the "automated creativity nut", but we believe we have caused a fissure from which we can pry it open. No doubt our powers of persuasion are not so strong as to ensure that our ideas will be uncontroversial, particularly as our suggestions run against the grain of much that has been written previously. Nevertheless, we are at this stage convinced of the merit of our definitions and their application, and hopeful that the problems of building autonomous creative systems can be tackled from the approaches we offer.

Acknowledgements This research was supported by Australian Research Council Discovery Grants DP0772667 and DP1094064.

References

Albert, R. S., & Runco, M. A. (1999). A history of research on creativity. In R. J. Sternberg (Ed.), *Handbook of creativity* (pp. 16–31). New York: Cambridge University Press. Chapter 2.

Bada, J. L., & Lazcano, A. (2002). Some like it hot, but not the first biomolecules. *Science*, *296*, 1982–1983.

Ball, P. (2001). *The self-made tapestry: pattern formation in nature*. Oxford: Oxford University Press.

Barrass, T. (2006). Soma (self-organising map ants). In *EvoMUSART2006, 4th European workshop on evolutionary music and art*.

Bentley, P. J. (2002). Is evolution creative? In P. J. Bentley & D. Corne (Eds.), *Creative evolutionary systems* (pp. 55–62). San Mateo: Morgan Kaufmann.

Berlyne, D. E. (1960). *Novelty, uncertainty, conflict, complexity*. *McGraw-Hill series in psychology* (pp. 18–44). New York: McGraw-Hill. Chapter 2.

Berry, R., Rungsarityotin, W., Dorin, A., Dahlstedt, P., & Haw, C. (2001). Unfinished symphonies—songs of 3.5 worlds. In E. Bilotta, E. R. Miranda, P. Pantano & P. M. Todd (Eds.), *Workshop on artificial life models for musical applications, sixth European conference on artificial life* (pp. 51–64). Editoriale Bios.

Boden, M. A. (2004). *The creative mind, myths and mechanisms* (2nd ed.). London: Routledge.

Borel, É. (1913). Mécanique statistique et irréversibilité. *Journal of Physics*, *3*(5), 189–196.

Borgia, G. (1995). Complex male display and female choice in the spotted bowerbird: specialized functions for different bower decorations. *Animal Behaviour*, *49*, 1291–1301.

Carroll, J. (2007). The adaptive function of literature. In C. Martindale, P. Locher & V. M. Petrov (Eds.), *Evolutionary and neurocognitive approaches to aesthetics, creativity and the arts* (pp. 31–45). Amityville: Baywood Publishing Company. Chapter 3.

Chamberlain, W. (1984). Getting a computer to write about itself. In S. Ditlea (Ed.), *Digital Deli, the comprehensive, user-lovable menu of computer lore, culure, lifestyles and fancy*. New York: Workman Publishing. http://www.atariarchives.org/deli/write_about_itself.php. Accessed 21 October 2010.

Colton, S. (2001). *Automated theory formation in pure mathematics*. PhD thesis, University of Edinburgh.

Colton, S., Bundy, A., & Walsh, T. (2000). On the notion of interestingness in automated mathematical discovery. *International Journal of Human Computer Studies*, *53*(3), 351–375.

Csikszentmihalyi, M. (1999). Creativity. In R. A. Wilson & F. C. Keil (Eds.), *The MIT encyclopaedia of the cognitive sciences* (pp. 205–206). Cambridge: MIT Press.

Dahlstedt, P. (1999). Living melodies: coevolution of sonic communication. In A. Dorin & J. McCormack (Eds.), *First iteration: a conference on generative systems in the electronic arts* (pp. 56–66). Melbourne: CEMA.

Dawkins, R. (1989). The evolution of evolvability. In C. G. Langton (Ed.), *Artificial life: proceedings of an interdisciplinary workshop on the synthesis and simulation of living systems* (pp. 201–220). Reading: Addison-Wesley.

Dorin, A. (2004). The virtual ecosystem as generative electronic art. In G. R. Raidl, S. Cagnoni, J. Branke & D. W. Corne (Eds.), *2nd European workshop on evolutionary music and art, applications of evolutionary computing: Evo workshops*, Portugal (pp. 467–476). Berlin: Springer.

Dorin, A., & Korb, K. B. (2009). A new definition of creativity. In K. Korb, M. Randall & T. Hendtlass (Eds.), *LNAI: Vol. 5865*. *Fourth Australian conference on artificial life*, Melbourne, Australia (pp. 1–10). Berlin: Springer.

Earnshaw, C. G. (1991). *Mozart's Dice Waltz*. Atari users. http://tamw.atari-users.net/mozart.htm. Retrieved 21 October 2010.

Eddington, A. S. (1927). *The nature of the physical world: Gifford lectures*. New York: Macmillan.

Eldridge, A. (2009). *You pretty little flocker*. Guildford Lane Gallery.

Gracia, J. J. E. (1996). *Texts: ontological status, identity, author, audience*. New York: State University of New York Press.

Harnad, S. (1990). The symbol grounding problem. *Physica D*, *42*(1–3), 335–346.

Hofstadter, D. (1995). *Fluid concepts and creative analogies, computer models of the fundamental mechanisms of thought*. Harper Collins: Basic Books.

Johnson-Laird, P. (1987). Reasoning, imagining and creating. *Bulletin of the British Psychological Society*, *40*, 121–129.

Joyce, G. F. (1989). RNA evolution and the origins of life. *Nature*, *338*, 217–224.

Kowaliw, T., Dorin, A., & McCormack, J. (2009). An empirical exploration of a definition of creative novelty for generative art. In K. Korb, M. Randall & T. Hendtlass (Eds.), *LNAI: Vol. 5865. Fourth Australian conference on artificial life (ACAL 2009)*, Melbourne, Australia (pp. 1–10). Berlin: Springer.

Kowaliw, T., Dorin, A., & McCormack, J. (2012). Promoting creative design in interactive evolutionary computation. *IEEE Transactions on Evolutionary Computation*, *16*(4), 523–536.

Langley, P., Simon, H. A., Bradshaw, G. L., & Zytkow, J. M. (1987). *Scientific discovery: computational explorations of the creative processes*. Cambridge: MIT Press.

Lenat, D. B. (1983). Eurisko: a program that learns new heuristics and domain concepts: the nature of heuristics III: program design and results. *Artificial Intelligence*, *21*, 61–98.

Maynard-Smith, J., & Szathmry, E. (1995). *The major transitions in evolution*. New York: Freeman.

McCormack, J. (2001). Eden: an evolutionary sonic ecosystem. In J. Kelemen & P. Sosik (Eds.), *LNAI: Vol. 2159. Advances in artificial life, 6th European conference, ECAL*, Prague, Czech Republic (pp. 133–142). Berlin: Springer.

Miller, G. F. (2001). Aesthetic fitness: how sexual selection shaped artistic virtuosity as a fitness indicator and aesthetic preferences as mate choice criteria. *Bulletin of Psychology and the Arts. Special issue on evolution, creativity, and aesthetics*, *2*(1), 20–25.

Nake, F. (2002). Personal recollections of a distant beginning. In L. Candy & E. A. Edmonds (Eds.), *Explorations in art and technology* (p. 6). London: Springer.

Odling-Smee, F. J., Laland, K. N., & Feldman, M. W. (2003). *Niche construction, the neglected process in evolution*. *Monographs population biology*. Princeton: Princeton University Press.

Phillips, J. (1935). Succession, development, the climax, and the complex organism: an analysis of concepts: part III. The complex organism: conclusions. *Journal of Ecology*, *23*(2), 488–508.

Putnam, H. (1979). *The meaning of "meaning", philosophical papers: mind, language and reality* (Vol. 2, pp. 215–271). Cambridge: Cambridge University Press. Chapter 12.

Rowe, J., & Partridge, D. (1993). Creativity: a survey of AI approaches. *Artificial Intelligence Review*, *7*, 43–70.

Saunders, R., & Gero, J. S. (2002). Curious agents and situated design evaluations. In J. S. Gero & F. M. T. Brazier (Eds.), *Agents in design* (pp. 133–149). Key Centre of Design Computing and Cognition, University of Sydney.

Schmidhuber, J. (2009). *Driven by compression progress: a simple principle explains essential aspects of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes*. arXiv:0812.4360v2 [cs.AI].

Sims, K. (1991). Artificial evolution for computer graphics. *Computer Graphics*, *25*(4), 319–328.

Smuts, J. C. (1927). *Holism and evolution*. London: Macmillan and Co.

Tatarkiewicz, W. (1980). *A history of six ideas, an essay in aesthetics*. Warsaw: PWN Polish Scientific Publishers.

Todd, S., & Latham, W. (1992). *Evolutionary art and computers*. San Diego: Academic Press.

Weesatchanam, A.-M. (2006). Are paintings by elephants really art? www.elephantartgallery.com. Retrieved 31 July 2009.

Chapter 14
Generative and Adaptive Creativity: A Unified Approach to Creativity in Nature, Humans and Machines

Oliver Bown

Abstract Computational creativity is not limited to the study of human-like creativity and forces us to think about creativity as a general process that can be applied wherever new things come into existence. In this chapter I propose that in order to unify various forms of creativity it is necessary to consider a distinction between two types of creativity: generative creativity, in which things are created as the result of a process regardless of their value, and adaptive creativity, in which things are created as adaptive responses by a system to its situation. Whilst individual human creativity is typically of the adaptive form, collectively humans are engaged in processes of generative creativity as well as adaptive creativity. It is helpful to understand human creative behaviour as part of a social process involving these two aspects, and this is relevant to understanding how manmade artefacts can act as creative agents in social networks.

14.1 Questions About Creativity

Theories of creativity are so commonly focused on human behaviour that for many researchers there is no need to address notions of creativity outside of the frame of reference of human psychology. In the disciplines centred around psychology this is appropriate. As part of the cognitive sciences, it is also reasonable for artificial intelligence (AI) to stick to this delimitation of creativity as an activity. The study of computational creativity, as demonstrated by the range of contributions to this book, is very different in that it poses scenarios which provoke us to view creativity in significantly more varied terms—scenarios, for example, where the computer acts as a generator of variation, heavily mediated by its user.

To this end, human psychology is a limited reference point. Computational creativity is the study of creativity by any computational means, not necessarily those modelled on human minds, or even on human goals, and as such the discipline takes on the challenge of developing a more fundamental understanding of what it means

O. Bown (✉)
Design Lab, Faculty of Architecture, Design and Planning, University of Sydney, Sydney, NSW 2006, Australia
e-mail: ollie@icarus.nu

J. McCormack, M. d'Inverno (eds.), *Computers and Creativity*,
DOI 10.1007/978-3-642-31727-9_14, © Springer-Verlag Berlin Heidelberg 2012

for something to create, and of where and when a creative process has occurred. In taking this broad stance on creativity, computational creativity necessarily concerns itself with acts of creation wherever they occur, not just in humans. We require a broader view of creativity as the process of creating novel things, not limited to a suite of psychological capacities. This richer notion of creativity emerges alongside practical innovations in our research and feeds back into informing an understanding of creativity in humans and elsewhere.

The most prevalent example of non-human creativity is the Earth's history of natural evolution,[1] associated with an early sense of the term "creativity", as in *creationism*, the exclusive remit of God (Williams 1983). Whether through Nature or God, the biological world, including us, stands as evidence of dramatic creativity.

In a broader sense still, computers routinely, mundanely, create things. An elementary type of creativity can be achieved by the rapid, random production of varieties from a set of generative rules. In the simplest case, this can be a set of parameter specifications that can be assigned different values to produce different outputs. These things are creations: new entities that would not otherwise have existed. This is creativity with a crucial caveat: somebody or something *else* has to come up with the generative rules. A pragmatic choice for very simple experiments in computational creativity, for example, is the spirograph, in which a set of gear ratios can be used to define a space of non-trivial visual patterns (Saunders 2001). Obviously, all you get from such a system is spirographs, pretty patterns with a minuscule potential to deliver something truly surprising. Equally, one could use a computer to begin to search the vast parameter space of all 500 by 500 pixel 24-bit colour images, an example discussed by McCormack (2008). One could proffer that there are still phenomenal 500 by 500 pixel colour images yet to be seen by a human eye, but the space to search is so vast that any naïve search has no certainty of finding them. With generative techniques, new things *can* be created that have never been created before, trivial though this may seem. This "generative creativity", in its simplest form, is a particularly weak form of creativity, but by such accounts it still seems to be creativity.

These examples of non-human creativity are, in quite different ways, distinct from the human cognitive activity associated with *being creative*, but they are both important to the study of computational creativity. This chapter uses the terms generative and adaptive creativity to help unify these diverse manifestations of the act of creating things into a common theory.

The structure of this chapter is as follows: I will explain the meaning of generative and adaptive creativity in the following section. In Sect. 14.3, I will discuss the relevance of generative and adaptive creativity to current research in computational creativity. I will then consider social systems firstly as creative systems that are more than just the sum of human creative acts, and then as both adaptively creative units

[1]Discussions on this topic can be found in the chapters by Cariani (Chap. 15) and McCormack (Chap. 2) in this book, in earlier work in computational creativity by, for example, Bentley (1999b), Perkins (1996) and Thornton (2007), and in more remote areas of study such as Bergson (1998) and de Landa (1991).

and as generatively creative units. I will consider examples from computational creativity research of modelling social creative systems. I will also consider individual humans as exhibiting generative as well as adaptive creativity. Finally, I will return to computational creativity (Sect. 14.4) and consider how ideas of generative and adaptive creativity can be used to support developments in computational creativity research.

14.2 Generative and Adaptive Creativity

From the broad perspective of poiesis—of how things come about—all the patterns, structures and behaviours that exist in the world can be taken as evidence of creativity. This jars with the traditional psychological view of creativity, and implies a distinction between two varieties, which I will refer to in this chapter as *generative* and *adaptive*. Generative creativity takes an indifferent approach to the problem of value, it is value-free creativity. In generative creativity, things are not created for a purpose. Things *can* come into existence without being created for their value.

The mechanical variation of the spirograph discussed above may be the least ambiguous, albeit banal, example of generative creativity. Natural evolution is a more impressive example, but more contentious because the relationship between nature's creative processes and the production of value is complex. Natural evolution provides means for lineages of organisms to adapt to their environments, but it is also responsible for producing both evolutionary challenges and their evolutionary solutions together in tandem, in the absence of an ultimate goal. Peacocks' tails are useful to peacocks, and the advancement of their progeny, because they are attractive to peahens, but *this utility is there for the genetic lineages of peacocks and peahens, it does not serve the process of evolution that produced them.*

Value (survival value, the value of sexual attractiveness) is part of the equation that feeds these evolutionary processes, but the creative processes that produced the peacock's tail is not, in this author's opinion, an adaptively creative process, as I will define it below.

This view may yet be mistaken: Gaia theory, for example, implies that there is a general process of improvement driven by evolution (Lovelock 1979). I may also be underselling the true value of peacocks' tails, in light of the handicap principle (Zahavi 1975) or honest signalling theory (Owings and Morton 1998). With these and other complex issues in evolutionary theory in mind, readers may understandably take the opposite view that natural evolution is not indifferent to value and is thus adaptively creative, as defined below. Indeed, the definitions below do not preclude the possibility of evolutionary adaptation providing examples of adaptive creativity. The view offered here, though, is that generative creativity is the more predominant aspect of natural evolution: whilst valuable functions are established during the evolutionary process, the subject (and beneficiary) of the value—the peacock for example—is not the creative agent behind the trait, and the process that, if anything, is the creative agent—an abstract evolutionary mechanism—is a non-entity as far as "benefits" are concerned.

Put more bluntly, a once barren planet now teems with life. There is no function that this life was brought into existence to perform. This is generative creativity.

Adaptive creativity is concerned with the process of creating things as an adaptive behaviour, perhaps one of a suite of adaptive behaviours, exhibited by a system (person, animal, computer, social group, etc.). The least ambiguous example of adaptive creativity would be everyday problem solving, where an individual finds a new way to directly improve their life. Basic problem solving is widely observed in the animal kingdom; a plausible explanation for this is that a suite of cognitive faculties evolved through selective advantage. Through networks of exchange, humans can also benefit from inventions that are not directly useful to them. The arts, with which this chapter is primarily concerned, are more controversial in this respect: many would question the relationship between art and adaptive behaviour. But the less controversial association between art and value is reason enough to place artistic creativity in this category. It would seem fair to say that artists generally benefit from the artworks they produce, even if we are not sure why. Creating something of value, albeit in a convoluted socially constructed context, is, I will assume, an adaptive behaviour.

There is a lot to unpack from these preliminary remarks, which I will do throughout this chapter. To begin, generative creativity and adaptive creativity will be defined as follows:

- *Generative Creativity*: an instance of a system creating new patterns or behaviours regardless of the benefit to that system. There is an explanation for the creative outcome, but not a reason.
- *Adaptive Creativity*: an instance of a system creating new patterns or behaviours to the benefit of that system. The creative outcome can be explained in terms of its ability to satisfy a function.

Adaptive creativity is here intended to describe the familiar understanding of human creativity as a cognitive capacity. Generative creativity is its more mysterious counterpart, and helps extend the scope of creativity to cover a greater set of creations. Although this duality could be represented simply as novelty with or without value, the terms are taken to emphasise two essentially different characters that creativity can assume.

Generative creativity may seem like a distraction from the problems of understanding human creativity, and of establishing human-like creativity in computational systems. The main aim of this chapter is to argue that the duality of generative and adaptive creativity is instead highly relevant to our understanding of human creativity, offering a framework with which to understand individual and distributed social creative processes in a common extensible and future-proof way. The generative/adaptive divide is argued to be as useful as the human/non-human divide, and is different in significant ways. This picks up the cause of emphasising how important the social dimensions of human creativity are, following socially-oriented theories such as those of Csikszentmihalyi (1999), but since this cause is already well established, the more relevant goal is a framework that unifies these elements.

I have briefly discussed natural evolution above, and its ambiguous relationship to adaptive creativity. A similar discussion will be applied to culture in greater depth.

In the following sections, I will discuss how the notions of generative and adaptive creativity apply in various instances associated with human social behaviour. In doing so I will draw on concepts from the social sciences which I believe can enrich the conceptual foundations of computational creativity. The discussion is geared towards arts-based computational creativity, but will draw from scenarios outside of the arts.

14.3 Generative and Adaptive Creativity in the Arts, in Humans, Human Groups and in Silico

The generative/adaptive framework can be applied to the way we characterise the creativity of artificial systems in the poorly understood domain of human artistic behaviour. Despite some clear achievements in arts-based computational creativity, success has been marred by the challenges of evaluating the products of creative systems in a meaningful way (Pearce and Wiggins 2001, Greenfield 2006, McCormack 2008). We struggle to disambiguate the creative input of the system's designer from the creative output of the system itself (Greenfield 2006), and we also struggle to establish valid contexts for either subjective or objective judgement of creative outputs. This ambiguity has made arts-based computational creativity particularly resistant to the bootstrapping of future developments from previous success. It is hard to make informed decisions about the relative virtues of different computational methods (genetic algorithms, neural networks, etc.) because arts-based computational creativity research lives mainly in the lab, with many additional steps required to get from this to the real field of human artistic activity. This suggests the need for a richer characterisation of creativity, including the role of social dynamics as a creative system, as well as the role of creative individuals within that system.

Above all, in labs, goals are implicitly defined for arts-based computationally creative systems which have ill-defined parallels in the world of human artistic goals. A person has a flexible and creative relationship with the value of the cultural artefacts he creates and is surrounded by, for example by being free *not* to be artistically creative. As currently conceived, an arts-based computationally creative system is, by contrast, a tool with a function assigned to it. It can only be adaptively creative within the limits imposed by its given function (it must make art). Functionality is a desirable property of all manmade systems. An artificial creative system is expected to do more than generative creativity (an end in itself), because it is required to produce outputs which are not just novel but have some externally established value. As such, arts-based computationally creative systems target a novel niche, in terms of their social embeddedness, which is yet to be clearly characterised. There is a need to establish a discourse that properly recognises this niche.

