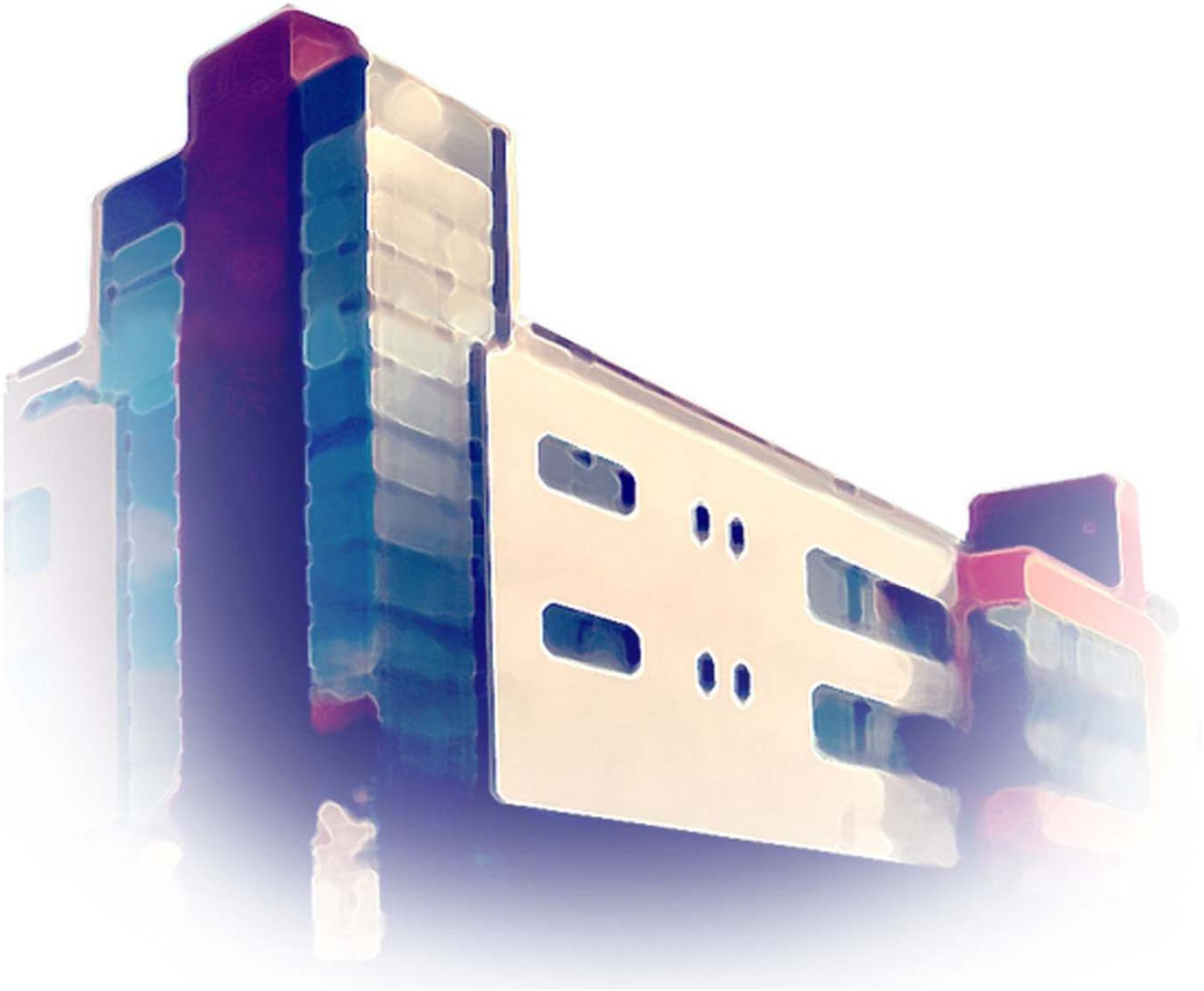
Tarek R. Besold, Kai-Uwe Kühnberger,
Marco Schorlemmer & Alan Smaill (eds.)

Proceedings of the Workshop "Computational Creativity, Concept Invention, and General Intelligence"



PICS

Publications of the Institute of Cognitive Science

Volume 1-2012

ISSN: 1610-5389

Series title: PICS
Publications of the Institute of Cognitive Science

Volume: 1-2012

Place of publication: Osnabrück, Germany

Date: August 2012

Editors: Kai-Uwe Kühnberger
Peter König
Sven Walter

Cover design: Thorsten Hinrichs

Tarek R. Besold
Kai-Uwe Kühnberger
Marco Schorlemmer
Alan Smaill (Eds.)