What, then, is the nature of natural creative systems in terms of function? To determine what kind of status a creative tool could take, it is important to look at human creativity not only at the individual level: individuals in isolation cannot provide us with an understanding of the creative nature of the arts because a complete understanding involves identifying how value is generated within a social system (Csikszentmihalyi 1996). I will expand this point with reference to relevant perspectives in the social sciences, as it is an important prerequisite for considering how social systems and individuals can be generatively and adaptively creative.

14.3.1 The Creativity of Social Systems Is More than the Sum of Individual Creative Acts

Our creative capabilities are contingent on the objects and infrastructure available to us, which help us achieve individual goals. One way to look at this is, as Clark (2003) does, in terms of the mind being *extended* to a distributed system with an embodied brain at the centre, and surrounded by various other tools, from digits to digital computers. We can even step away from the centrality of human brains altogether and consider social complexes as distributed systems involving more or less cognitive elements. Latour (1993) considers various such complexes as *actor networks*, arguing that what we think of as agency needs to be a flexible concept, as applicable to such networks as to individuals under the right circumstances.

Gell (1998), proposing an anthropology of art, likewise steps away from the centrality of human action by designating primary and secondary forms of agency. Artefacts are clearly not agents in the sense that people are, that is, *primary agents*, but they can be granted a form of agency, *secondary agency*, to account for the effect they have on people in interaction. Gell argues that we need this category of secondary agency, which at first seems counterintuitive, in order to fully account for the networks of interaction that underpin systems of art. Artworks, he argues, abduct agency from their makers, extended to new social interactions where they must necessarily be understood as independent. This idea of art as an extension of influence beyond the direct action of the individual is also emphasised by Coe (2003) as key to understanding the extent of kin-group cohesion in human societies: by extending influence through time, such as through decorative styles, an ancestor can establish strong ties between larger groups of descendants than otherwise would be possible.

It is hard to be precise about exactly what is meant by artworks and artefacts here. For example, decorative styles are in a sense just concepts, in that they can only be reproduced through individual cognition, and the same is true of artefacts if they are to be reproduced and modified in continued lineages (Sperber 2000)—a situation that would radically change with self-reproducing machines. But artefacts are concepts built around a physical realisation, which is itself a carrier for the concept. That is, objects participate in the collective process of remembering and learning that allow a culture to persevere and evolve over time.

Artworks can take on a slightly different status. They are typically defined as unreproducable, even if they are effectively reproduced in many ways (Gell 1998). Like many other artefacts involved in social interaction, such as telephones and social networking tools, they act to shape a distributed creative process. These may be the products of individual human invention, but they do not simply get added to a growing stockpile of passive resources, but instead join the process as secondary agents (Gell 1998). Put another way, when culture is seen as a creative system, it is inclusive of all of these objects, which contribute functionally to the system, just as the distributed modular mechanisms of the human brain may be required to be in place for individual creative cognition to function. Similarly, a memetic view of culture sees ideas, concepts, designs and so on as not just cumulative but effective, a new form of evolutionary *raison d'être* added to the biological world (Dawkins 1976). These "memes" are understood as having an emergent teleological functionality: brains evolved under selective pressures, and memes were thus unleashed. But what are memes for? The answer: they are only for memes. Sperber (2007) provides a strong argument for rejecting memes, but promotes the idea of distinguishing between a designated function (a function we ascribe to something) and a teleofunction (a function of something in an evolutionary sense, in service of its own survival).

Just as neuroscientists care about the behaviour of synapses as much as they do neurons, a theory of social creativity depends on the functional importance of both primary and secondary agents. We can view any digital arts tool as a secondary agent, but arts-based computational creativity holds the promise of introducing secondary agents that are richly interactive, and as such creatively potent (if not adaptively creative), encroaching on the territory of primary agents. Arts-based computational creativity researchers, by definition, study the possibility of artefacts with agency, and in doing so reveal a gradient of agency rather than a categorical division.

The *Interactive Genetic Algorithm* (IGA) (Dawkins 1986), for example, is an artificial evolutionary system in which a user selectively "breeds" aesthetic artefacts of some sort (see Takagi 2001 for a survey), or manipulates an evolutionary outcome via a user-defined fitness function (e.g. Sims 1994, Bentley 1999a). The IGA can only possibly achieve adaptive creativity by being coupled with a human user, in a generate-and-test cycle. However, it allows the user to explore new patterns or behaviours beyond those he would have devised using imagination or existing forms of experimentation (Bentley 1999b). As such, it is not autonomous, and yet it is active and participatory, grounded in an external system of value through a human user.

Researchers in IGAs continue to struggle to find powerful genetic representations of aesthetic patterns and behaviour that could lead to interesting creative discovery (e.g. Stanley and Miikkulainen 2004). But more recently, IGAs have also be used to couple multiple users together in a shared distributed search (Secretan et al. 2008). Whilst an individual approach views IGAs as creative tools that extend individual cognition, as in the extended mind model, the distributed notion of an IGA embodies a social view in which no one mind can be seen as the centre of an artificially extended system. Instead, minds and machines form a heterogeneous network of interaction, forcing us to view this hybrid human and artificial evolutionary system on a social level. In this and other areas, arts-based computational creativity is well-poised to bootstrap its future development on the emergence of social computing, which presents a training and evaluation environment on the scale of a real human social system.

14.3.2 Social Systems Can Exhibit Both Generative and Adaptive Creativity

To be adaptively creative, social systems must be shown to form coherent units to which creativity is beneficial. Society contains many structures that naturally appear to us as unified wholes, such as families, organisations and nations, even when the individual members of these groups might be interchangeable. In certain cases, these unities act with intention and planning and can be creative in a sense that is more than simply an accumulation of individual creative acts, through the formation of structures which incentivise, intensify and exploit individual creativity. In other circumstances, overlapping with these adaptive scenarios, social groups lack the shared intention required to view them as adaptive units, but nevertheless act as powerful generative systems. In such cases it is not possible to view the system's creativity as serving specific adaptive goals at the level that they are generated. For the groups concerned, treated as units, the system's creativity may be counter-productive, even if it involves adaptive creativity at the individual level. From an evolutionary perspective, this position is no surprise: we do not generally expect groups to sustain collective adaptive behaviour, as they can be easily undermined by selfish individuals exploiting the adaptive properties of the group (Wilson 1975). The important point for defining generative creativity is that there are properties of the system that are necessary for an understanding of how new patterns and behaviours emerge, that are not adaptations by the group, and that go beyond the accumulation of individual adaptive creativity.

14.3.2.1 The Causes and Effects of Culture

As this implies, understanding adaptive and generative creativity in social systems depends on understanding the dynamics of cooperation and competition in human social behaviour. A simple way that adaptive and generative creativity can be seen to be linked through culture is through the following evolutionary explanation:

1. Cultural behaviour arose due to specific evolutionary adaptations in individuals such as the ability to imitate successful behaviour, to understand others' goals, to manipulate behaviour and to be adaptively creative (Barkow et al. 1992);
2. This leads to:
   a. The formation of structured adaptive social units stemming originally from family groups and simple local alliances, in which a collective interest can become established (Fisher 1958, Axelrod 1997, Maynard Smith and Szathmáry 1995, Hamilton 1963); and
   b. The cultural conditions for less cooperative interaction which can have runaway generative effects, especially under the constraints of structured adaptive social units (Boyd and Richerson 1985, Blackmore 1999).

As with the definitions of generative and adaptive creativity, we can try out statements (2a) and (2b) against specific instances of social behaviour. This theoretical outline is most clearly conveyed in Boyd and Richerson's influential body of work (Boyd and Richerson 1985), in which they propose that much cultural behaviour can be thought of as "baggage", unexpected but tolerable spin-offs from the powerful success of applying simple frugal heuristics for social learning, fitting the scenario of (2b). The same pattern is expressed by a number of theorists who share the conviction that an evolutionary explanation does not mean simply seeking a function for each trait and behaviour of the human individual. For example, Pinker (1998) explains music as evolutionary cheesecake: something that is pleasurable and compelling not because it is adaptively useful in itself, but because it combines a number of existing adaptive traits. Being creative, he proposes, we have found new ways to excite the senses, and continue to do so. Thus, controversially, music joins more obviously maladaptive and sinister "inventions", such as drugs and pornography.

At the same time, Boyd and Richerson (1985) posit that social learning can alter individual human evolutionary trajectories by coordinating and homogenising certain aspects of behaviour across a group, counteracting the effect of disruptive selfish behaviour, and explaining how groups of individuals can consolidate collective interests over evolutionary time. This fits the scenario in (2a). Others have explored how the social structures enabled by more complex social behaviours can lead to the evolution of increasingly group-cooperative behaviour. For example, Hagen and Bryant (2003) explain music as a means for groups to demonstrate the strength of their coalition as a fighting unit. The basis for their theory is that since practising complex coordinated behaviour such as dance and music takes time and individual commitment, a well-coordinated performance is a clear—or *honest* in evolutionary terms—indicator of how cohesive the group is. Honest indicators of fighting strength and commitment, like a dog's growl, can serve both parties weighing up their chances in a fight, by indicating the likely outcome through indisputable displays (Zahavi 1975, Krebs and Dawkins 1984, Owings and Morton 1998).

A growing body of research into the relationship between music, early socialisation (such as mother-infant interaction) and group cohesion supports this basic thrust (Dissanayake 2000, Cross 2006, Parncutt 2009, Richman 2001, Merker 2001, Brown 2007), although it may be manifest in alternative ways. Such theories are evolutionarily plausible despite being seemingly "group-selectionist", because they can be understood in terms of kin-selection and honest signalling taken to increasingly extended and complex social structures (e.g. Brown 2007, Parncutt 2009). As Dunbar (1998) has demonstrated, vocal communication in humans may form a natural extension to the kinds of honest signalling of allegiance found amongst primates, correlating with both group size and brain size. These theories are also commensurate with the widely consistent observations of anthropologists that the representation of social structure, such as in totem groups and myths, is a universal cultural preoccupation (Lévi-Strauss 1971, Coe 2003).

14.3.2.2 Social Groups as Adaptive Units

Organised and cohesive social groups frequently coordinate and structure individual creative behaviour, and increasingly so in more complex societies, which are able to incentivise, intensify and exploit creative discovery through the distribution of resources and power (Csikszentmihalyi 1996). Consider an illustrative example from Sobel (1996): at the height of colonial competition a grand challenge for naval science was the discovery of a technique to determine longitude at sea. Whilst latitude could be determined entirely from the height of stars above the horizon, longitude could only be reckoned in this way relative to the time of day, which could not be known accurately using existing clocks. Errors in longitude estimation could be disastrous, costing time, money and lives, and the demand for a solution intensified as the problem of longitude became increasingly pivotal to naval supremacy. A substantial prize was offered by the British government, and the prize stood for many years before being claimed. The solution came from a lone, self-taught clockmaker, John Harrison, who made numerous small innovations to improve clock accuracy.

Of interest here is not only John Harrison himself but the great efforts invested by his competitors in pursuit of the prize money. Some had far flung ideas, others pursued fanciful but reasonable alternatives, others still were clockmakers like Harrison himself, pursuing different techniques. More serious competition came in the form of an astronomical solution which required knowing the future trajectory of the moon for years to come. Together, these disparate groups "collaborated through competition" to discover the solution to the problem of longitude, naturally dividing their efforts between different domains, and giving each other clear ground to occupy a certain region of the search space.

Here, within-group competition was artificially driven by a prize, constrained by certain socially imposed factors: the prize money established a common goal, and awareness of existing research drove specific innovators down divergent pathways, and incentivised outsiders to bring their skills to the challenge. The prize encouraged outsiders with far-flung interests to put effort into a solution, and at no expense to the government. This underlines the difference between a prize, for which only one innovator from the domain gains, and a series of grants for research. The former is indifferent to effort, excellence or even potential and is motivated by uncertainty about where a winning solution might turn up. Like the fitness function in an optimisation algorithm, it cares only for success, and it has a clear means for determining success. The latter invests in potential and uses effort and excellence as indicators of likely success. Both forms of finance played a role in establishing the solution, since Harrison actually received occasional grants from the Board of Longitude to fund his gradual development, indicating that they has some confidence in a clock-based solution. Harrison was once a maverick outsider, drawn by the prize. Through his early promising efforts he became a refined and trusted investigator, deserving of further funding.

Only through the constant jostle of shifting social interaction can this outcome be explained. Historical examples of social creativity such as the Longitude Prize have helped to build our modern world of research councils, music industry major labels and venture capitalism, for example by demonstrating the powerful creative potential of open markets. Harrison, an unlikely outsider to the challenge, was first motivated, then identified as having a chance, then allowed to flourish. The prize also had its losers, whose time and perhaps great talent went unrewarded, wasted in pursuit of a prize they didn't win. Their attempts at individual adaptive creativity may have failed, and yet inadvertently they contributed to the adaptive creativity of some larger social group with their various negative results. That is not to say they were duped or that they acted maladaptively. Many modern professionals, such as architects and academics, compete against challenging odds to get coveted funding or commissions. Most find they can reapply their efforts elsewhere, which is in itself a creative skill.

It seems plausible that this kind of competitive dynamic also has an inherently self-maintaining structure: those who are successful, and therefore able to impose greater influence on future generations, may behave in such as way as to reinforce the principles of competition in which they were successful. A prize winner may speak in later years of the great social value of the prize. Those who are successful at working their way up in organisations might be likely to favour the structures that led to their success, and may try to consolidate them.

In other cases, the emergence of new social structures or the technologies that underpin new social arrangements, innovated by various means, may act to the detriment of individuals. An example is the innovation of agriculture as presented by Diamond (1992), which was a successful social organisation because it enabled the formation of larger centralised social groups with a greater division of labour, despite worsening the diet of the average individual.

14.3.2.3 Social Groups as Non-adaptive Generators

According to the idea that cultural behaviour attracts baggage—runaway cultural patterns of behaviour (Boyd and Richerson 1985)—the same mechanisms of incentivisation can occur in generative creative processes, that is, in situations in which the collective system is not behaving adaptively in sight of a goal. For example, whether or not music or the arts are valuable to social groups, individuals adaptively pursue goals as musicians or artists (Huron 2001 provides a non-Western example), and in doing so change the world of arts as a whole over time. The change itself need not necessarily be the result of individual innovation. Although we have a taste for novelty, artistic behaviour is also constrained by conservative forces: musicians and artists are compelled to work within a style, and success is by no means proportional to the degree of novelty of the producer (Boden 1990, Csikszentmihalyi 1996, Martindale 1990). Thus it cannot be taken as given that the explanation for variation in the arts comes down only to individual creative innovation.

A musical fad, for example, is characterised by the explosion of interest in a radical new style. When that explosion occurs, individuals from diverse backgrounds may redirect the skills they have nurtured elsewhere to this domain (a derogatory expression for which is "jumping on the bandwagon"). This results in novel music, but it is the cultural process—the rapid spread of a fad through a population—that actually underlies the processes of exploration and combination that contribute to a creative outcome, not the individual creative capacities of individuals to innovate successful solutions to an artistic goal. Individuals may actually be acting not innovatively, but identically and predictably: applying existing habits and background knowledge to a new domain, and engaging in something of a lottery over the future direction of musical style. Indeed, musical change over decades may be less to do with innovation than to do with waves of individuals, generations, restructuring musical relevance according to their own world view, involving a combination of group collaboration and within-group and between-group competition. Hargreaves (1986) considers such fashion cycles in the social psychology of music. Fads offer an indication of how creative change at the social level can occur as a combinatoric process built on gradual mutation and simple individual behaviour, and can only be understood at that level. The negative connotations of a fad as ephemeral and ultimately inconsequential emphasise the generatively creative nature of this process: a fad satisfies no goal at the level on which it occurs, although many individuals may be satisfying individual goals in the making of that process.

The nature of the arts both with respect to adaptive human social behaviour, and as a collective dynamical system, is becoming better understood, but sociologists and anthropologists have struggled with good reason to develop a solid theoretical framework for such processes, and we still have far to go before we can disentangle adaptive and generative aspects of social artistic creativity.

14.3.2.4 Modelling Creativity in Social Systems

Strands of arts-based computational creativity research have focused on generative aspects of social systems and their relationship to individual adaptive creativity. Gero, Sosa and Saunders have explored a large space of social models of design creativity in which individuals collectively define the social conditions in which they both produce and judge creative artefacts (Saunders 2001, Saunders and Gero 2001, Sosa and Gero 2003). Saunders and Gero (2001), for example, demonstrates clique formation through mutual influence and learning of specialised interest, suggesting a generative creative process in which a population of agents spawn novel styles through a group dynamic. Such models often establish the necessary conditions for generative creativity by establishing that what determines fitness is not an external environment but the population itself through a process of feedback (Laland et al. 1999, Bown and Wiggins 2005), a fundamental consideration in evolutionary psychology (Dunbar 1998, Tomasello 1999), and a property of other generatively creative processes in nature, such as sexual selection (Miller 2000) and niche construction (Odling-Smee et al. 2003).

I have adapted such models to an evolutionary context in order to explore the potential influence of generative social dynamics on evolutionary change (Bown 2008). Pinker (1998) uses "evolutionary cheesecake" as a description of music inviting the question of whether such a cultural development might actually become reinforced, and thus biologically locked-in, through evolutionary adaptations: if musical behaviour becomes adaptively beneficial to individuals through its increasing prominence in social life—which is the implication of Pinker's hypothesis—then we could ask whether humans have even evolved to become more musical under constructed social pressures. The resulting model illustrated that this reinforcement could happen through kin selection exploiting social interaction "games" in which individuals rewarded each other with prestige. According to this model, it isn't even necessary to assume that music appeared at first as a culturally innovated *susceptibility to enchantment* (Bown 2008), since the susceptibility itself could be seen to emerge as a result of the social dynamics.

Such models can in some cases provide a proof-of-concept for mechanisms of evolution and social change. However, they necessarily remain abstract and far removed from attempts to conduct predictive modelling of social dynamics (Gilbert 1993).

14.3.3 Individual Humans Can Exhibit Generative and Adaptive Creativity

It is reasonable to accept the assumption that individual human creativity is strictly of the adaptive type. Darwinian evolutionary theory predicts that as evolved organisms, most of our behaviour is ruthlessly adaptive, at least adapted with respect to some evolutionary environment of the past (Wilson 1975, Dawkins 1976). But there are reasons why a human's behaviour could also fit the description of generative creativity instead. For an individual to be generatively creative, this would mean that they generate and sustain novel patterns or behaviours without any regard to the externally determined value of these patterns or behaviours. If art was only about innovation of artefacts designed to stimulate other people for individual gain, this would be an unlikely pattern of behaviour. But consider art as a multi-faceted cultural complex, involving elements such as identity. In service of an identity, an individual might generatively create effectively arbitrary patterns or behaviour, the value of which then come through association with other properties of the individual.

Coe (2003) proposes visual decoration as a mechanism for identifying the codescendants of a common ancestor in the traditional small-scale societies of our evolutionary history. Similarly decorated individuals are both common genetic descendants and, more importantly, cultural descendants of an ancestral figure, inheriting and thus preserving styles and art techniques originated by that ancestor. Here, to call the ancestor adaptively creative for successfully innovating a style or technique is misleading, since the value of the style or technique created might only be realised through its role for the group. The style or technique might otherwise be arbitrary. Although the social system maintains the style over the long-term, the individual may well generatively develop styles and techniques according to idiosyncratic methods, and sustain them for some time. Admittedly we may never know the true generative status of individuals, since their creative output is always manifest in social interactive contexts, but introspectively we can all appreciate the generative capacity of the mind at work, producing thoughts in the background without regard for their value. The technique of brainstorming involves the idea of holding back value judgement, so that generative thought processes can operate more freely in the individual, thus shifting the process of value judgement required for adaptive creativity to the collective level.

Generatively created patterns or behaviours can thus be exported to social systems through a process of "creating value". The capacity to create value may itself be an adaptive skill, or a matter of social context or luck: a more influential individual might have more freedom to act generatively than someone trying to fit in; a maternal ancestor might have experienced reproductive success for distinct genetic reasons, which carries the success of their otherwise insignificant cultural behaviour through vertical cultural transmission. Value creation does not necessarily mean "adding value" (as in making the world a better place), but "manipulating value": shifting the social conditions within which other individuals must act. A challenge for arts-based computational creativity is to understand whether "adding value" is at all meaningful: can we make better art through technology? To assume so without evidence would justifiably be viewed as complacency.

Alternative social aspects of the arts such as identity cast into doubt the centrality to arts-based computational creativity of the capacity to evaluate, which is commonly cited as critical in building artificial creative systems. From the perspective of strict adaptive creativity this is less problematic: an individual cannot behave adaptively if it cannot determine the real-world value of its creative produce. But if an individual is able to create value through influence, then the role of evaluation in the creative process should strike a balance with other elements. Evaluation in human artistic behaviour must be understood in the context of value creation, and other aspects of artistic social interaction. We risk turning evaluation into a bottleneck through which we squeeze all artistic interaction. Escaping the narrow focus on assessing aesthetic value, which avoids the need for a social individual that might be capable of exporting or creating value, is an important but challenging direction for arts-based computational creativity: what other dimensions of response, meaning and interaction are needed in computational systems?

14.4 Generative and Adaptive Approaches to Arts-Based Computational Creativity

Human-like adaptive creativity is the more traditional goal of arts-based computationally creative systems, but faces the challenge that the embodiment and situatedness of the artificial system is a poor reproduction of that of the human. It also faces the additional challenge of building adaptively creative systems that satisfy the constrained target of "valued" artistic output. Subsequently, some of the more successful examples of computational creativity have been human-system collaborations, such as Harold Cohen and his *AARON* software (McCorduck 1990), or George Lewis and his *Voyager* software (Lewis 2000).

A generative creativity approach seems equally problematic since generative creative systems are not adapted to goals and so cannot perform functions similar to human adaptive creativity. But if artistic creativity in cultural systems and humans involves the kinds of interaction between generative and adaptive processes discussed above, then a useful goal for arts-based computational creativity is to better understand this interaction in models and in experiments with interaction in artistic social behaviour, including studying the role of value as a medium of interaction between different systems. Through this understanding we can find ways to hybridise generative and adaptive creative processes. Two useful avenues of research are as follows.

14.4.1 Generative Creative Systems Can Be Externally Useful

Arthur (2009) describes technology in terms of phenomena: aspects of the world revealed through experimental interaction. In this view, innovation occurs through the exploitation of phenomena in the service of human goals. By revealing new phenomena, generative creative processes make new innovations feasible. We exploit the properties of materials, which can be produced through a generative process of chemical interaction. Cosmological and geological processes have produced numerous useful materials without purpose, and have not produced others. Likewise, although the products of natural evolution can be seen as having evolved to fulfil a purpose, we may exploit those products in ways that have nothing to do with their evolutionary origins: using a bird's feather as a writing implement, for example. Invention goes from feasible to easy to obvious when the generative process not only makes a material but makes it abundant, as in this example. This is often described in terms of the *affordances* offered by some structure. A celebrated form of human creativity involves re-appropriating existing things for new uses. The fact that things can regularly be re-appropriated indicates the efficacy of generative creative processes.

Pharmaceutical companies search the rich ecosystems of uncharted rainforest for novel species that might offer medicinal utility. Rather than being a coincidence that natural evolution generates things that are useful to humans without having evolved for this purpose, this seems to be more likely to reflect a simple principle that things useful for one purpose can be useful for others. Similarly, such companies search for new synthetic drugs by brute force, testing each candidate for a number of effects (not one specific goal). At the extreme, the side effects of a drug are noted in case there are novel effects that could be of use: putting solutions before problems.

As before, those who prefer to see natural evolution as more of an adaptively creative process, may prefer to see the above as a case of the transferability of adaptive creativity from one domain (what it evolved for) to another. This is discussed in the following section.

The same reasoning can be applied to the potential for artificial generative creative systems to produce artistic material. The *Creative Ecosystems* project at the Centre for Electronic Media Art (McCormack and Bown 2009, Bown 2009) has explored the creative potential of ecosystem models, even if those models are closed and not responding to the requirements of human "users". The output of a generatively creative virtual ecosystem can be of direct aesthetic value, through the generation of inherently fascinating patterns and behaviours, in the same way that the products of natural evolution are an endless source of aesthetic fascination. This is based on the assumption that complex structure and behaviour geared to an emerging purpose, one that is generated from within the system, is meaningful and compelling. This may require the hands of a skilled artist to be fully realised. It may also be possible to develop methodologies that allow creative design to become more tightly coupled with simulated ecosystemic processes, so that someone working within a creative domain can apply ecosystemic tools in order to generate novel outputs that are appropriate to that domain.

Given the generatively creative potency of natural evolution, artificial evolutionary ecosystems, if successful, might demonstrate computational generative creativity applicable to artistic outputs. But more commonplace generative creativity can be found in existing approaches to creative computing (Whitelaw 2004), for example in which stochastic processes can be used to generate infinite variations on a theme. A common practice in electronic music production is to implement rich generative processes that exhibit constant variation within given bounds and then either search the parameter space of such processes for good settings which can be used as required, or record the output of the process for a long time and select good sections from this recording as raw material. In both cases, a generative creative process (one which is in no way coupled to the outside world of value, and also, in most cases, has no internal value system either) is itself a creative output, but also plays the role of a tool for generating useful output. Such systems can only be involved in adaptively creative processes with an adaptively creative individual masterminding this process.

Multi-agent approaches such as the ecosystemic approach discussed here, and attempts at creative social models, such as those of Miranda et al. (2003), are also different in that they do contain a notion of value internal to the system. This means that they can potentially be generators of adaptive creativity, and that potentially we may be able to find ways to couple the value system found within a model to that found in the outside world. One solution has been proposed by Romero et al. (2009) in their Hybrid Society model, in which human and artificial users interact in a shared environment.

14.4.2 Adaptive Creative Systems Can Be Useful to Others

Individual adaptive creativity can be useful to others in two ways: firstly many individual innovations, such as washing food, are innovations that are both immediately useful to the individual that makes the innovation, and also to others who are able to imitate the behaviour. In this way, imitation and creativity are linked by the fact that the more adaptively creative one's conspecifics are, the more valuable it is to imitate their behaviour. They are likely to have discovered behaviours that are useful to themselves, and by virtue of your similarity, probably useful to you too (although by no means definitely). Adaptive imitation of behaviour is a particularly human capability, and a challenging cognitive task (Conte and Paolucci 2001). The imitation of successful behaviours allows human social systems to be cumulatively creative, amassing knowledge and growing in complexity (Tomasello 1999). Secondly, as discussed in Sect. 14.3.2.2, social structures bind individuals together into mutually adaptive behaviours: John Harrison did not build clocks so that he himself could better tell the time at sea.

How this common or mutually adaptive value works in the arts, however, is less clear, since the value of any individual behaviour is determined not with respect to a static physical environment but a dynamic social one (Csikszentmihalyi 1996). Whereas the value of washing food does not change the more individuals do it, the value of an artistic behaviour can change radically as it shifts from a niche behaviour to a mainstream one. In this way, copying successful behaviour does not necessarily lead to an accumulation of increasingly successful behaviour, as in the accumulation of scientific knowledge, but can also lead to turbulence: unstable social dynamics predicated on feedback. The value of artworks to individuals is highly context-specific and suggests this kind of dynamic. Thus it seems more appropriate to look to the second way in which adaptively creative systems can be of use to others, by being locked into mutually beneficial goals through social structures, but also to recognise that copying successful styles is an essential part of this process. The arts appear to involve *ad hoc* groupings of individuals who share common goals, into which adaptively creative arts-based computational systems could become integrated and be of benefit to individual humans. This points to the idea that achieving success in arts-based computational creativity is as much a matter of establishing appropriate individual and social creative contexts, practices and interfaces as it is of designing intelligent systems.

The *Drawbots project* investigated the idea of producing physically embodied autonomous robot artists which could honestly be described as the authors of their own work, rather than as proxies for a human's creativity (Bird and Stokes 2006). This would have overcome the limitations to the agency of the software in examples such as Harold Cohen's *AARON* (McCorduck 1990), where Cohen is clearly the master of the creative process, and AARON the servant. The project illustrated the fundamental conundrum of attempting to embed an artificial system into an artistic context without proper channels through which value can be managed. In fact, the drawings produced by the Drawbot were no more independent of their makers than AARON's, and arguably had less of a value connection to the outside world than AARON did, even though AARON's connection was heavily mediated by Cohen, its proverbial puppeteer. The Drawbots possessed independence in a different sense, in so far as they were embedded in their own artificial system of value. Thus each individual Drawbot was individually adapted (the product of an evolutionary process) but not adaptively creative, and the entire system was generatively creative (able to lead to new patterns and behaviours) but also not adaptively creative, and thus not creative in the sense of a human artist.

14.5 Conclusion

In the last example and elsewhere in this chapter I have linked a discussion of generative and adaptive creativity in social systems and individual humans to research in arts-based computational creativity and its goals. Arts-based computational creativity is well underway as a serious research field, but it faces a truly grand challenge. There is still some way to go to break down this challenge into manageable and clearly defined goals, frustrated by the ill-defined nature of artistic evaluation. But a pattern is emerging in which arts-based computationally creative systems can be categorised in terms of how they relate to the wider world of human artistic value, either as prosthetic extensions of individual creative practices, as in the case of AARON and many other uses of managed generatively creative processes, or as experiments in adaptive creativity which are generally not presently capable of producing valued artistic output, such as the DrawBots project and various models of social creative processes. In the case of artificial generatively creative systems, the analysis presented here suggests that it is important to analyse such systems both as tools in an adaptively creative process involving goal-driven individuals, and as elements in a heterogeneous social network which itself exhibits generative creativity. In both cases, it is valuable to consider what status such systems will possess in terms of primary and secondary agency.

As long as adaptive and generative creativity can be recognised as distinct processes, they can be addressed simultaneously in a single project. For example, the ecosystemic approach mentioned in Sect. 14.4.1 attempts to straddle these areas of interest by acting both as a generative tool, of direct utility to artists, and as a virtual environment in which the potential for adaptive creativity by individual agents can be explored. In this way, methods might be discovered for coupling the value system that the artist is embedded in and the emergent value system within the artificial ecosystem. The latter may be a simulation of the former, or a complementary generative system. Furthermore, since novel arts-based computational creativity technologies can be shared, modified and re-appropriated by different users, they already have a social life of their own as secondary agents, even if they are not primary social agents. As such they are adaptive in a memetic sense. Both this and the ecosystemic approach may be able to offer powerful mechanisms for bootstrapping arts-based computational creativity towards increasingly complex behaviours, greater artistic success, and an increased appearance of primary agency, without modelling human cognition.

Acknowledgements This chapter stems from ideas formed during my PhD with Geraint Wiggins at Goldsmiths, University of London, and further developed whilst working as a post-doctoral researcher at the Centre for Electronic Media Art (CEMA), with Jon McCormack (funded by the Australian Research Council under Discovery Project grant DP0877320). I thank Jon, Alan Dorin, Alice Eldridge and the other members of CEMA for two years of fascinating recursive discussions.

I am grateful to all of the attendees of the 2009 Dagstuhl symposium on Computational Creativity, in particular the organisers, Jon McCormack, Mark d'Inverno and Maggie Boden, for contributing to a first-rate creative experience. I also thank the anonymous reviewers for their valuable feedback.

References

Arthur, W. B. (2009). *The nature of technology: what it is and how it evolves*. Baltimore: Penguin.

Axelrod, R. (1997). *The complexity of cooperation*. Princeton: Princeton University Press.

Barkow, J. H., Cosmides, L., & Tooby, J. (1992). *The adapted mind: evolutionary psychology and the generation of culture*. New York: OUP.

Bentley, P. J. (1999a). From coffee tables to hospitals: generic evolutionary design. In P. J. Bentley (Ed.), *Evolutionary design by computers*. San Francisco: Morgan Kaufmann.

Bentley, P. J. (Ed.) (1999b). *Evolutionary design by computers*. San Francisco: Morgan Kaufmann.

Bergson, H. (1998). *Creative evolution*. New York: Dover.

Bird, J., & Stokes, D. (2006). Evolving minimally creative robots. In S. Colton & A. Pease (Eds.), *Proceedings of the third joint workshop on computational creativity (ECAI '06)* (pp. 1–5).

Blackmore, S. J. (1999). *The meme machine*. New York: OUP.

Boden, M. (1990). *The creative mind*. George Weidenfeld and Nicolson Ltd.

Bown, O. (2008). *Theoretical and computational models of cohesion, competition and maladaptation in the evolution of human musical behaviour*. PhD thesis, Department of Computing, Goldsmiths College, University of London.

Bown, O. (2009). Ecosystem models for real-time generative music: A methodology and framework. In *Proceedings of the 2009 international computer music conference (ICMC 2009)*, Montreal, Canada.

Bown, O., & Wiggins, G. A. (2005). Modelling musical behaviour in a cultural-evolutionary system. In P. Gervàs, T. Veale, & A. Pease (Eds.), *Proceedings of the IJCAI'05 workshop on computational creativity*.

Boyd, R., & Richerson, P. J. (1985). *Culture and the evolutionary process*. Chicago: University of Chicago Press.

Brown, S. (2007). Contagious heterophony: A new theory about the origins of music. *Musicæ Scientiæ*, *11*(1), 3–26.

Clark, A. (2003). *Natural-born cyborgs: minds, technologies, and the future of human intelligence*. London: Oxford University Press.

Coe, K. (2003). *The ancestress hypothesis: visual art as adaptation*. New Brunswick: Rutgers University Press.

Conte, R., & Paolucci, M. (2001). Intelligent social learning. *Journal of Artificial Societies and Social Simulation*, *4*(1).

Cross, I. (2006). The origins of music: some stipulations on theory. *Music Perception*, *24*(1), 79–82.

Csikszentmihalyi, M. (1996). *Creativity: flow and the psychology of discovery and invention*. New York: Harper Collins.

Csikszentmihalyi, M. (1999). Implications of a systems perspective for the study of creativity. In R. J. Sternberg (Ed.), *The handbook of creativity*. New York: Cambridge University Press.

Dawkins, R. (1976). *The selfish gene*. New York: OUP.

Dawkins, R. (1986). *The blind watchmaker: why the evidence of evolution reveals a universe without design*. London: Penguin.

de Landa, M. D. (1991). *War in the age of intelligent machines*. Swerve Editions.

Diamond, J. M. (1992). *The rise and fall of the third chimpanzee*. New York: Vintage.

Dissanayake, E. (2000). Antecedents of the temporal arts in early mother-infant interaction. In N. L. Wallin, B. Merker & S. Brown (Eds.), *The origins of music*. Cambridge: MIT Press.

Dunbar, R. (1998). The social brain hypothesis. *Evolutionary Anthropology*, *6*, 178–190.

Fisher, R. A. (1958). *The genetical theory of natural section*. London: Dover.

Gell, A. (1998). *Art and agency: an anthropological theory*. Oxford: Clarendon Press.

Gilbert, N. (1993). Computer simulation of social processes. *Social Research Update, 6*. University of Surrey.

Greenfield, G. R. (2006). Art by computer program == programmer creativity. *Digital Creativity*, *17*(1), 25–35.

Hagen, E. H., & Bryant, G. A. (2003). Music and dance as a coalition signaling system. *Human Nature*, *14*(1), 21–51.
Hamilton, W. D. (1963). The evolution of altruistic behaviour. *American Naturalist*, *97*, 354–356.
Hargreaves, D. J. (1986). *The developmental psychology of music*. Cambridge: Cambridge University Press.
Huron, D. (2001). Is music an evolutionary adaptation. In R. J. Zatorre & I. Peretz (Eds.), *Annals of the New York academy of sciences: Vol. 960. The biological foundations of music* (pp. 43–61). New York: New York Academy of Sciences.
Krebs, J. R., & Dawkins, R. (1984). Animal signals: mind reading and manipulation. In J. R. Krebs & N. B. Davies (Eds.), *Behavioural ecology: an evolutionary approach* (2nd ed., pp. 380–402). Oxford: Blackwell.
Laland, K. N., Odling-Smee, J., & Feldman, M. W. (1999). Niche construction, biological evolution and cultural change. *Behavioral and Brain Sciences*, *21*(1).
Latour, B. (1993). *We have never been modern*. Cambridge: Harvard University Press.
Lévi-Strauss, C. (1971). *The elementary structures of kinship*. Beacon Press.
Lewis, G. E. (2000). Too many notes: computers, complexity and culture in voyager. *Leonardo Music Journal*, *10*, 33–39.
Lovelock, J. (1979). *Gaia. A new look at life on Earth*. New York: OUP.
Martindale, C. (1990). *The clockwork muse: the predictability of artistic change*. New York: Basic Books.
Maynard Smith, J., & Szathmáry, E. (1995). *The major transitions in evolution*. New York: Oxford University Press.
McCorduck, P. (1990). *AARON's code: meta-art, artificial intelligence, and the work of Harold Cohen*. New York: Freeman.
McCormack, J. (2008). Facing the future: evolutionary possibilities for human-machine creativity. In J. Romero & P. Machado (Eds.), *The art of artificial evolution: a handbook on evolutionary art and music* (pp. 417–451). Heidelberg: Springer.
McCormack, J., & Bown, O. (2009). Life's what you make: niche construction and evolutionary art. In *Applications of evolutionary computing: EvoWorkshops, 2009*.
Merker, B. (2001). Synchronous chorusing and human origins. In N. L. Wallin, B. Merker & S. Brown (Eds.), *The origins of music*, Cambridge: MIT Press.
Miller, G. (2000). *The mating mind*. New York: Random House.
Miranda, E. R., Kirby, S., & Todd, P. M. (2003). On computational models of the evolution of music: from the origins of musical taste to the emergence of grammars. *Contemporary Music Review*, *22*(3), 91–111.
Odling-Smee, F. J., Laland, K. N., & Feldman, M. W. (2003). *Niche construction: the neglected process in evolution. Monographs in population biology: Vol. 37*. Princeton: Princeton University Press.
Owings, D. H., & Morton, E. S. (1998). *Animal vocal communication: a new approach*. New York: Cambridge University Press.
Parncutt, R. (2009). Prenatal and infant conditioning, the mother schema, and the origins of music and religion. *MusicæScientiæ*, 119–150. Special issue on evolution and music.
Pearce, M. T., & Wiggins, G. A. (2001). Towards a framework for the evaluation of machine compositions. In *Proceedings of the AISB'01 symposium on artificial intelligence and creativity in the arts and sciences* (pp. 22–32). Brighton: SSAISB.
Perkins, D. N. (1996). Creativity: beyond the Darwinian paradigm. In M. Boden (Ed.), *Dimensions of creativity* (Chap. 5, pp. 119–142). Cambridge: MIT Press.
Pinker, S. (1998). *How the mind works*. London: Allen Lane The Penguin Press.
Richman, B. (2001). How music fixed "nonsense" into significant formulas: on rhythm, repetition and meaning. In N. L. Wallin, B. Merker & S. Brown (Eds.), *The origins of music*. Cambridge: MIT Press.
Romero, J., Machado, P., & Santos, A. (2009). On the socialization of evolutionary art. In M. Giacobini, A. Brabazon, S. Cagnoni, G. A. Di Caro, A. Ekárt, A. Esparcia-Alcázar, M. Farooq, A. Fink, P. Machado, J. McCormack, M. O'Neill, F. Neri, M. Preuss, F. Rothlauf, E. Tarantino

& S. Yang (Eds.), *Lecture notes in computer science: Vol. 5484. EvoWorkshops* (pp. 557–566). Berlin: Springer.

Saunders, R. (2001). *Curious design agents and artificial creativity*. PhD thesis, Faculty of Architecture, The University of Sydney.

Saunders, R., & Gero, J. S. (2001). The digital clockwork muse: a computational model of aesthetic evolution. In G. A. Wiggins (Ed.), *Proceedings of the AISB'01 symposium on AI and creativity in arts and science, SSAISB*, University of York, York, UK (pp. 12–21).

Secretan, J., Beato, N., Ambrosio, D. B., Rodriguez, A., Campbell, A., & Stanley, K. O. (2008). Picbreeder: evolving pictures collaboratively online. In *CHI '08: proceeding of the twenty-sixth annual SIGCHI conference on human factors in computing systems* (pp. 1759–1768). New York: ACM.