Computational Creativity, Concept Invention, and General Intelligence

1st International Workshop, C3GI @ ECAI 2012,
Montpellier, France, August 27, 2012

Proceedings

Volume Editors

Tarek R. Besold
Institute of Cognitive Science
University of Osnabrück

Kai-Uwe Kühnberger
Institute of Cognitive Science
University of Osnabrück

Marco Schorlemmer
IIIA-CSIC, Barcelona

Alan Smaill
School of Informatics
University of Edinburgh

This volume contains the proceedings of the workshop “Computational Creativity, Concept Invention, and General Intelligence (C3GI)” at ECAI-2012.

Preface

This volume contains the proceedings of the 1st International Workshop for Computational Creativity, Concept Invention, and General Intelligence held in conjunction with ECAI 2012 in Montpellier, France.

The aim of this workshop is to bring together researchers from different fields of AI working on computational models for creativity, concept formation, concept discovery, idea generation, and their overall relation and role to general intelligence as well as researchers focusing on application areas, like computer-aided innovation.

Although the different approaches to questions concerning the aforementioned aspects do share significant overlap in underlying ideas, the cooperation between the respective communities is still in an early stage, and can greatly profit from interaction and discussion between people from the respective fields, forming trans- and interdisciplinary alliances in research and application.

We are convinced that it is time to revitalize the old AI dream of “Thinking Machines” which now has been almost completely abandoned for decades. And we are not alone in doing so: More and more researchers presently recognize the necessity and feasibility of returning to the original goal of creating systems with human-like intelligence. We think that an event gathering leading researchers in computational creativity, general intelligence, and concept invention is in direct connection to this overall goal, providing and generating additional valuable support in turning the current spirit of optimism into a lasting research endeavor.

August, 2012

Tarek R. Besold
Kai-Uwe Kühnberger
Marco Schorlemmer
Alan Smaill

Program Committee

Committee Co-Chairs

– Tarek R. Besold, University of Osnabrück
– Kai-Uwe Kühnberger, University of Osnabrück
– Marco Schorlemmer, IIIA-CSIC, Barcelona
– Alan Smaill, University of Edinburgh

Committee Members

– Josep Lluís Arcos, IIIA-CSIC, Barcelona
– John Barnden, University of Birmingham
– Gaetano Cascini, Politecnico di Milano
– Jörg Cassens, University of Lübeck
– Hamid Ekbia, Indiana University Bloomington
– Markus Guhe, University of Edinburgh
– Helmar Gust, University of Osnabrück
– Ivan M. Havel, Charles University Prague
– Anders Kofod-Petersen, NTNU Trondheim
– Ulf Krumnack, University of Osnabrück
– Maricarmen Martínez Baldares, University of the Andes, Bogota
– Alison Pease, University of Edinburgh
– Francisco Pereira, University of Coimbra
– Enric Plaza, IIIA-CSIC, Barcelona
– Thomas Roth-Berghofer, University of West London
– Ute Schmid, University of Bamberg
– Petros Stefaneas, National Technical University Athens
– Tony Veale, University College Dublin
– Pei Wang, Temple University Philadelphia

Table of Contents

Computationally Speaking, Does General Intelligence Require Creativity?

Selmer Bringsjord

Rensselaer AI & Reasoning (RAIR) Lab, Rensselaer Polytechnic Institute

Even if we assume that the rich and loaded (but profound and profoundly important) concepts of general intelligence (GI) and creativity have both been rigorously defined to everyone's satisfaction, my title has no fixed, univocal meaning in the absence of how to understand 'Require'. Suppose that we understand this term to align with deductive entailment. Then our question, massaged to reflect our computational/machine-intelligence context, becomes

Q: Computationally speaking, can we deduce that if an agent is general-intelligent, it must be creative as well?

If, turning to psychometrics, we identify general intelligence with the infamous factor-analytic g, and creativity with high performance on the Torrance Tests of Creative Thinking (TTCT), the answer to Q is provably "No," since we can obtain a consistency proof for the conjunction of a computing machine's high performance on a g-loaded test (eg, Raven's Progressive Matrices) and abject failure on TTCT.

If, turning to "success stories" in AI, we identify GI with what Deep Blue or Watson did and can do at the time I write this sentence, and creativity with high performance on TTCT, the answer to Q remains a firm negative.

Nonetheless, I argue that even within the bounds of "psychometric AI" and nuts-and-bolts AI engineering, and even under the deductive interpretation of 'Requre,' the answer to Q is in fact "Yes." If I'm right, it of course follows by modus tollens that if an agent isn't creative, it isn't general-intelligent.

How TheoryMine Creates Novel and (Moderately) Interesting Theorems

Alan Bundy

School of Informatics, University of Edinburgh

TheoryMine (`http://theorymine.co.uk/`) is a company in the novelty gift sector. It generates novel and (moderately) interesting mathematical theorems to which customers can assign a name [1].