Sims, K. (1994). Artificial life IV proceedings. In *Evolving 3D morphology and behaviour by competition*. Cambridge: MIT Press.

Sobel, D. (1996). *Longitude: the true story of a lone genius who solved the greatest scientific problem of his time*. Baltimore: Penguin.

Sosa, R., & Gero, J. S. (2003). Design and change: a model of situated creativity. In C. Bento, A. Cardoso & J. Gero (Eds.), *Proceedings of the IJCAI'03 workshop on creative systems*.

Sperber, D. (2000). An objection to the memetic approach to culture. In *Darwinizing culture*. London: Oxford University Press.

Sperber, D. (2007). Seedless grapes: nature and culture. In E. Margolis & S. Laurence (Eds.), *Creations of the mind: theories of artefacts and their representation*. London: Oxford University Press. Chapter 7.

Stanley, K. O., & Miikkulainen, R. (2004). Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, *21*, 63–100.

Takagi, H. (2001). Interactive evolutionary computation: fusion of the capabilities of ec optimization and human evaluation. *Proceedings of the IEEE*, *89*, 1275–1296.

Thornton, C. (2007). How thinking inside the box can become thinking outside the box. In A. Cardoso & G. A. Wiggins (Eds.), *Proceedings of the 4th international joint workshop on computational creativity* (pp. 113–119). Goldsmiths: University of London.

Tomasello, M. (1999). *The cultural origins of human cognition*. Harvards: Harvard University Press.

Whitelaw, M. (2004). *Metacreation: art and artificial life*. Cambridge: MIT Press.

Williams, R. (1983). *Keywords* (revised ed.). London: Fontana Press.

Wilson, E. O. (1975). *Sociobiology: the new synthesis*. Harvards: Harvard University Press.

Zahavi, A. (1975). Mate selection—a selection for a handicap. *Journal of Theoretical Biology*, *53*(1), 205–214.

Chapter 15
Creating New Informational Primitives in Minds and Machines

Peter Cariani

Abstract Creativity involves the generation of useful novelty. Two modes of creating novelty are proposed: via new combinations of pre-existing primitives (combinatoric emergence) and via creation of fundamentally new primitives (creative emergence). The two modes of creativity can be distinguished by whether the changes still fit into an existing framework of possibility, or whether new dimensions in an expanded interpretive framework are needed. Although computers are well suited to generating new combinations, it is argued that computations within a framework cannot produce new primitives for that framework, such that non-computational constructive processes must be utilised to expand the frame. Mechanisms for combinatoric and creative novelty generation are considered in the context of adaptively self-steering and self-constructing goal-seeking percept-action devices. When such systems can adaptively choose their own sensors and effectors, they attain a degree of epistemic autonomy that allows them to construct their own meanings. A view of the brain as a system that creates new neuronal signal primitives that are associated with new semantic and pragmatic meanings is outlined.

15.1 Introduction

Open-endedness is an important goal for designing systems that can autonomously find new and unexpected solutions to combinatorically-complex and ill-defined problems. Classically, issues of open-ended generation of novelty in the universe have come under the rubric of the problem of emergence.

In this discussion we distinguish two general modes of creating novelty: *combinatoric emergence* and *creative emergence*. In combinatoric emergence new combinations of existing primitives are constructed, whereas in creative emergence entirely new primitives are created anew. Although combinatoric systems may differ in numbers of possible combinations, their set of possibilities is closed. Creative systems, on the other hand, have open-sets of possibilities because of the partial or

P. Cariani (✉)
Department of Otology & Laryngology, Harvard Medical School, Boston, MA, USA
e-mail: cariani@mac.com

J. McCormack, M. d'Inverno (eds.), *Computers and Creativity*,
DOI 10.1007/978-3-642-31727-9_15, © Springer-Verlag Berlin Heidelberg 2012

ill-defined nature of the space of possible primitives. The dual, complementary conceptions provide two modes for describing and understanding change and creativity: as the unfolding consequences of fixed combinatorial rules on bounded sets of predefined primitives or as the effects of new covert processes and interactions that come into play over time to provide new effective dimensional degrees of freedom.

We face several related problems. We want to know how to recognise creative novelty when it occurs (the *methodological problem*). We also want to understand the creative process in humans and other systems (the *scientific problem*) such that creativity in human-machine collaborations can be enhanced and semi-autonomous, creative devices can be built (the *design problem*).

The methodological problem can be solved by the "emergence-relative-to-a-model" approach in which an observer forms a model of the behaviour of a system (Sect. 15.4). Novelty and creativity are inherently in the eye of the observer, i.e. relative to some model that specifies expected behaviours amongst possible alternatives. If the behaviour changes, but it can still be predicted or tracked in terms of the basic categories or state set of the model, one has rearrangement of trajectories of existing states (combinatorial creativity). If behaviour changes, but in a manner that requires new categories, observables, or states for the observer to regain predictability, then one has the creation of new primitives (emergent creativity).

Solution of the scientific problem of creativity requires a clear description of what creativity entails in terms of underlying generative and selective processes. Creativity exists in the natural world on many levels, from physical creation (particles, elements, stars, galaxies) through the origins and evolution of life (multicellularity, differentiated tissues, circulatory, nervous, and immune systems) to concept formation in brains and new modes of social organisation. What facilitating conditions and organisations lead to such creativity? In biological evolutionary contexts the main underlying mechanisms are Darwinian processes of genetic inheritance with variation/recombination, genetically-steered phenotypic construction, and selection by differential survival and reproduction. On the other hand, in neural contexts that support creative learning processes the mechanisms appear to involve more directed, Hebbian stabilisations of effective neural connectivities and signal productions.

Ultimately we seek to build artificial systems that can enhance human creativity and autonomously create new ideas that we ourselves unaided by machines would never have discovered. This will entail designing mechanisms for combinatorial generation and for creation of new primitives. Essentially all adaptive, trainable machines harness the power of combinatorial spaces by finding ever better combinations of parameters for classification, control, or pattern-generation. On the contemporary scene, a prime example is the genetic algorithm (Holland 1975; 1998), which is a general evolutionary programming strategy (Fogel et al. 1966) that permits adaptive searching of high-dimensional, nonparametric combinatorial spaces.

Unfortunately, very few examples of artificial systems capable of emergent creativity are yet to be found. For the most part, this is due to the relative ease and economy with which we humans, as opposed to machines, can create qualitatively new solutions. We humans remain the pre-eminent generators of emergent creativity on our planet. It is also due in part to the primary reasons that we create machines—to carry out pre-specified actions reliably and efficiently. We usually prefer our devices to act predictably, to carry out actions we specify, rather than to surprise us in some fundamental way. In contrast, we expect our artists, designers, and scientists to continually surprise us.

Nonetheless, creatively emergent artificial systems are possible and even desirable in some contexts. In Sect. 15.3 we consider the problem of creativity in the context of adaptive goal-seeking percept-action systems that encapsulate the functional organisations of animals and robots (Figs. 15.2, 15.3, 15.6). Such systems carry out operations of measurement (via sensors), action (via effectors), internal coordination (via computational mappings, memory), steering (via embedded goals), and self-construction (via mechanisms for plastic modification). We then discuss the semiotics of these operations in terms of syntactics (relations between internal informational sign-states), semantics (relations between sign-states and the external world), and pragmatics (relations between sign-states and internal goals). New primitive relations can be created in any of these realms (Table 15.1).

We are already quite adept at creating ever more powerful computational engines, and we can also construct robotic devices with sensors, effectors, and goal-directed steering mechanisms that provide them with fixed, pre-specified semantics and pragmatics. The next step is to design machines that can create new meanings for themselves. What is needed are strategies for creating new primitive semantic and pragmatic linkages to existing internal symbol states.

Three basic strategies for using artificial devices to create new meanings and purposes present themselves:

1. *via new human-machine interactions* (mixed human-machine systems in which machines provoke novel insights in humans who then provide new interpretations for machine symbols),
2. *via new sensors and effectors on an external world* (epistemically-autonomous evolutionary robots that create their own external semantics), and
3. *via evolving internal analog dynamics* (adaptive self-organisation in mixed analog-digital devices or biological brains in which new internal linkages are created between internal analog representations that are coupled to the external world and goal-directed internal decision states).

The first strategy uses machines to enhance human creative powers, and arguably, most current applications of computers to creativity in the arts and sciences involve these kinds of human-machine collaborations. But the processes underlying human thought and creativity in such contexts are complex and ill-defined, and therefore difficult to study by observing overt human behaviour.

The second and third strategies focus on building systems that are capable of emergent creativity in their own right. In Sect. 15.3 and Sect. 15.5 respectively, we outline a basic accounts of how new primitives might arise in adaptive percept-action systems of animals and robots (emulating emergence in biological evolution) and how new neural signal primitives might arise in brains (emulating creative processes in individual humans and animals). Combinatoric and creative emergence is first considered in the framework of a taxonomy of adaptive, self-constructing cybernetic robotic percept-action systems. One can then also consider what such an open-ended functional framework might mean for envisioning new kinds of neural networks that are capable of forming novel internal informational primitives. In this framework, adaptively-tuned neuronal assemblies function as self-constructed internal sensors and signal generators, such that new signal types associated with new concepts can be produced. The new signals then serve as internal semantic tags that function as annotative additions to the input signals that evoked their production. Emergence of new signal types in such a system increases the effective dimensionality of internal signal spaces over time, thus bringing new conceptual primitives into action within the system.

15.2 Emergence and Creativity

Emergence concerns the means by which novelty arises in the world. Intuitively, emergence is the process by which new, more complex order arises from a simpler or more predictable preceding situation. As such, images of birth, development, and evolution infuse our notions of emergence. These images provide intuitive explanations for how novelty, spontaneity, and creativity are possible and how complex organisations arise and become further elaborated.

All around us we see the complex organisations that are the emergent products of biological, psychological and social processes, and as a result, our current discourses on emergence encompass a wide range of phenomena. Novelty appears in the form of new material structures (thermodynamic emergence), formal structures (computational emergence), biological structures and functions (emergent evolution), scientific theories (emergence vs. reduction), modelling relations in observers, percepts, ideas, notational systems, and economic and social relations. Novelty and innovation are integral processes in natural and social worlds, and are coming to play ever-larger roles in artificial worlds as well.

Two fundamental kinds of emergent novelty can be distinguished, which we can call *combinatoric emergence* and *creative emergence*. Lloyd Morgan (1931) in his book “Emergent Evolution” made a similar distinction, labelling new combinations “resultants” and new primitives “emergents”. This distinction became central to my work on epistemology and evolutionary robotics, which developed an operational systems-theoretic methodology for distinguishing one process from the other (Cariani 1989). Some of my earliest inspirations came from considering the nature of novelty in biological evolution, where creation of new combinations of existing genetic alternatives and refinements of existing functions (“microevolution”) can be contrasted with creation of entirely new genes, species, morphologies, and functions (“macroevolution”). The combinatoric/creative distinction also parallels Margaret Boden’s division of explorative vs. transformational creativity (Boden 1990a; 1994; 1994b; 2006).

The two kinds of combinatoric and creative novelty reflect different deeply divergent conceptions of order and its origins, “order-from-order” vs. “order-from-noise”

(Piatelli-Palmarini 1980), that are associated with different organising paradigms (Maruyama 1977), and "world hypotheses" (Pepper 1942). Where order comes from order, novelty is but a preformationist unfolding of latent possibility or recombination of existing parts; where order arises from noise, chaos, formlessness, or ambiguity, novelty entails *de novo* formation of new realms of possibility vis-à-vis existing observational and interpretive frameworks.

My purpose in considering emergence and this combinatoric-creative distinction is and has always been primarily pragmatic. For this reason, we focus here primarily on developing heuristics for generating useful novelty rather than in engaging in philosophical debates over the status of emergent novelty vis-a-vis various postulated ontological frameworks. For useful general introductions to the philosophical problems and their implications for mind-body relations, free will, and ontological emergence, see Kim (2008) and Clayton (2004). For similar reasons we will almost entirely sidestep the literature on complexity and emergence. Complexity in and of itself does not necessarily produce anything useful, nor does it necessarily provide insights into how to do so. On the other hand, variety is the mother of invention, and increased structural complexity does provide variety in the form of more accessible states and effective degrees of freedom. Processes of "complication" (von Neumann 1951) thus serve as fodder for new functions.

15.2.1 What Constitutes a New Primitive?

Both kinds of emergence, combinatoric and creative, entail recognition of basic sets of possibilities that constitute the most basic building blocks of the order, i.e. its atomic parts or "primitives".

By a "primitive", we mean an indivisible, unitary entity, atom, or element in a system that has no internal parts or structure of its own in terms of its functional role in that system. Individual symbols are the primitives of symbol string systems, binary distinctions are the primitives of flip-flop-based digital computers, and machine states are the primitives of finite state automata. To paraphrase Gregory Bateson, a primitive is a unitary "difference that makes a difference".

Emergence then entails either the appearance of new combinations of previously existing primitives or the formation of entirely new ones (Fig. 15.1). The primitives in question depend upon the discourse; they can be structural, material "atoms"; they can be formal "symbols" or "states"; they can be functionalities or operations; they can be primitive assumptions of a theory; they can be primitive sensations and/or ideas; they can be the basic parts of an observer's model.

Most commonly, the primitives are assumed to be structural, the parts that are put together in various combinations to make aggregate structures. Reductionist biology in effect assumes that everything that one would want to say about biological organisms can be expressed in terms of molecular parts. For many contexts and purposes, such as molecular biology and pharmaceutical development, where structure is key, this is an appropriate and effective framework. For other pursuits,

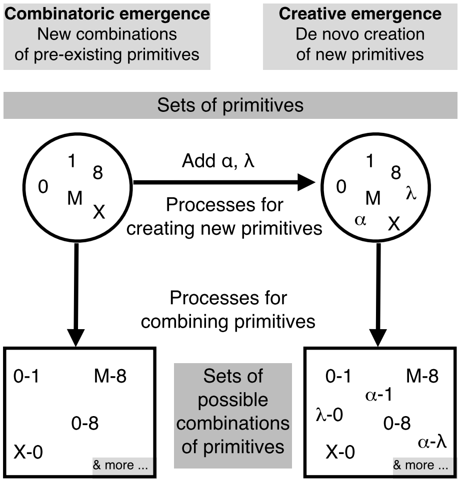


Fig. 15.1 Combinatoric and creative emergence

additional organisational and functional primitives are needed. If one wants to understand how an organism functions as a coherent, self-sustaining whole, one needs more than reductive parts-lists and local mechanisms. One needs concepts related to organisation and function, and knowledge of the natural history of how these have arisen. Living systems are distinct from nonliving ones because they embody particular organisations of material processes that enable organisational regeneration through self-production (Maturana and Varela 1973). Biological organisations also lend themselves to functional accounts that describe how goal-states can be embedded in their organisation and how goals can reliably realised by particular arrangements of processes. Full molecular descriptions of organisms do not lead to these relational concepts. Similarly, full molecular descriptions of brains and electronic computers, though useful, will not tell us how these systems work as information processing engines. If artificial systems are to be designed and built along the same lines as organisms and brains, new kinds of primitives appropriate for describing regenerative organisation and informational process are required.

15.2.2 Primitives and Interpretive Frames

Once one has defined what the primitives are or how they are recognised, then one has constructed a frame for considering a particular system. To say that an entity is "primitive" relative to other objects or functions means it cannot be constructed from combinations of the other entities of that frame, i.e. its properties cannot be logically deduced from those of the other entities. Although it may be possible, in reductionist fashion, to find a set of lower level primitives or observables from which the higher level primitives can be deduced, to do so requires a change of frame—one is then changing the definition of the system under consideration.

In the example of Fig. 15.1, the individual Roman and Greek letters and numerals are the primitives of a symbol-string system. Although concrete letters and numerals themselves do indeed have internal structure, in terms of strokes, arcs, and straight lines, these parts play no functional role in the system beyond supporting the distinction and recognition of their unitary symbol types. Once the classification of the type of symbol is made, the internal structure of the lines and curves become irrelevant. Were we to suddenly to adopt a frame in which the lines and curves are the primitives, then the appearance of the new symbols on the right, the Greek letters alpha and lambda, would not surprise us because these can be formed though combinations of the lower level strokes.

The combinatoric-creative distinction parallels ontological vs. epistemological modes of explanation. The debate that occurred in 1976 in France between Piaget, Chomsky, and Fodor over the origins of new ideas is illuminating. As organiser-participant Piatelli-Palmarini (1980) so elegantly pointed out, this was really a debate over the existence and nature of emergent novelty in the world. The two poles of the debate were held by Fodor (1980) and Piaget (1980). Fodor argued an extreme preformationist view in which all learning is belief-fixation, i.e. selection from a fixed repertoire of possible beliefs, such that entirely new ideas are not possible. Piaget presented an emergentist view in which qualitatively novel, irreducible concepts in mathematics have been created anew over the course of its history.

All that is possible in traditional ontological frameworks is recombination of existing possible constituents, whereas in epistemological frameworks, novelty can reflect surprise on the part of a limited observer. Another way of putting this is that ontologically-oriented perspectives adopt fixed, universal frames, whereas epistemologically-oriented ones are interested in which kinds of systems cause the limited observer to change frames and also what changes occur in the limited observer when frames are changed.

Second-order cybernetics (von Foerster 2003), systems theory (Kampis 1991), pragmatist theories of science (van Fraassen 1980), and constructivist epistemologies (von Glasersfeld 2007) are all concerned with "observing systems" that construct their own observational and interpretative frames. In Piaget's words "Intelligence organises itself to organise the world" (von Glasersfeld 1992). We examine different kinds of conceivable self-constructing observing-acting systems in Sect. 15.3. When these systems change their frames, they behave in novel ways that cause those observing them to alter their own frames (Sect. 15.4).

15.2.3 Novel Combinations of Closed Sets of Primitives

Combinatoric emergence engages a fixed set of primitives that are combined in new ways to form emergent structures. In biological evolution the genetic primitives are DNA nucleotide sequences. On shorter evolutionary timescales microevolutionary processes select amongst combinations of existing genetic sequences, whereas on longer timescales macroevolutionary processes entail selection amongst entirely new genes that are formed through novel sequences. On higher levels of biological organisation, emergent structures and functions can similarly arise from novel combinations of previously existing molecular, cellular, and organismic constituents. In psychology, associationist theories hold that emergent mental states arise from novel combinations of pre-existing primitive sensations and ideas. Whether cast in terms of platonic forms, material atoms, or mental states, combinatoric emergence is compatible with reductionist programs for explaining macroscopic structure through microscopic interactions (Holland 1998).

This strategy for generating structural and functional variety from a relatively small set of primitive parts is a powerful one that is firmly embedded in many of our most advanced informational systems. In the analytic-deductive mode of exploration and understanding, one first adopts some set of axiomatic, primitive assumptions, and then explores the manifold, logically-necessary consequences of those assumptions. In the realm of logic and mathematics, the primitives are axioms and their consequences are deduced by means of logical operations on the axioms. Digital computers are ideally suited for this task of generating combinations of symbol-primitives and logical operations on them that can then be evaluated for useful, interesting, and/or unforeseen formal properties. In the field of symbolic artificial intelligence (AI) these kinds of symbolic search strategies have been refined to a high degree. Correspondingly, in the realm of adaptive, trainable machines, directed searches use evaluative feedback to improve mappings between features and classification decisions. Ultimately these decisions specify appropriate physical actions that are taken. In virtually all trainable classifiers, the feature primitives are fixed and pre-specified by the designer, contingent on the nature of the classification problem at hand. What formally distinguishes different kinds of trainable machines is the structure of the combination-space being traversed, the nature of the evaluative feedback, and the rules that steer the search processes.

15.2.4 Limits on Computations on Existing Primitives

Combinatoric novelty is a dynamic, creative strategy insofar as it constantly brings into being new combinations of elements. However, such combinatoric realms are inherently limited by their fixed, closed sets of primitive elements. Consider the set of the digits 0–9 vs. a set of 10 arbitrarily distinguished objects. The first set is well-defined, has 10 actual members, and is closed, while the latter set is ill-defined, has an indefinite number of potential members, and is open.[1]

All that can happen within well-defined universes are recombinations of existing, pre-specified symbols—there is no means by which new primitive symbols can be created by simply recombining existing ones. It seems obvious enough that one

[1]A Platonist could claim that all sets are open because they can include null sets and sets of sets *ad infinitum*, but we are only considering here sets whose members are collections of concrete individual elements, much in the same spirit as Goodman (1972).

does not create new alphabetical letter types by stringing together more and more existing letters—new types must be introduced from outside the system. This is typically carried out by an external agent. Likewise, in our computer simulations, we set up a space of variables and their possible states, but the simulation cannot add new variables and states simply by traversing the simulation-states that we have previously provided.

These ideas bear directly on fundamental questions of computational creativity. What are the creative possibilities and limitations of pure computations? Exactly how one defines "computation" is critical here. In its more widely used sense, the term refers to any kind of information-processing operation. Most often, the issue of what allows one to distinguish a computation from a non-computational process in a real-world material system is completely sidestepped, and the term is left loose and undefined. However, in its more precise, foundations-of-mathematics sense, the term refers to concrete formal procedures that involve unambiguous recognitions and reliable manipulations of strings of meaningless symbols. It is this latter, more restrictive, sense of computation as formal procedure that we will use here. For practical considerations, we are interested in computations that can be carried out in the real world, such as by digital electronic computer, and not imagined operations in infinite and potentially-infinite realms.[2]

In these terms, pure computation by itself can generate new combinations of symbol primitives, e.g. new strings of existing symbols, but not new primitive symbols themselves. In order for new symbol primitives to be produced, processes other than operations on existing symbols must be involved—new material dynamics must be harnessed that produce new degrees of freedom and new attractor basins that can support additional symbol types. To put it another way, merely running programs on a computer cannot increase the number of total machine states that are enabled by the hardware. In order to expand the number of total machine states that are available at any given time, we must engage in physical construction, such as fabricating and wiring in more memory.

There was a point in the history of computing devices at which self-augmenting and self-organising physical computational devices were considered. In the early 1960s, when electronic components were still expensive, growing electronic logical components "by the pound" was contemplated:

---

[2]Popular definitions of computation have evolved over the history of modern computing (Boden 2006). For the purposes of assessing the capabilities and limitations of physically-realisable computations, we adopt a very conservative, operationalist definition in which we are justified in calling an observed natural process a *computation* only in those cases where we can place the observable states of a natural system and its state transitions in a one-to-one correspondence with those of some specified deterministic finite state automaton. This definition has the advantage of defining computation in a manner that is physically-realisable and empirically-verifiable. It results in classifications of computational systems that include both real world digital computers and natural systems, such as the motions of planets, whose observable states can be used for reliable calculation. This finitistic, verificationist conception of computation also avoids conceptual ambiguities associated with Gödel's Undecidability theorems, whose impotency principles only apply to infinite and potentially-infinite logic systems that are inherently not realisable physically.

> We believe that if the "complexity barrier" is to be broken, a major revolution in production and programming techniques is required, the major heresies of which would mean weakening of machine structural specificity in every possible way. We may as well start with the notion that with 10 billion parts per cubic foot (approximately equal to the number and density of neurons in the human brain), there will be no circuit diagram possible, no parts list (except possibly for the container and the peripheral equipment), not even an exact parts count, and certainly no free and complete access with tools or electrical probes to the "innards" of our machine or for possible later repair... We would manufacture 'logic by the pound', using techniques more like those of a bakery than of an electronics factory. (Stewart 1969)

Such ideas persist today in visions of self-replicating nanobot nanotechnologies (now with the accompanying spectre of "grey goo" ecological disaster). At various times there have also existed notions of universal self-organising analog computers (see discussion of the Russian *Gutenmacher project* in Carello et al. (1984)). Such computational systems that physically grow their own hardware would be desirable, but the practical need for such self-expansions has been obviated by human ingenuity and creativity in the form of fast-evolving Moore's-Law manufacturing efficiencies. It is simply easier to design and build a yet larger or faster machine than one that organically grows to become bigger or faster. Today's large, scalable server arrays that permit simple addition of new modules are perhaps the closest systems we have to such growing automata.

In any case, these various means of making additional, new material dynamics accessible to the device lie outside the realm of symbol manipulation—they are non-computational. Non-computational breakout strategies are therefore required for computational systems to transcend their own initial primitive symbol sets.

15.2.5 Creation of New Primitives

Classically, "emergence" has concerned those processes that create new primitives, i.e. properties, behaviours, or functions that are not logical consequences of pre-existing ones (Broad 1925, Morgan 1931, Bergson 1911, Alexander 1927, Clayton 2004). How to create such fundamental novelty is the central issue for creative and transcendent emergence.

The most extreme example of emergence concerns the relationship of conscious awareness to underlying material process (Kim 1998, Clayton 2004, Rose 2006). All evidence from introspection, behavioural observation, and neurophysiology suggests that awareness and its specific contents is a concomitant of particular organised patterns of neuronal activity (Koch 2004, Rose 2006). If all experienced, phenomenal states supervene on brain states that are organisations of material processes, and these states in turn depend on nervous systems that themselves evolved, then it follows that there was some point in evolutionary history when conscious awareness did not exist.

This state-of-affairs produces philosophical conundrums. One can deny the existence of awareness entirely on behaviouristic grounds, because it can only be observed privately, but this contradicts our introspective judgement that waking awareness is qualitatively different from sleep or anaesthesia. One can admit the temporal, evolution-enabled appearance of a fundamentally new primitive aspect of the world, a creative emergent view (Alexander 1927, Broad 1925, Morgan 1931), but this is difficult to incorporate within ontological frameworks that posit timeless sets of stable constituents. Or one can adopt a panpsychist view, with Spinoza and Leibniz, that the evolved nervous systems combine in novel ways simple distinctions that are inherent in the basic constituents of matter (Skrbina 2005). Accordingly, we could further divide creative emergence into the appearance of new structural and functional primitives that require epistemological, but not ontological reframing, and appearance of new, transcendent aspects of the world, such as the evolutionary appearance of consciousness, which require both.

Attempting to produce emergent awareness in some artificially constructed system is a highly uncertain prospect, because awareness is accessible only through private observables. One has no means, apart from indirect structural-functional analogy, of assessing success, i.e. whether any awareness has been brought into being. This is why even conscious awareness in animals, which have nervous systems extremely similar to ours, is a matter of lively debate.

More practical than *de novo* creation of new forms of being is the creation of new functions, which are both verifiable and useful to us—creativity as useful novelty. To my mind, the most salient examples of functional emergence involve the evolution of new sensory capabilities in biological organisms. Where previously there may have been no means of distinguishing odours, sounds, visual forms or colours, eventually these sensory capacities evolve in biological lineages. Each new distinction becomes a relative primitive in an organism's life-world, its sensorimotor repertoire.

Combinations of existing sensory distinctions do not create new primitive distinctions. We cannot directly perceive x-rays using our evolution-given senses, no matter how we combine their distinctions. In Sect. 15.3 we outline how evolutionary robotic devices could adaptively evolve their own sensors and effectors, thereby creating new primitives for sensorimotor repertoires.

Over the arc of evolution, the sensorimotor life-worlds of organisms have dramatically expanded. When a new sensory distinction or primitive action appears, the dimensionality of the sensorimotor combinatorial repertoire space increases. In an evolutionary landscape, the effective dimensionality of the fitness surfaces increases as life-worlds become richer and there are more means through which organisms can interact. Theoretical biologist Michael Conrad called this process "extradimensional bypass" (Cariani 2002, Chen and Conrad 1994, Conrad 1998).

The evolution of a new sensorimotor distinction and its dimensional increase can actually simplify problems of classification and decision-making. For gradient-ascending, hill-climbing optimisers, local maxima traps may become saddle points in higher dimensional spaces that open up entirely new avenues for further ascent. In the last decade, workers developing self-organising semantic webs for automated computer search have proliferated features to produce sparse, high-dimensional relational spaces (Kanerva 1988) whose partitioning becomes tractable via regularisation and linear classification techniques.

The senses of animals perform the same fundamental operations as the measurements that provide the observables of scientific models (Pattee 1996, Cariani 2011) and artificial robotic systems. Outside of artificially restricted domains it is not feasible to outline the space of possible sensory distinctions because this space is relational and ill-defined. It is analogous to trying to outline the space of possible measurements that could ever be made by scientists, past and future. Emergent creativity can be said to take place when new structures, functions, and behaviours appear that cannot be accounted for in terms of the previous expectations of the observer. For combinatorial creativity, the observer can see that the novel structures and functions are explicable in terms of previous ones, but for emergent creativity, the observer must enlarge the explanatory frame in order to account for the change. More will be said about emergent creativity and the observer in Sect. 15.4.

In this epistemological view of emergence, surprise is in the eye of the beholder. Because the observer has a severely limited model of the underlying material system, there are processes that can go on within the system that are hidden to direct observation that can qualitatively alter overt behaviour. In biological, psychological, and social systems, internal self-organising, self-complexifying processes can create novel structures and functions that in turn can produce very surprising behaviours. Because the epistemological approach is based on a limited set of macroscopic observables that do not claim any special ontological status, there is no necessary conflict with physical causality or reduction to microscopic variables (where possible). No new or mysterious physical processes or emergent, top-down causalities need to be invoked to explain how more complex organisations arise in physical terms or why they can cause fundamental surprise in limited observers. The novelty that is generated is partially due to internal changes in the system and partially due to the limited observer's incomplete model of the system, such that the changes that occur cause surprise.

15.2.6 Combinatoric and Creative Emergence in Aesthetic Contexts

A first strategy for computational creativity is to use artificial devices to cause creative responses in human participants. In aesthetic realms distinguishing between combinatoric and emergent creativity is made difficult by indefinite spaces of generative possibilities, as well as ambiguities in human interpretation and expectation. Often many prior expectations of individual human observers and audiences may be implicit and subliminal and therefore not even amenable to conscious analysis by the human participants themselves. Nonetheless, to the extent that cultural conventions exist, then it is possible to delineate what conforms to those expectations and what doesn't.

One rule of thumb is that combinatorial creative works operate within a set of stylistic or generative rules that explore new forms within an existing framework. An audience implicitly understands the contextual parameters and constraints of the medium, and the interest is in the play of particular new combinations, motifs, or plot wrinkles. If the element recombinations are trivial, then a piece is perceived as predictable and clichéd. Emergent creative works break conventional, stylistic rules and may violate basic expectations related to the nature of the aesthetic experience itself. One thinks of the Dadaists and the world's reception of Duchamp's urinal as a found-art object.

Usually, the more creatively emergent a production, the fewer the number of people who will immediately understand it, because understanding a new art form or approach requires constructing new conceptual observables and interpretive frames in order to follow radical shifts of meaning. There is stress associated with the uncertainties of orientation and interpretation. For high degrees of novelty, the "shock of the new" causes high degrees of arousal that are in turn are experienced as unpleasant.

The relation between arousal, pleasure, and aesthetics was studied by 19th century psychologists (Machotka 1980). The bell-shaped, Wundt curve plots empirical psychological data related to the relation between arousal (novelty) and experienced pleasure. Low novelty produces boredom, low arousal, and low pleasure, while extremely high novelty produces high arousal that is experienced as unpleasant. Between these two extremes is an optimal level of novelty that engages to produce moderate levels of arousal that are experienced positively. The degree to which a new piece shocks (and its unpleasantness enrages) its audiences is an indication of how many expectations have been violated. An individual's response tells us something about the novelty of the piece in relation to his or her own Wundt curve.

15.3 Creativity in Self-constructing Cybernetic Percept-Action Systems

A second strategy for computational creativity involves expansions of the informational realms in the artificial devices themselves. In this section we consider artificial devices that create their own percept and action primitives, and argue that self-construction guided by internal goals and evaluative faculties is necessary to provide the structural autonomy and implicit dynamics needed to create new useful semantic linkages.

15.3.1 A Taxonomy of Adaptive Devices

The most straightforward way of tackling the problem of how such devices might be designed and built is to consider the different possible kinds of devices that can be conceivably constructed using a set of basic functionalities. A few taxonomies of possible mixed analog-digital adaptive and self-constructing cybernetic devices have been proposed (Cariani 1989; 1992; 1998, de Latil 1956, Pask 1961).

Here we present our own taxonomy of devices in which some functionalities are fixed, while others are adaptively modified or constructed (Figs. 15.2, 15.3, 15.4). It

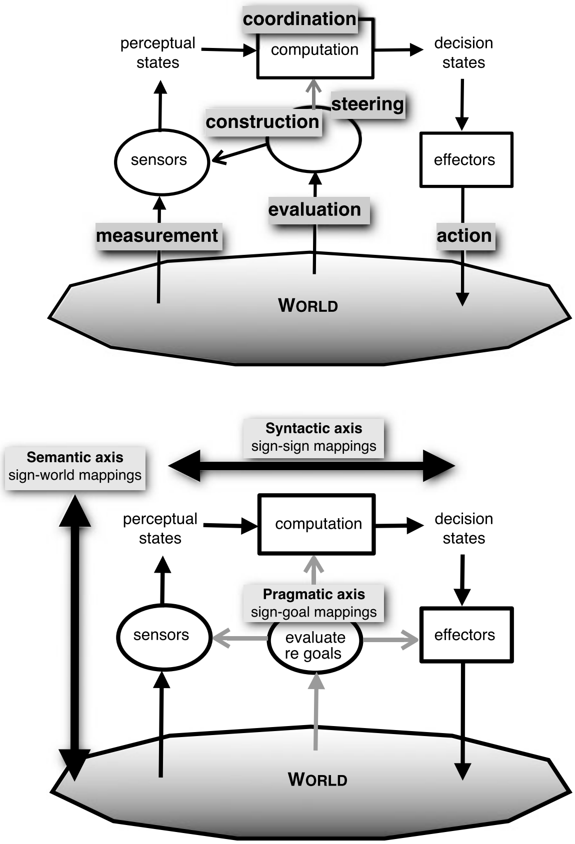


Fig. 15.2 The functional organisation of cybernetic devices. *Top*: basic functionalities and their associated operations. *Bottom*: relation of semiotic dimensions to functionalities and operations

then becomes possible to consider the general capabilities and limitations of various classes of devices that possess varying abilities to adapt and evolve. Although the more structural autonomy or constructive license given to a device, the greater the potential creativity that is permitted, it should be remembered that greater degrees of autonomy and creativity come at the expense of greater complexity and longer periods of adaptive construction and evaluative testing.

The basic functionalities that constitute the functional organisation of adaptive self-constructing cybernetic devices in this taxonomy are *coordination, measurement, action, evaluation, steering*, and *construction* (Fig. 15.2, top). Computational operations here entail coordinative linking of output states with input states, and include memory mechanisms for recording and reading out past inputs. Measurement operations are carried out by an array of sensors that produce symbolic outputs whose values are contingent on the interaction of the sensors with their environs. Actions are carried out by effectors that influence the external world. Effectors produce actions contingent upon internal decisions and commands that are the output of the coordinative part. Steering mechanisms alter particular device states or state-transitions without altering the device's set of accessible states or state-transitions. Construction processes involve adding new coordinations (computations, states, state-transitions), measurements (sensors, observables), actions (effectors), goal states and evaluative criteria, and new construction mechanisms as well. One can think of steering as switching between software alternatives, and construction as the physical construction of new hardware. Steering (switching) processes do not change the effective dimensionality of the system, whereas construction does. Many of these operations could be realised by analog, digital, or mixed analog-digital processes.

These basic functionalities arguably account for the basic operational structure of the observer-actor. There is the cycling of signals from sensors to coordinative elements to effectors (outer loop in the diagram) and "feedback to structure" (inner loops) in which evaluative mechanisms steer the modification and/or construction of hardware (sensors, computational, coordinative structures, effectors).

15.3.2 Semiotics of Adaptive Devices

It is useful to discuss such devices and their creative capabilities in terms of the semiotic triad of Charles Morris, which consists of syntactic, semantic, and pragmatic aspects (Morris 1946, Nöth 1990). Syntactics describes rule-governed linkages between signs; semantics, the relation of signs to the external world; and pragmatics, the relation of signs to purposes (goal states). These different semiotic relations are superimposed on the functional schematic of cybernetic percept-action devices in the bottom panel of Fig. 15.2. The syntactic axis runs horizontally, from sign-states related to sensory inputs to those related to coordinative transformations, and finally to decision states that ultimately lead to actions. The semantic axis runs vertically between the sign-states and the external world, where sensory organs determine world-sign causalities and effectors determine sign-world causalities. The pragmatic axis in the centre covers adaptive relationships between sign states and embedded goals. These are implemented by evaluative and adjustment processes that steer the percept-action linkages that govern behaviour and guide the construction of the device itself.

Some devices have fixed functionalities (stable systems), some can autonomously switch amongst existing alternative states to engage in combinatorial search (combinatoric systems), and some can add functional possibilities to creating new primitives (creative systems). Table 15.1 lists the effects of stable, combinatoric, and creative change for different semiotic relations. Creative emergence in the syntactic realm involves creation of new internal sign-states (or computational states) that enable entirely new mappings between states. Creative emergence in the semantic realm involves creating new observables and actions (e.g. sensors, effectors) that contingently link the outer world with internal states. Creative emergence in the pragmatic realm involves creating new goals and evaluative criteria. Table 15.1 and Fig. 15.3 schematise different classes of devices with respect to their creative capabilities.

Table 15.1 Modes of creativity with respect to semiotic dimensions

Aspect	Primitives	Type of system		
		Stable *Maintain structure*	**Combinatoric** *Search existing possibilities*	**Creative** *Add possibilities*
Syntactic	Sign-states & computations	Deterministic finite-state automata	Adaptive changes in state-transition rules *trainable machines*	Evolve new states & rules *growing automata*
Semantic	Measurements & actions	Fixed sensors & effectors (fixed robots)	Adaptive search for optimal combinations of existing sensors & effectors	Evolve new observables, actions *epistemic autonomy*
Pragmatic	Goals	Fixed goals	Search combinations of existing goals *adaptive priorities*	Evolve new goals *creative self-direction motivational autonomy*

15.3.3 Capabilities and Limitations of Adaptive Devices

One can consider the capabilities and limitations of devices with computational coordinative parts, sensors, effectors, and goal-directed mechanisms for adaptive steering and self-construction (Fig. 15.3). For the sake of simplicity, we will think of these systems as robotic devices with sensors and effectors whose moment-to-moment behaviour is controlled by a computational part that maps sensory inputs to action decisions and motor outputs. In biological nervous systems these coordinative functions are carried out by analog and mixed analog-digital neural mechanisms.

Purely computational devices (top left) deterministically map symbolic, input states to output states, i.e. they are formally equivalent to deterministic finite state automata. As they have no non-arbitrary linkages to the external world, their internal states have no external semantics save those that their human programmer-users assign to them. Because their computational part is fixed and functionally stable, such devices are completely reliable. However, they are not creative in that they cannot autonomously generate either new combinations (input-output mappings) or new primitives (sign states).

Some of the functional limitations of formal systems and computational devices are due to their purely syntactic nature, that the sign-states lack intrinsic semantics or pragmatics. The signs and operations are meaningless and purposeless, aside from any meanings or purposes that might be imposed on them by their users. Other limitations arise from their fixed nature, that pure computations do not receive contingent inputs from outside the sign-system, and therefore have no means of adaptively adjusting their internal operations—they do not learn.

One might retort that we have all sorts of computers that are constantly receiving updates from external sources and adjust their behaviour accordingly, but the moment a machine acts in manner that depends not only on its initial state and state-transition rules, its behaviour is no longer a pure computation—it is no longer

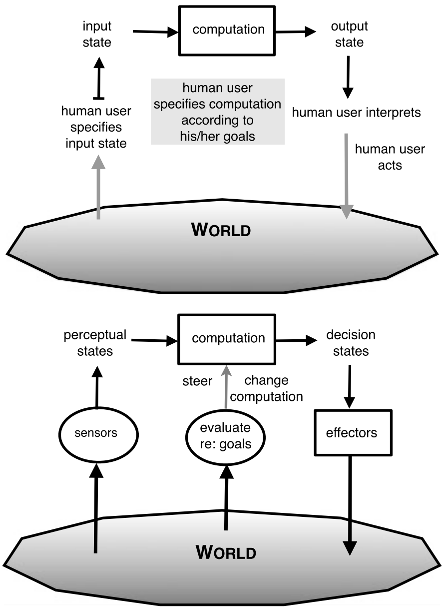


Fig. 15.3 A taxonomy of cybernetic devices. *Top left*: a fixed computational device. *Top right*: a fixed robotic device. *Bottom left*: an adaptive robotic device that modifies its computational input-output mapping contingent on its evaluated performance. *Bottom right*: a robotic device that adaptively constructs its sensing, effecting, and computational hardware contingent on its evaluated performance

emulating the behaviour of a formal system. It is as if one were to perform a calculation, say of the thousandth digit of π, but midway in the calculation the result depends partially on fine variations of the temperature in the room. Only rarely will two such procedures produce the same result, and one now has a process that is the antithesis of a formal procedure. When coupled this way such devices, in formal terms, become machines with inputs from oracles, where the internal workings of the oracle are left ill-defined (Turing 1939). Coupling a deterministic finite state automaton to a sensor that makes measurements converts the composite device to a finite state oracle machine, a decidedly different kind of beast (Hodges 2008).

Adding measurements are useful for some purposes, such as responding appropriately to changes in an external environment, but highly detrimental to others, such as performing reliable, determinate calculations, where one is interested in the purely logical consequences of the application of specified rules on initial inputs. For these reasons, our physical computing devices have been designed and built, as much as possible, to operate in a manner that is independent of their environs.

Accordingly, one can add fixed sensors and effectors to purely computational devices to create robotic devices (Fig. 15.3, top right) that have behaviours that are qualitatively different from those of formal systems. These kinds of systems, which include animals and artificial robots, have specific perception and action linkages to the external world, thereby endowing their internal states with external semantics.

Here the output productions are actions rather than symbols per se, but these devices are also not creative in that they cannot autonomously generate new behaviours.

One can then add evaluative sensors and steering mechanisms that switch the behaviour of the computational part to produce adaptive computational machines (Fig. 15.3, bottom left). This is the basic high-level operational structure of virtually all contemporary trainable machines that use supervised learning feedback mechanisms (adaptive classifiers and controllers, genetic algorithms, neural networks, etc.). Here the internal states and their external semantics are fixed, such that the evaluative-steering mechanism merely switches input-output (percept-action, feature-decision) mappings using the same set of possible states. This is a form of combinatorial creativity, because the machine searches through percept-action combinations to find more optimal ones.

Consider the case, however, where the evaluation mechanism guides the construction of the hardware of the device rather than simply switching input-output mappings (Fig. 15.3, bottom right). If sensors are adaptively constructed contingent on how well they perform a particular function, then the external semantics of the internal states of the device are now under the device's adaptive control. When a device has the ability to construct itself, and therefore to choose its sensors—which aspects of the world it can detect—it attains a partial degree of *epistemic autonomy*. Such a device can adaptively create its own meanings vis-à-vis the external world. A system is purposive to the extent that it can act autonomously to steer its behavior in pursuit of embedded goals. When it is able to modify its evaluative operations, thereby modifying its goals, it achieves a degree of *motivational autonomy*. Such autonomies depend in turn on structural autonomy, a capacity for adaptive self-construction of hardware.

To summarise, combinatoric creativity in percept-action systems entails an ability to switch between existing internal states (e.g. "software"), whereas creative emergence requires the ability to physically modify material structures (e.g. "hardware") that create entirely new states and state-transitions, sensors, effectors, and/or goals.

15.3.4 Pask's "Organic Analogues to the Growth of a Concept"

The most striking example of a creative emergent device is an adaptive self-constructing electrochemical assemblage that was conceived and fabricated by the brilliant and eccentric British cybernetician Gordon Pask in the late 1950s (Cariani 1989; 1993, Pask 1960; 1961, Bird and Di Paolo 2008, Pickering 2010). Pask demonstrated his device at the *Mechanisation of Thought Processes* conference in London in 1958 and described it in a paper provocatively entitled "Organic Analogues to the Growth of a Concept." (Pask 1959)

The device's purpose was to show how a machine could evolve its own "relevance criteria", i.e. its own external semantic meanings. Current was passed through an array of platinum electrodes immersed in an aqueous ferrous sulphate/sulphuric acid

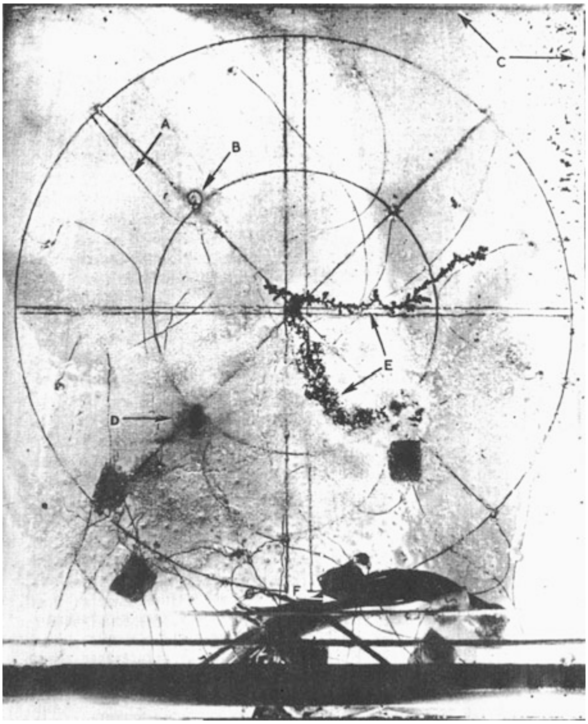


Fig. 15.4 Gordon Pask's creative emergent electrochemical assemblage, from (Pask 1959, p. 919). The photograph looks down on a glass tank containing an aqueous solution of ferrous sulphate and sulphuric acid. Original caption labels: A. Connecting wires for electrodes. B. Platinum pillar electrodes. C. Edges of glass tank containing ferrous sulphate. D. Chemical reaction in progress. E. "Tree" threads being formed. F. Connecting cables

medium, such that iron filaments grew outwards to form bridges between the electrodes (Fig. 15.4). Here the electrodes that extend down into the medium are perpendicular to the plane of the photograph. Iron threads whose conductivity co-varied in some way with an environmental perturbation were rewarded with electric current that caused them to grow and persist in the acidic milieu. Through the contingent allocation of current, the construction of structures could be adaptively steered to improve their sensitivity. The assemblage acquired the ability to sense the presence of sound vibrations and then to distinguish between two different frequencies.

> We have made an ear and we have made a magnetic receptor. The ear can discriminate two frequencies, one of the order of fifty cycles per second and the other on the order of one hundred cycles per second. The "training" procedure takes approximately half a day and once having got the ability to recognise sound at all, the ability to recognise and discriminate two sounds comes more rapidly. I can't give anything more detailed than this qualitative assertion. The ear, incidentally, looks rather like an ear. It is a gap in the thread structure in which you have fibrils which resonate at the excitation frequency." (Pask 1960, p. 261)

In effect, the device had evolved an ear for itself, creating a set of sensory distinctions that it did not previously have. Albeit, in a very limited way, the artificial device automated the creation of new sensory primitives, thereby providing an existence proof that creative emergence is possible in adaptive devices. As Pask explicitly pointed out, one could physically implement an analog perceptron with such an assemblage: the conductances between electrodes in the electrochemical array correspond to connection weights in a connectionist neural network. His intent, however, was to show how a device could produce emergent functionality. Instead of switching inter-electrode connectivities, the thread structures could be steered and selected to become sensitive to other kinds of perturbations, such that they could be tuned with the appropriate rewards. By rewarding conductance changes associated with a particular kind of environmental disturbance, the assemblage could evolve its own sensitivities to the external world.

In the preface to Pask's book *An Approach to Cybernetics* (Pask 1961), Warren McCulloch declared: "With this ability to make or select proper filters on its inputs, such a device explains the central problem of epistemology. The riddles of stimulus equivalence or of local circuit action in the brain remain only as parochial problems." The central, most fundamental problem in epistemology is how to obtain the right observables needed to solve a particular problem. Once these are found, everything else is a matter of searching through the possibilities that these observables afford.

15.3.5 Organisational Closure and Epistemic Autonomy

Creativity and learning both require some degree of autonomy on the part of the system in question. The system needs to be free to generate its own novel, experimental combinations and modifications independent of pre-specification by a designer. The more autonomy given the system, the greater the potential for novelty and surprise on the part of the designer. The less autonomy given, the more reliable and unsurprising the system's behaviour.

When a device gains the ability to construct its own sensors, or in McCulloch's words "this ability to make or select proper filters on its inputs", it becomes *organisationally closed*. The device then controls the distinctions it makes on its external environment, the perceptual categories which it will use. On the action side, once a device acquires the ability to construct its own effectors, it thereby gains control over the kinds of actions it has available to influence the world. The self-construction of sensors and effectors thus leads to attainment of greater *epistemic autonomy* and *enactive autonomy*, where the organism or device itself can become the major determinant of the nature of its relations with the world at large. Structural autonomy and organisational closure guided by open-ended adaptive mechanisms lead to functional autonomy.

These ideas, involving adaptive self-construction and self-production link with many of the core concepts of theoretical biology and cybernetics, such as semantic closure (Pattee 1982; 2008, Stewart 2000), autopoiesis and self-production (Maturana and Varela 1973, Maturana 1981, Varela 1979, Rosen 1991, Mingers 1995), self-modifying systems (Kampis 1991), regenerative signalling systems (Cariani 2000), and self-reproducing automata (von Neumann 1951). Life entails autonomous self-construction that regenerates parts and organisations.

15.4 Recognising Different Types of Creativity

How does one distinguish combinatoric from emergent creativity in practice? This is the methodological problem. The distinction is of practical interest if one wants to build systems that generate fundamental novelty—one needs a clear means of evaluating whether the goal of creating new primitives has been attained.

15.4.1 Emergence-Relative-to-a-Model

Theoretical biologist Robert Rosen (1985) proposed a systems-theoretic, epistemological definition of emergence as the deviation of the behaviour of a material system from the behaviour predicted by a model of that system. At some point the behaviour of a material system will deviate from its predicted behaviour because of processes in the material world that are unrepresented in the model.

This concept can be put into concrete practice by formulating an operational definition. Like the description of an experimental method, an operational definition specifies the procedures by which different observers can reliably make the same classifications, e.g. is a given behaviour emergent, and if so, how? In this case an emergent event is one that violates the expectations of an observer's predictive model. However, simple violations, such as when the engine of one's car fails, are not so interesting or particularly useful, because they can be outward signs of the breakdown of internal mechanisms as much as signs of the evolution of new functions. Instead, we are interested in deviations from expected behaviour that are due to adaptive construction of new functions and qualitatively new behaviours of the system under study.

In the late 1980s, in conjunction with the adaptive systems taxonomy, we developed a systems-theoretic methodology for how one could go about recognising the operations of measurements, computations, and actions from the observed state-transitions of a natural system (Cariani 1989; 1992; 2011). In the process operational definitions were formulated for how these functions can be distinguished from each other, and how changes in a given functionality can be recognised. The method partitions the state-transition structure of the system into regions of state-determined transitions that resemble computations and regions of indeterminate, contingent transitions that resemble measurements and actions.

15.4.2 Tracking Emergent Functions in a Device

Consider the case of an observer following the behaviour of a device (Fig. 15.5, top panel). The observer has a set of observables on the device that allow him/her/it to observe the device's internal functional states and their transitions. Essentially if one were observing a robotic device consisting of sensors, a computational coordinative

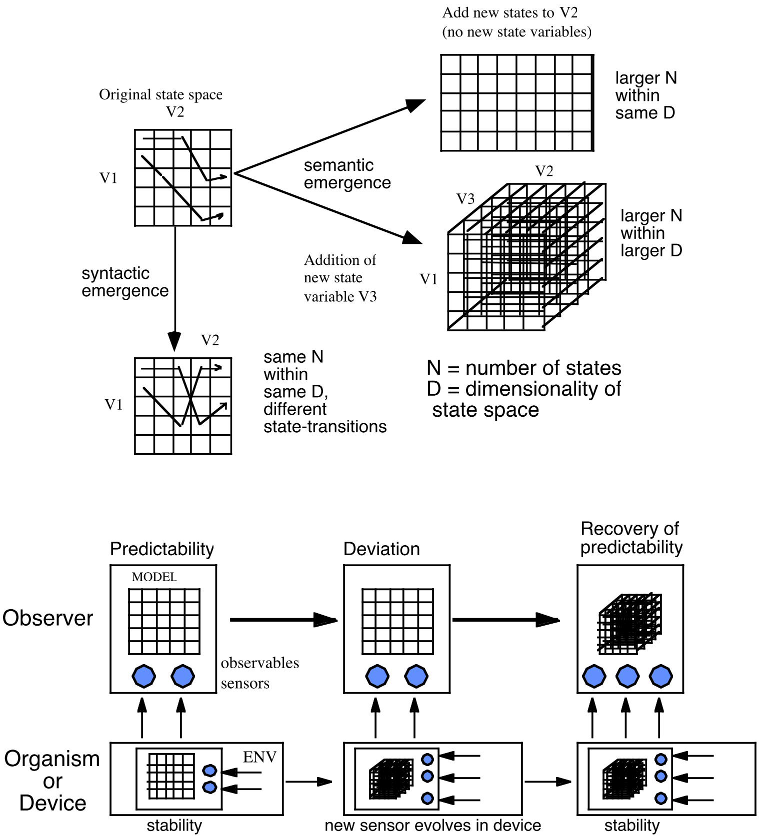


Fig. 15.5 Emergence relative-to-a-model. What changes need to be adopted by an observer in order to continue to predictively track the behaviour of an evolving, complexifying system?

part, and effectors, the computational part of the device that mapped sensory inputs into motor commands would have state-determined transitions.

One can determine if the input-output mapping of the computational part has changed by observing its state-transition structure (Fig. 15.5, top panel). If the computational part is a fixed program, this sensorimotor mapping will remain invariant. If the computational part is switched by some adaptive process, as in a trainable machine, then the sensorimotor mapping will change with training, and a new determinate input-output state transition behaviour will then ensue. From an observer's perspective, the predictive model will fail every time training alters the computational sensorimotor mapping. In order to recover predictability, the observer would have to change the state-transition rules of his or her predictive model. Thus an observer can determine whether the device under observation is performing fixed computations or whether these are being adjusted in some way over time.

Similarly, if the device evolves a new sensor, such that its behaviour becomes dependent on factors that are not registered in the observer's set of measurements, then the observer will also lose predictability. In order to regain predictability, the observer would need to add an extra observable that was roughly correlated with the output of the device's new sensor. Thus if the observer needs to add a sensor to continue to track the device, then it can be inferred that the device itself has effectively evolved a new sensor.

The general principle involves what modifications the observer needs to make in his or her modelling framework to maintain the ability to track the behaviour of the system. If this involves rearrangement of existing states, then the system under observation appears to be combinatorically-emergent. If it requires increasing the dimensionality of his or her observational frame, then the system under observation appears to be creatively emergent. The new dimensions in the observer's complexifying modelling frame coevolve with the creation of new primitives in the observed system.

15.5 New Signal Primitives in Neural Systems

A third, and last strategy for creative emergence is to attempt to understand and emulate the creative processes inside our brains. We humans are the most formidable sources of creativity in our world. Our minds are constantly recombining existing concepts and meanings and also creating entirely new ones. The most obvious process of the construction of new concepts is language acquisition, where children reportedly add 10–15 new word meanings per day to their cognitive repertoires, with the vast majority of these being added without explicit instruction. It seems likely that this "mundane creativity" in children's brains operates through adaptive neural processes that are driven by sensorimotor and cognitively-mediated interactions with the external world (Barsalou and Prinz 1997). Although most of these processes may well fall under the rubric of the combinatorics of syntactic, semantic, and pragmatically grounded inference engines, there are rarer occasions when we experience epiphanies associated with genuinely new ways of looking at the world.

One can contemplate what creation of new signal primitives would mean for neural networks and brains (Cariani 1997). Essentially we want an account of how combinatoric productivity is not only possible, but is so readily and effortlessly achieved in everyday life. We also want explication of how new concepts might be formed that are not simply combinations of previous ones, i.e. how the dimensionality of a conceptual system might increase with experience. How might these creative generativities be implemented in neuronal systems?

We have to grapple with the problem of the primitives at the outset. Even if the brain is mostly a combinatorically creative system, the conceptual primitives need to be created by interactive, self-organising sensorimotor integration processes, albeit constrained by genetically mediated predispositions.

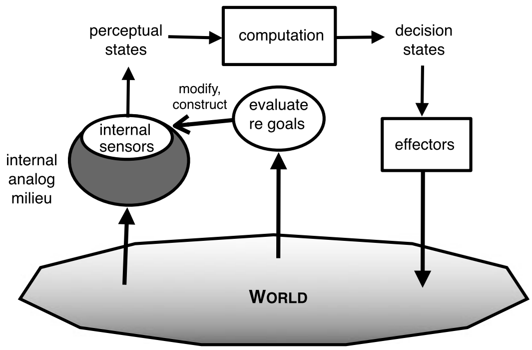


Fig. 15.6 Creation of new semantic primitives by means of internal sensors. Neural assemblies play the role of sensors on an internal milieu of neural activity patterns

15.5.1 New Primitives in Signalling Networks

I came to think about how neural networks might create new primitives from considering how a signalling network might increase its effective dimensionality. The simplest way of conceiving this is to assume that each element in a network is capable of producing and receiving specific signals that are in some way independent of one another, as with signals consisting of tones of different frequencies. A new communications link is established whenever a tone frequency emitted by one element can be detected by another. The effective dimensionality of such a network is related to the number of operating independent communications links. If the elements can be adaptively tuned to send and receive new frequencies that are not already in the network, then new signal primitives with new frequencies can appear over time and with them, new communications links. The dimensionality of the signalling network has thus increased.

One can conceive of the brain as a large signalling network that consists of a large number of neural assemblies of many neurons. If each neural assembly is capable of adaptively producing and detecting specific spatial and temporal patterns of action potential pulses, then new patterns can potentially arise within the system that constitute new signal primitives. In the brain we can think of Hebb's neural assemblies (Hebb 1949, Orbach 1998) as ensembles of neurons that act as internal sensors on an analog internal milieu (Fig. 15.6). The creation of a new neural assembly through an activity-dependent modification of neuronal synapses and axons can be conceived as the equivalent to adding a new internal observable on the system. Here a new concept is a new means of parsing the internal activity patterns within the nervous system. If an adaptively-tuned neural ensemble produces a characteristic pattern of activity that is distinguishable from stereotyped patterns that are already found in the system, then the neural network has created a new signal primitive that can become a marker for the activity of the ensemble and some complex combination of conditions that activates it.

The remainder of the chapter presents an outline of how a neural system might utilise this kind of dimensionally open-ended functional organisation. The nature of the central neural code in the brain is still one of science's biggest unsolved mysteries, and the present situation in the neurosciences is not unlike biology before DNA-based mechanisms of inheritance were understood.

Despite resurgent interest in neuronal temporal synchronies and oscillations, mainstream opinion in the neurosciences still heavily favours neural firing rate codes and their connectionist architectures over temporal codes and timing architectures. For introductory overviews of how connectionist networks operate, see (Arbib 1989; 2003, Horgan and Tienson 1996, Churchland and Sejnowski 1992, Anderson et al. 1988, Boden 2006, Marcus 2001, Rose 2006). Although strictly connectionist schemes can be shown to work in principle for simple tasks, there are still few concrete, neurally-grounded demonstrations of how connectionist networks in real brains might flexibly and reliably function to carry out complex tasks, such as the parsing of visual and auditory scenes, or to integrate novel, multimodal information. We have yet to develop robust machine vision and listening systems that can perform in real world contexts on par with many animals.

In the late 1980s a "connectionism-computationalism" debate ensued about whether connectionist networks are at least theoretically capable of the kinds of combinatorial creativities we humans produce when we form novel, meaningful sentences out of pre-existing lexical and conceptual primitives (Marcus 2001, Horgan and Tienson 1996, Boden 2006, Rose 2006). Proponents of computationalism argued that the discrete symbols and explicit computations of classical logics are needed in order to flexibly handle arbitrary combinations of primitives. On the other hand, the brain appears to operate as a distributed network that functions through the mass statistics of ensembles of adaptive neuronal elements, where the discrete symbols and computational operations of classical logics are nowhere yet to be seen. But difficulties arise when one attempts to use subsymbolic processing in connectionist nets to implement simple conceptual operations that any child can do. It's not necessarily impossible to get some of these operations to work in modified connectionist nets, but the implementations generally do not appear to be robust, flexible, scalable, or neurally plausible (Marcus 2001). It is possible, however, that fundamentally different kinds of neural networks with different types of signals and informational topologies can support classical logics using distributed elements and operations.

Because we have the strong and persistent feeling that we do not yet understand even the basics of how the brain operates, the alternative view of the brain outlined here, which instead is based on multidimensional temporal codes, should be regarded as highly provisional and speculative in nature, more rudimentary heuristic than refined model.

15.5.2 Brains as Networks of Adaptive Pattern-Resonances

Brains are simultaneously communications networks, anticipatory correlation machines, and purposive, semantic engines that analyse their sensory inputs in light of previous experience to organise, direct, and coordinate effective action. Animals with nervous systems are cybernetic, goal-seeking percept-action systems (Arbib 1989, Powers 1973, Sommerhoff 1974, Cariani 2011, Boden 2006, Pickering 2010, McCulloch 1965, Rose 2006). Nervous systems evolved in motile animals in order to better coordinate effective action in rapidly changing situations. Lineages of animals whose nervous systems enhanced survival to reproduction persisted, whereas those with less effective steering mechanisms tended to perish. Like the adaptive devices discussed above in Sect. 15.3, animals have sensory systems that register interactions with the external world, and motor systems that influence events in the world. They have coordinative sensorimotor linkages with varying degrees of complexity, from automatic reflexes to heavily deliberated actions. Brains have embedded goal systems that enforce drive states that steer both perception and action. Embedded goals and steering mechanisms are what make intentional, purposive action possible. Finally brains have reward systems that reconfigure internal circuits to change steering functions and to build neuronal assemblies that facilitate new sensory analyses, cognitive representations, affective and motivational responses, and motor sequencing programs.

Thus globally, the brain is a goal-directed system in which the most primal goal structures, which motivate actions such as breathing, drinking, eating, fleeing, fighting, and mating, are embedded in limbic structures and dopamine-mediated reward systems. Neural goal-seeking mechanisms steer sensory and motor thalamocortical systems to act on system-goals by means of the *basal ganglia*, a large set of brain structures that mediate connections of sensory, cognitive, and motor areas of the cerebral cortex with other limbic goal and reward circuits (Redgrave 2007). The basal ganglia facilitate amplification of goal-relevant sensory information and motor programs by releasing the relevant neuronal subpopulations from inhibition, i.e. "disinhibition".

This arrangement for steering can be thought of as a control brake on a slave braking system; when the control brake is applied, the slave brake is released. If the normal state of recurrent circuits with the inhibitory, slave brakes on is attenuating, but becomes weakly amplifying when the inhibitory slave brakes themselves are inhibited (by the basal ganglia control inputs), then those patterns of sensory, cognitive, and motor activity that are relevant to the system-goals that are currently most pressing will be the ones that will be amplified and regenerated. Neural signal regeneration in these weakly amplifying loops will then continue until some other set of competing goals becomes dominant. The behavioural steering system is reminiscent of Kilmer and McCulloch's reticular formation action selection model (Kilmer and McCulloch 1969) and of Rodney Brooks' subsumption architectures (Brooks 1999) in that the different goals are in constant competition for control over the organism's mode of behaviour.

15.5.3 Regenerative Loops

The basic functionalities involved in perception, coordination, and action may be implemented by global architectures that consist of many reciprocally connected

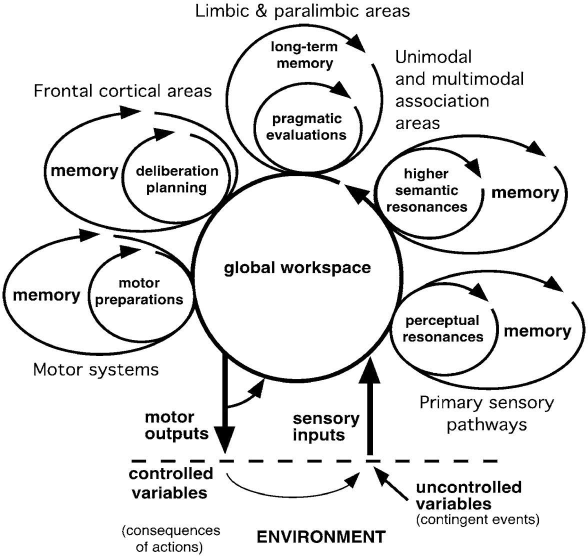


Fig. 15.7 High level functional schematic of the brain as an overlapping and interconnected set of pattern-resonance loops each of which supports a different functionality

neuronal populations. The loops support pattern amplification cycles of neuronal signals that allow signals to be dynamically regenerated within them. The loops might be recurrent connectionist networks (Carpenter and Grossberg 2003, McCulloch 1965) or alternatively, they could be closed circuits that propagate temporal patterns of neuronal activity (Thatcher and John 1977).

That a system has the ability to regenerate alternative sets of neural signals—neural "pattern resonances"—means that the system can support alternative persistent informational states. The system can then keep a set of neuronal signals circulating indefinitely as long as they remain relevant to ongoing goals. This ability to hold signals dynamically is critical for short-term memory, informational integration in global workspaces, and may itself be a requisite for conscious awareness (Baars 1988, Cariani 2000). In dynamical systems terms these mental states are complex signal attractor states that are the stable eigen-behaviour modes of the neural system (Rocha 1996, von Foerster 2003, Cariani 2001b). In analogy with autocatalytic networks that produce the components of living cells, one could think of mutually-supporting signal-regenerations in the loops as an autopoiesis of neural signal productions (Maturana and Varela 1973).

One can sketch out a loose functional organisation based on regenerative processing loops (Fig. 15.7). Sensory information comes into the system through a number of modality-specific sensory pathways. Neural sensory representations are built up in each of the pathways through bottom-up and top-down loops that integrate information in time to form stable perceptual images. When subsequent sensory patterns are similar to previous ones, these patterns are amplified and stabilised; when they diverge, new dynamically-created images are formed from the difference between expectation and input. Such divergences are seen in evoked brain gross electrical potentials as “mismatch negativities” and are modelled as adaptive resonances (Carpenter and Grossberg 2003, Grossberg 1988, Rose 2006).

As successive neural populations respond to the stabilised perceptual images of lower loops, higher semantic resonances are created as objects are recognised and other sets of neural assemblies that are sensitive to their implications are activated. As ensembles of neural assemblies that function as semantic “cognitive nodes” are concurrently excited, evaluative and steering circuits are also being activated and signals associated with hedonic valences become added to the circulating mixture of signals. Both short term and long term memory processes play modulatory roles in the pattern amplifications that go on in the loops, either facilitating or retarding the amplification and buildup of particular circulating patterns. In this conception, some sets of signals are limited to only one or two circuits that reciprocally connect pairs of brain regions, while others of more general relevance are circulated more broadly, through a global workspace. Sets of circuits related to evaluation of the various consequences of action and its planning are activated and finally motor execution programs and pattern generators are engaged to move muscles that act on the external world.

15.5.4 Multidimensional Signals

Although modern neuroscience has identified specific neuronal populations and circuits that subserve all these diverse functions, there is much poorer understanding of how these different kinds of informational considerations might be coherently integrated. Although, most information processing appears to be carried out in local brain regions by neural populations, a given region might integrate several different kinds of signals. Traditional theories of neural networks assume very specific neuronal interconnectivities and synaptic weightings, both for local and long-distance connections. However, flexibly combining different kinds of information from different brain regions poses enormous implementational problems. On the other hand, if different types of information can have their own recognisable signal types, then this coordination problem is drastically simplified. If different signal types can be nondestructively combined to form multidimensional vectors, then combinatorial representation systems are much easier to implement. Communications problems are further simplified if the multiple types of information can be sent concurrently over the same transmission lines without a great deal of destructive interference.

15.5.5 Temporal Coding and Signal Multiplexing

Multiplexing of signals permits them to be combined nondestructively, broadcast, and then demultiplexed by local assemblies that are tuned to receive them. Temporal coding of information in patterns of spikes lends itself to multidimensional signalling, multiplexed transmission, and broadcast strategies for long-distance neural coordinations. The brain can thus be reconceptualised, from the connectionist image of a massive switchboard or telegraph network to something more like a radio broadcast network or even an internet (John 1972).

Neurophysiological evidence exists for temporal coding in virtually every sensory system, and in many diverse parts of the brain (Cariani 1995; 2001c, Miller 2000, Perkell and Bullock 1968, Mountcastle 1967), and at many time scales (Thatcher and John 1977). We have investigated temporal codes for pitch in the early stages of the auditory system (Cariani and Delgutte 1996, Ando and Cariani 2009, Cariani 1999). The neural representation that best accounts for pitch perception, including the missing fundamental and many other assorted pitch-related phenomena, is based on interspike intervals, which are the time durations between spikes in a spike train. Periodic sounds impress their repeating time structure on the timings of spikes, such that distributions of the interspike intervals produced in auditory neurons reflect stimulus periodicities. Peaks in the global distribution of interspike intervals amongst the tens of thousands of neurons that make up the auditory nerve robustly and precisely predict the pitches that will be heard. In this kind of code, timing is everything, and it is irrelevant which particular neurons are activated the most. The existence of such population-based statistical, and purely temporal representations begs the question of whether information in other parts of the brain could be represented this way as well (Cariani and Micheyl 2012).

Temporal patterns of neural spiking are said to be stimulus-driven in they reflect the time structure of the stimulus or stimulus-triggered if they produce response patterns that are unrelated to that time structure. The presence of stimulus-driven patterns of spikes convey to the rest of the system that a particular stimulus has been presented. Further, neural assemblies can be electrically conditioned to emit characteristic stimulus-triggered endogenous patterns that provide readouts that a given combination of rewarded attributes has been recognised (John 1967, Morrell 1967).

The neuronal evidence for temporal coding also provokes the question of what kinds of neuronal processing architectures might conceivably make use of information in this form. Accordingly several types of neural processing architectures capable of multiplexing temporal patterns have been conceived (Izhikevich 2006, Cariani 2004, Chung et al. 1970, Raymond and Lettvin 1978, Pratt 1990, Wasserman 1992, Emmers 1981, Singer 1999).

We have proposed neural timing nets that can separate out temporal pattern components even if they are interleaved with other patterns. They differ from neural networks that use spike synchronies amongst dedicated neural channels, which is a kind of time-division multiplexing. Instead, signal types are encoded by characteristic temporal patterns rather than by “which neurons were active when”. Neural timing nets can support multiplexing and demultiplexing of complex temporal pattern signals in much more flexible ways that do not require precise regulations of neural interconnections, synaptic efficacies, or spike arrival times (Cariani 2001a; 2004).

The potential importance of temporal-pattern-based multiplexing for neural networks is fairly obvious. If one can get beyond scalar signals (e.g. spike counts or firing rates), then what kind of information a given spike train signal contains can be conveyed in its internal structure. The particular input line on which the signal arrives is then no longer critical to its interpretation. One now has an information processing system in which signals can be liberated from particular wires. Although there are still definite neuronal pathways and regions where particular kinds of information converge, these schemes enable processing to be carried out on the level of neuronal ensembles and populations. They obviate the need for the ultra-precise and stable point-to-point synaptic connectivities and transmission paths that purely connectionistic systems require.

15.5.6 Emergent Annotative Tags and Their Uses

Both stimulus-driven and stimulus-triggered temporal response patterns can function as higher-level annotative "tags" that are added to a signal to indicate that it has a particular cognitive attribute. Neural signal tags characteristic of a particular neural assembly would signify that it had been activated. Tags produced by sensory association cortical areas would connote sensory attributes and conjunctions; those produced by limbic circuits would indicate hedonic, motivational, and emotive valences, such that these neural signal patterns would bear pragmatic content. A neural assembly producing a characteristic triggered response pattern could potentially function as a cognitive timing node (MacKay 1987).

Neural assemblies could be adaptively tuned to emit new tag patterns that would mark novel combinations of perceptual, cognitive, conative, and mnemonic activation. New tags would constitute new symbolic neural signal primitives that are associated with new attributes and concepts. The appearance of a particular tag indicates that a particular state-of-affairs has been detected. Formation of new assemblies and associated signal tags would be means by which new, dedicated "perceptual symbols" could be formed from semantically and pragmatically meaningful iconic sensory representations (Barsalou 1999).

The global interconnectedness of cortical and subcortical structures permits widespread sharing of information that has built-up to some minimal threshold of global relevance, in effect creating a global workspace (Baars 1988, Dehaene and Naccache 2001, Rose 2006). In response to a particular triggering stimulus, say a picture of a large dog, the contents of such a global workspace would become successively elaborated over time as the signals produced by different sets of neural assemblies interacted (Fig. 15.8). Successive assemblies would add their own annotations to the circulating pattern associated with various experiences with other similar animals and these would in turn facilitate or suppress the activation of other assemblies.

Linkages between particular sensory patterns and motivational evaluations could be formed that add tags related to previous reward or punishment history, thereby adding to a sensory pattern a hedonic marker. In this way, these complex, elaborated neural signal productions could be imbued with pragmatic meanings which could be conferred on sensory representations that in turn have causal linkages with the external world. Neural signal tags with different characteristics could thus differentiate

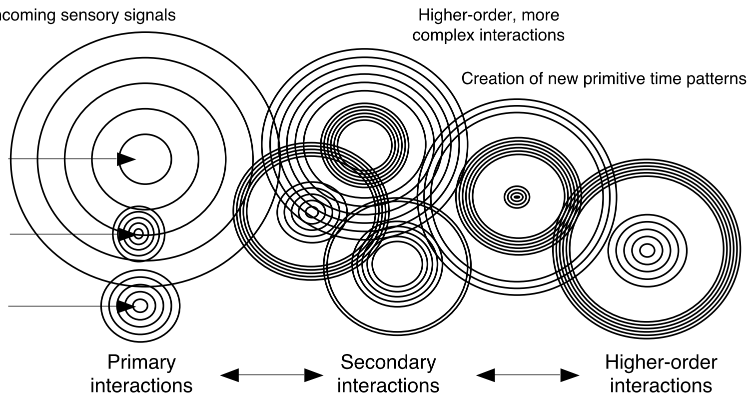


Fig. 15.8 A visual metaphor for the elaboration of evoked neural temporal pattern resonances through successive interaction. The *concentric circles* represent stimulus-driven and stimulus-triggered temporal patterns of spikes produced by neural assemblies, which interact with those of other assemblies

patterns that encode the syntactic, semantic, and pragmatic aspects of an elaborated neural activity pattern. Eventually, the signals in the global workspace would converge into a stable set of neural signals that then sets a context for subsequent events, interpretations, anticipations, and actions.

What we have outlined is an open-ended representational system in which existing primitives can be combined and new primitives formed. Combinatoric creativity is achieved in such a system by independent signal types that can be nondestructively combined in complex composite signals. The complex composites form vectors of attributes that can be individually accessed. Emergent creativity is achieved when new signal types are created through reward-driven adaptive tuning of new neural assemblies. When new signal types are created, new effective signal dimensions appear in the system.

What we do not yet know is the exact nature of the central temporal codes that would be involved, the binding mechanisms that would group attribute-related signals together into objects, the means by which relations between objects could be represented in terms of temporal tags, and how universals might be distinguished from individual instances (Marcus 2001).

Despite its incomplete and highly tentative nature, this high level schematic nevertheless does provide a basic scaffold for new thinking about the generation of novel conceptual primitives in neural networks. We want to provide encouragement and heuristics to those who seek to design mixed analog-digital self-organising artificial brains that might one day be capable of producing the kinds of combinatorial and emergent creativities that regularly arise in our own heads.

Acknowledgements I would like to gratefully thank Margaret Boden, Mark d'Inverno, and Jon McCormack and the Leibniz Center for Informatics for organising and sponsoring the Dagstuhl Seminar on Computational Creativity in July 2009 that made this present work possible.

References

Alexander, S. (1927). *Space, time, and deity*. London: Macmillan & Co.

Anderson, J. A., Rosenfeld, E., & Pellionisz, A. (1988). *Neurocomputing*. Cambridge: MIT Press.

Ando, Y., & Cariani, P. G. E. (2009). *Auditory and visual sensations*. New York: Springer.

Arbib, M. (2003). *The handbook of brain theory and neural networks*. Cambridge, MA: MIT Press.

Arbib, M. A. (1989). *The metaphorical brain 2: neural nets and beyond*. New York: Wiley.

Baars, B. J. (1988). *A cognitive theory of consciousness*. Cambridge: Cambridge University Press.

Barsalou, L. W. (1999). Perceptual symbol systems. *Behavioral and Brain Sciences*, *22*, 577–660.

Barsalou, L. W., & Prinz, J. J. (1997). Mundane creativity in perceptual symbol systems. In T. Ward, S. M. Smith & J. Vaid (Eds.), *Creative thought: an investigation of conceptual structures and processes* (pp. 267–307). Washington: American Psychological Association.

Bergson, H. (1911). *Creative evolution*. New York: Henry Holt, and Company.

Bird, J., & Di Paolo, E. (2008). Gordon Pask and his maverick machines. In P. Husbands, O. Holland & M. Wheeler (Eds.), *The mechanical mind in history* (pp. 185–211). Cambridge: MIT Press.

Boden, M. A. (1990a). *The creative mind*. London: George Weidenfeld and Nicolson Ltd.

Boden, M. A. (1994). *Dimensions of creativity*. Cambridge: MIT Press.

Boden, M. A. (1994b). What is creativity? In M. A. Boden (Ed.), *Dimensions of creativity* (pp. 75–117). Cambridge: MIT Press.

Boden, M. A. (2006). *Mind as machine: a history of cognitive science*. Oxford: Oxford University Press.

Broad, C. D. (1925). *The mind and its place in nature*. New York: Harcourt, Brace and Co.

Brooks, R. A. (1999). *Cambrian intelligence: the early history of the new AI*. Cambridge: MIT Press.

Carello, C., Turvey, M., Kugler, P. N., & Shaw, R. E. (1984). Inadequacies of the computer metaphor. In M. S. Gazzaniga (Ed.), *Handbook of cognitive neuroscience* (pp. 229–248). New York: Plenum.

Cariani, P. (1989). *On the design of devices with emergent semantic functions*. PhD, State University of New York at Binghamton, Binghamton, New York.

Cariani, P. (1992). Emergence and artificial life. In C. Langton, C. Taylor, J. Farmer & S. Rasmussen (Eds.), *Santa Fe institute studies in the science of complexity: Vol. X. Artificial life II* (pp. 775–798). Redwood: Addison-Wesley.

Cariani, P. (1993). To evolve an ear: epistemological implications of Gordon Pask's electrochemical devices. *Systems Research*, *10*(3), 19–33.

Cariani, P. (1995). As if time really mattered: temporal strategies for neural coding of sensory information. *Communication and Cognition—Artificial Intelligence (CC-AI)*, *12*(1–2), 161–229. Reprinted in: K. Pribram (Ed.) (1994). *Origins: brain and self-organization* (pp. 208–252). Hillsdale: Lawrence Erlbaum.

Cariani, P. (1997). Emergence of new signal-primitives in neural networks. *Intellectica*, *1997*(2), 95–143.

Cariani, P. (1998). Towards an evolutionary semiotics: the emergence of new sign-functions in organisms and devices. In G. Van de Vijver, S. Salthe & M. Delpos (Eds.), *Evolutionary systems* (pp. 359–377). Dordrecht: Kluwer.

Cariani, P. (1999). Temporal coding of periodicity pitch in the auditory system: an overview. *Neural Plasticity*, *6*(4), 147–172.

Cariani, P. (2000). Regenerative process in life and mind. In J. L. R. Chandler & G. Van de Vijver (Eds.), *Annals of the New York academy of sciences: Vol. 901. Closure: emergent organizations and their dynamics*, New York (pp. 26–34).

Cariani, P. (2001a). Neural timing nets. *Neural Networks*, *14*(6–7), 737–753.

Cariani, P. (2001b). Symbols and dynamics in the brain. *Biosystems*, *60*(1–3), 59–83.

Cariani, P. (2001c). Temporal coding of sensory information in the brain. *Acoustical Science and Technology*, *22*(2), 77–84.

Cariani, P. (2002). Extradimensional bypass. *Biosystems*, *64*(1–3), 47–53.

Cariani, P. (2004). Temporal codes and computations for sensory representation and scene analysis. *IEEE Transactions on Neural Networks, Special Issue on Temporal Coding for Neural Information Processing*, *15*(5), 1100–1111.

Cariani, P. (2011). The semiotics of cybernetic percept-action systems. *International Journal of Signs and Semiotic Systems*, *1*(1), 1–17.

Cariani, P. A., & Delgutte, B. (1996). Neural correlates of the pitch of complex tones. I. Pitch and pitch salience. II. Pitch shift, pitch ambiguity, phase-invariance, pitch circularity, and the dominance region for pitch. *J. Neurophysiology*, *76*.

Cariani, P., & Micheyl, C. (2012). Towards a theory of infomation processing in the auditory cortex. In D. Poeppel, T. Overath & A. Popper (Eds.), *Human auditory cortex: Springer handbook of auditory research* (pp. 351–390). New York: Springer.

Carpenter, G., & Grossberg, S. (2003). Adaptive resonance theory. In M. Arbib (Ed.), *The handbook of brain theory and neural networks* (pp. 87–90). Cambridge: MIT Press.

Chen, J.-C., & Conrad, M. (1994). A multilevel neuromolecular architecture that uses the extradimensional bypass principle to facilitate evolutionary learning. *Physica D*, *75*, 417–437.

Chung, S., Raymond, S., & Lettvin, J. (1970). Multiple meaning in single visual units. *Brain, Behavior and Evolution*, *3*, 72–101.

Churchland, P. S., & Sejnowski, T. J. (1992). *The computational brain*. Cambridge: MIT Press.

Clayton, P. (2004). *Mind and emergence: from quantum to consciousness*. Oxford: Oxford University Press.

Conrad, M. (1998). Towards high evolvability dynamics. In G. Van de Vijver, S. Salthe & M. Delpos (Eds.), *Evolutionary systems* (pp. 33–43). Dordrecht: Kluwer.

de Latil, P. (1956). *Thinking by machine*. Boston: Houghton Mifflin.

Dehaene, S., & Naccache, L. (2001). Towards a cognitive neuroscience of consciousness: basic evidence and a workspace framework. *Cognition*, *79*(1–2), 1–37.

Emmers, R. (1981). *Pain: a spike-interval coded message in the brain*. New York: Raven Press.

Fodor, J. (1980). On the impossibility of acquiring "more powerful" structures: fixation of belief and knowledge acquisition. In M. Piatelli-Palmarini (Ed.), *Language and learning: the debate between Jean Piaget and Noam Chomsky* (pp. 142–162). Cambridge: Harvard.

Fogel, L. J., Owens, A. J., & Walsh, M. J. (1966). *Artificial intelligence through simulated evolution*. New York: Wiley.

Goodman, N. (1972). A world of individuals. In N. Goodman (Ed.), *Problems and projects* (pp. 155–172). Indianapolis: Bobbs-Merrill. Originally appeared in The Problem of Universals, Notre Dame Press, 1956.

Grossberg, S. (1988). *The adaptive brain, Vols I. and II*. New York: Elsevier.

Hebb, D. O. (1949). *The organization of behavior*. New York: Simon and Schuster.

Hodges, A. (2008). What did Alan Turing mean by "machine"? In P. Husbands, O. Holland & M. Wheeler (Eds.), *The mechanical mind in history* (pp. 75–90). Cambridge: MIT Press.

Holland, J. (1998). *Emergence*. Reading: Addison-Wesley.

Holland, J. H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. Ann Arbor: University of Michigan Press.

Horgan, T., & Tienson, J. (1996). *Connectionism and the philosophy of psychology*. Cambridge: MIT Press.

Izhikevich, E. M. (2006). Polychronization: computation with spikes. *Neural Computation*, *18*(2), 245–282.

John, E. R. (1967). Electrophysiological studies of conditioning. In G. C. Quarton, T. Melnechuk & F. O. Schmitt (Eds.), *The neurosciences: a study program* (pp. 690–704). New York: Rockefeller University Press.

John, E. R. (1972). Switchboard vs. statistical theories of learning and memory. *Science*, *177*, 850–864.

Kampis, G. (1991). *Self-modifying systems in biology and cognitive science*. Oxford: Pergamon Press.

Kanerva, P. (1988). *Sparse distributed memory*. Cambridge: MIT Press.

Kilmer, W., & McCulloch, W. (1969). The reticular formation command and control system. In K. Leibovic (Ed.), *Information processing in the nervous system* (pp. 297–307). New York: Springer.

Kim, J. (1998). *Mind in a physical world: an essay on the mind-body problem and mental causation*. Cambridge: MIT Press.

Kim, J. (2008). Making sense of emergence. In M. Bedau & P. Humphreys (Eds.), *Emergence: contemporary readings in philosophy and science* (pp. 127–153). Cambridge: MIT Press.

Koch, C. (2004). *The quest for consciousness: a neurobiological approach*. Denver: Roberts and Co.

Machotka, P. (1980). Daniel Berlyne's contributions to empirical aesthetics. *Motivation and Emotion*, *4*, 113–121.

MacKay, D. G. (1987). *The organization of perception and action*. New York: Springer.

Marcus, G. F. (2001). *The algebraic mind*. Cambridge: MIT Press.

Maruyama, M. (1977). Heterogenistics: an epistemological restructuring of biological and social sciences. *Cybernetica*, *20*, 69–86.

Maturana, H. R. (1981). Autopoiesis. In M. Zeleny (Ed.), *Autopoiesis: a theory of the living*. New York: North Holland.

Maturana, H., & Varela, F. (1973). Autopoiesis: the organization of the living. In H. Maturana & F. Varela (Eds.), *Boston studies in the philosophy of science: Vol. 42*. *Autopoiesis and cognition (1980)*. Dordrecht: Reidel.

McCulloch, W. S. (1965). *Embodiments of mind*. Cambridge: MIT Press.

Miller, R. (2000). *Time and the brain, conceptual advances in brain research*. Australia: Harwood Academic Publishers/Gordon and Breach.

Mingers, J. (1995). *Self-producing systems*. New York: Plenum Press.

Morgan, L. (1931). *Emergent evolution* (3rd ed.). New York: Henry Holt.

Morrell, F. (1967). Electrical signs of sensory coding. In G. Quarton, T. Melnechuck, & F. Schmitt (Eds.), *The neurosciences: a study program* (pp. 452–469). New York: Rockefeller University Press.

Morris, C. (1946). *Signs, language, and behavior*. New York: George Braziller.

Mountcastle, V. (1967). The problem of sensing and the neural coding of sensory events. In G. Quarton, T. Melnechuk & F. Schmitt (Eds.), *The neurosciences: a study program*. New York: Rockefeller University Press.

Nöth, W. (1990). *Handbook of semiotics*. Indianapolis: Indiana University Press.

Orbach, J. (1998). *The neuropsychological theories of Lashley and Hebb*. Lanham: University Press of America.

Pask, G. (1959). *Physical analogues to the growth of a concept* (pp. 765–794). London: H.M.S.O.

Pask, G. (1960). The natural history of networks. In M. Yovits & S. Cameron (Eds.), *Self-Organizing systems. Proceedings of an interdisciplinary conference*, May 5–6, 1959 (pp. 232–263). New York: Pergamon Press.

Pask, G. (1961). *An approach to cybernetics*. *Science today series*. New York: Harper and Brothers.

Pattee, H. H. (1982). Cell psychology: an evolutionary view of the symbol-matter problem. *Cognition and Brain Theory*, *5*, 325–341.

Pattee, H. H. (1996). The problem of observables in models of biological organizations. In E. L. Khalil & K. E. Boulding (Eds.), *Evolution, order, and complexity* (pp. 249–264). London: Routledge.

Pattee, H. H. (2008). The necessity of biosemiotics: matter-symbol complementarity. In M. Barbieri (Ed.), *Introduction to biosemiotics* (pp. 115–132). Dordrecht: Springer.

Pepper, S. C. (1942). *World hypotheses, a study in evidence*. Berkeley: University of California Press.

Perkell, D., & Bullock, T. (1968). Neural coding. *Neurosciences Research Program Bulletin*, *6*(3), 221–348.

Piaget, J. (1980). The psychogenesis of knowledge and its epistemological significance. In M. Piatelli-Palmarini (Ed.), *Language and learning. The debate between Jean Piaget and Noam Chomsky* (pp. 23–34). Cambridge: Harvard University Press.

Piatelli-Palmarini, M. (1980). How hard is the hard core of a scientific paradigm? In M. Piatelli-Palmarini (Ed.), *Language and learning. The debate between Jean Piaget and Noam Chomsky*. Cambridge: Harvard University Press.

Pickering, A. (2010). *The cybernetic brain: sketches of another future*. Chicago: University of Chicago Press.

Powers, W. (1973). *Behavior: the control of perception*, New York: Aldine.

Pratt, G. (1990). *Pulse computation*. PhD, M.I.T.

Raymond, S., & Lettvin, J. (1978). Aftereffects of activity in peripheral axons as a clue to nervous coding. In S. Waxman (Ed.), *Physiology and pathobiology of axons*. New York: Raven Press.

Redgrave, P. (2007). Basal ganglia. *Scholarpedia*, *2*(6), 1825.

Rocha, L. (1996). Eigen-states and symbols. *Systems Research*, *13*(3), 371–384.

Rose, D. (2006). *Consciousness. Philosophical, psychological, and neural theories*. Oxford: Oxford University Press.

Rosen, R. (1985). *Anticipatory systems*. Oxford: Pergamon Press.

Rosen, R. (1991). *Life itself*. New York: Columbia University Press.

Singer, W. (1999). Neuronal synchrony: a versatile code for the definition of relations? *Neuron*, *24*(1), 49–65. 111–125.

Skrbina, D. (2005). *Panpsychism in the west*. Cambridge: MIT Press.

Sommerhoff, G. (1974). *Logic of the living brain*. London: Wiley.

Stewart, J. (2000). From autopoiesis to semantic closure. *Annals of the New York Academy of Sciences*, *901*, 155–162.

Stewart, R. M. (1969). Electrochemically active field-trainable pattern recognition systems. *IEEE Transactions on Systems Science and Cybernetics*, *SSC-5*(3), 230–237.

Thatcher, R. W., & John, E. R. (1977). *Functional neuroscience, Vol. I. Foundations of cognitive processes*. Hillsdale: Lawrence Erlbaum.

Turing, A. (1939). Systems of logic based on ordinals. *Proceedings of the London Mathematical Society*, *45*(2), 161–228.

van Fraassen, B. C. (1980). *The scientific image*. Oxford: Oxford University Press.

Varela, F. (1979). *Principles of biological autonomy*. New York: North Holland.

von Foerster, H. (2003). *Understanding understanding: essays on cybernetics and cognition*. New York: Springer.

von Glasersfeld, E. (1992). Aspects of constructivism: Vico, Berkely, Piaget. In M. Ceruti (Ed.), *Evoluzione e conoscenza*, Lubrina, Bergamo, Italy (pp. 421–432). Reprinted in von Glasersfeld, *Key works of radical constructivism* (pp. 421–429).

von Glasersfeld, E. (2007). Cybernetics and the theory of knowledge. In M. Larochelle (Ed.), *Key works in radical constructivism* (pp. 153–169). Rotterdam: Sense Publishers.

von Neumann, J. (1951). The general and logical theory of automata, in. In L. A. Jeffress (Ed.), *Cerebral mechanisms of behavior (the Hixon symposium)* (pp. 1–41). New York: Wiley.

Wasserman, G. S. (1992). Isomorphism, task dependence, and the multiple meaning theory of neural coding. *Biological Signals*, *1*, 117–142.

Part IV
Epilogue

Chapter 16
Computers and Creativity: The Road Ahead

Jon McCormack and Mark d'Inverno

Abstract This final chapter proposes a number of questions that we think are important for future research in relation to computers and creativity. Many of these questions have emerged in one form or another in the preceding chapters and are divided into four categories as follows: how computers can enhance human creativity; whether computer art can ever be properly valued; what computing can tell us about creativity; and how creativity and computing can be brought together in learning.

16.1 Where to From Here?

At the end of the book it seems important to consider the most critical questions that have arisen whilst editing the preceding chapters. Throughout the book, a broad range of views on computers and creativity have been expressed. Some authors argue that computers are potentially capable of exhibiting creative behaviours, or of producing artefacts which can be evaluated in a similar context as human artworks. Others believe that computers will never exhibit autonomous creativity and that we should think of computers and creativity only in the sense of how computers can stimulate creativity in humans. A number of authors even downplay the concept of creativity itself, seeing other approaches such as training and practice, or social mechanisms, as more central in understanding the creation of novel artefacts.

Whilst there is some disagreement about the relationship between computers and creativity, there is a general consensus that computers can transform and inspire human creativity in significantly different ways than any other artificial or human made device. The range of possibilities is evident in this volume, which contains many exciting efforts describing the computer's use in developing art practices, music composition and performance.

J. McCormack (✉)
Centre for Electronic Media Art, Monash University, Caulfield East, Victoria 3145, Australia
e-mail: Jon.McCormack@monash.edu

M. d'Inverno
Department of Computing, Goldsmiths, University of London, London, UK
e-mail: dinverno@gold.ac.uk

J. McCormack, M. d'Inverno (eds.), *Computers and Creativity*,
DOI 10.1007/978-3-642-31727-9_16,

Nevertheless, on the broad issue of how computers relate to creativity, we are still left with many more questions than we have answers. This final chapter contains a selective cross-section of what we think are the twenty-one most important questions, many of which are raised in one form or another in the preceding chapters. Whilst all these questions are clearly interrelated and overlapping, we have categorised them into four topics: (i) how computers can enhance human creativity, (ii) whether computer art can ever be properly valued, (iii) what computing and computer science can tell us about creativity, and finally—while not covered specifically in this book but an important motivation for future research—(iv) how creativity and computing can be brought together in learning.

I **How Can Computers Enhance Human Creativity?**

- i No one likes software that makes simplistic assumptions about what we mean or are trying to do (think of the failed Microsoft Word paperclip or automated typing correction). This raises the question: what are the kinds of responses and interactions we desire of computational systems so as to inspire, provoke, and challenge us to develop meaningful creative dialogues with machines, and to have both the confidence in the system and in ourselves?
- ii Relatedly, how can we remain mindful about the ways in which new technology can limit or defer creativity? We are increasingly seeing software developed which is intended to make creative decisions on our behalf. For example, modern digital cameras now take responsibility for many aspects of the creative photographic process, automatically adjusting numerous dependent properties in order to give the "best" picture. Should we be concerned when creative decision making is implicitly transferred to software at the expense of human creative exploration?
- iii Can we re-conceptualise the methods of interaction between computers and people so as to better encourage creative flow and feedback? We have had many years of the mouse, keyboard and screen as the primary interface, but we have now entered the era of networked mobility and surface touch interfaces, where simple hand or body gestures form the locus of interaction. What new ways of enhancing creative exchange are possible if we move beyond the standard mass-market paradigms and consumer technologies?
- iv How can our developing relationship with computers be better understood in order to encourage new opportunities for experiencing both human- and computer-generated creative artefacts?
- v Is there a point at which individual human creativity can no longer be enhanced by technology or society, no matter how sophisticated? A number of recent computational systems have demonstrated a "counterintuitive" design logic that exceeds human designs significantly. These designs were possible for a computer to find, but seemingly impossible for human designers to discover. Will the goal of augmenting or enhancing human creativity always be limited by our cognitive capacity and inherent genetically and socially conferred biases? Do computers face different limitations, or can they exceed areas of human creativity independently as they have begun to do in limited areas of human endeavour?

II **Could Computer Art Ever Be Properly Valued?**

i When is the computer considered to have had "too much" involvement in the process of making art? To what extent is the produced artefact then devalued as a potential work of art because of the amount of automation? Is it right to challenge this perception, and, if so, how can it be challenged?

ii What are the implications of being clearer and bolder about just how much computing is impacting on any creative output?

iii In relation to the previous question, are there ways of revealing the process of computation that would provide an alternative or additional aesthetic to the completed artefact or of the developing partnership between computers and artists in producing their art?

iv Does it even make sense to ask if the same value system that humans use to experience art can be applied to art made by a computer? If not, then is there another value system that we can use to interact more richly and less dismissively with computer generated artefacts?

v What creative authorship can we attribute to a work that is assembled from existing code that has been written by others (who may be anonymous)? There is clearly creativity in a remix or mash up (where different musical fragments are bought together for a specific project), even though we know the person doing the remix was not the original composer of each musical phrase or fragment. With software things are different because the code is generally hidden and is not so distinctively familiar as it is with music, for example. This creates a new and challenging perspective about the ambiguity of authorship in art that is partially or completely produced by software.

III **What Can Computing Tell Us About Creativity?**

i Is autonomous creative thinking beyond the capacity of any machine that we can make now or in the future?

ii Does creativity necessarily involve the creation of useful or appropriate novelty? Relatedly, how relevant is "value" to the definition of creativity? And what kind of value matters most?

iii Broadly, the humanist view values what humans produce above what all other things produce. Does the ability of software to produce unusual and potentially non-human work mean that it can ever be given equal or even greater value? Could we potentially benefit in some way by challenging our value system and rethinking how things have value (and not just to us)?

iv What is the most practical approach to building creative systems? Should we aim to mimic our own creative behaviour, the behaviours we find in nature, or design completely new mechanisms? How can concepts of "emergence" be usefully exploited in designing creative machines? Is it enough for a machine to produce new combinations of existing primitives or does it have to create completely new primitives to be a truly creative system?

v If we could ever define an algorithm that described in detail everything we do as artists, then do we necessarily become limited as artists within that description?

vi The concept of creativity itself has changed significantly over the years. How will the increasing adoption of computers for creative use change the concept of creativity further?

IV **How Does Creativity and Computing Matter to Education?**

i Computing is not seen as a creative subject by the general public or even at schools and universities in many countries around the world. How then can we change the perception of computing, especially in early learning, so that programming is seen as an engaging creative subject in the same way as science, music and the arts? How can we then inspire students to develop their creativity through computing?

ii In asking numerous friends, students and colleagues who are artists and musicians, and who have mastered both their artistic and programming practice, whether artistic creation is more or less creative than programming, nearly all say they are equally creative. Certainly we have never heard anyone say that playing music is creative but programming music software is not, for example. How can we use this kind of personal evidence to persuade people in education and the arts that programming is also a creative act?

iii What kinds of environments provide the right level of feedback, intuition and control to inspire the idea of programming as a creative act in early learning?

iv Can we find new ways of revealing and explaining computational processes where the flow of computation is more readily accessible to an audience? Could that help us in our desire to attract a greater diversity of students into computing?

v Many companies are now beginning to recognise that they want technologists who can think like artists. However, traditional methods of education in mainstream computing that focus exclusively on engineering-based problem solving will not be sufficient for the new challenges of software development. How can we design university computing programs that provide graduates with the necessary knowledge and skills to best achieve their creative potential?

Undoubtably there are many more questions that could easily be posed here, but it's clear to us that a better understanding of how computing impacts upon creativity in all its guises will become increasingly paramount in coming years. Looking back at the last decade, there is little doubt that the most influential new development with computers in this period has been their role in enhancing our social and cognitive space, and it is now social concerns that drive the design of many major computing initiatives. Looking to the future, whilst it is clear that social concerns will remain a driving force in the design of software, it also seems clear that many of the next major innovations in the design of hardware and software will come from attempts to extend our individual and collective creativity. As we set about building these future computing systems, we hope that this book has served to inspire new ideas on the origins, possibilities, and implications of the creative use of computers.

Index

J. McCormack, M. d'Inverno (eds.), *Computers and Creativity*,
DOI 10.1007/978-3-642-31727-9, © Springer-Verlag Berlin Heidelberg 2012