The generation of novel theorems is composed of the following stages.

The creation of novel recursive mathematical objects. By a mathematical object we mean a number, vector, matrix, etc. A recursive object is one that is built by the repeated application of constructors to one or more base objects. For instance, the non-negative integers can be built from the base object 0 by repeatedly constructing the next successive number: 1, 2, 3,. By inventing new kinds of constructor, TheoryMine can create new kinds of mathematical objects, thus ensuring that its theorems are novel.

The creation of new recursive functions on these objects. Recursive functions are created by defining how the function behaves on the base objects and then how it behaves on successive objects in terms of its behaviour on earlier ones.

The generation of interesting conjectures. TheoryMine constructs conjectures by composing functions together. It only considers conjectures that are in their simplest form. This also ensures that they cannot be proved merely by further simplification; they require at least one application of mathematical induction. This is a rule of inference that first proves the theorem just for the base cases, e.g., 0, then proves it for successive instances, e.g., n+1, assuming it has already been proved for earlier ones, e.g., n. TheoryMines heuristic for ensuring interestingness is that it only creates simple conjectures that require a non-trivial proof.

Obviously false conjectures are filtered out. Most of these conjectures are false, but can easily be rejected by trying them on a few objects and showing that one of the resulting instances is false.

Proving the remaining theorems. TheoryMine uses a powerful technique, called proof planning, to prove theorems by induction. A proof plan is an outline of the proof that is used to guide the theorem-proving program. In particular, it uses our rippling method to reduce the conjecture when applied to successive objects to the conjecture when applied to earlier objects. Rippling illustrates the impact of appropriate representation of proof methods on successful proof.

References

[1] A. Bundy, F. Cavallo, L. Dixon, M. Johansson, and R. L. McCasland. The theory behind TheoryMine. In *Automatheo*. FLoC, 2010.

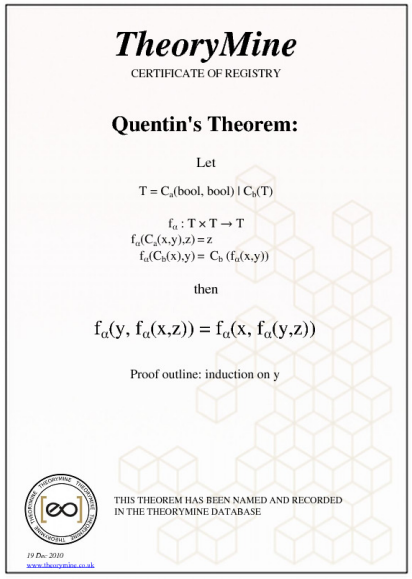


Figure 1: Customers are issued with a certificate that gives the recursive definitions of new objects and functions, together with the statement of the theorem about them. The example certificate shown above was named after Quentin Cooper, who interviewed us for his Radio 4 Material World programme.

Towards Automatic Ideation and Imaginative Reasoning

Simon Colton

Department of Computing, Imperial College London

Ideation is the term used to describe the human processes underlying the generation of ideas for numerous application domains. Ideation exists already in various concept formation approaches within AI research, most notably within Machine Learning applications.

In our group at Imperial (ccg.doc.ic.ac.uk), we have looked at idea generation for various creative applications, including mathematical discovery, and the generation of painted collages, poems, board games, and most recently the production of simple arcade-style games. Our approaches have revolved around formal treatments of concept formation, in addition to informal extraction of ideas from news feeds, Twitter feeds and other social media.

In the first part of the talk, I will introduce some of the creative applications with simple idea generation at their heart that we have been involved with.

In the second part of the talk, I will generalise from these exemplars, and introduce a formalism for a computational approach to constructive ideation. I will extend our recent research in concept formation, and our recent efforts in formalising descriptive models of creative acts in terms of generative acts producing examples, concepts, aesthetics and framing information. The extensions will consider sentiment, emotional affect, tense, ownership, ramifications, renderings and relationships between ideas, and I will introduce a diagrammatic formalism to capture some aspects of idea generation.

I will end the talk by describing a proposed implementation of automatic ideation within the "WhatIf Machine", which not only generates ideas, but also employs imaginative reasoning methods which can flesh the ideas out, justify and ground them, determine ramifications, and rate the ideas with respect to various criteria. The criteria all relate to how useful the idea could be as the basis for a culturally important artefact such as a poem, game, painting, composition or theorem - the production of which is our ultimate aim.

Using grounded theory to suggest types of framing information for Computational Creativity

Alison Pease and **John Charnley** and **Simon Colton**[1]

Abstract. In most domains, artefacts and the creativity that went into their production is judged within a context; where a context may include background information on how the creator feels about their work, what they think it expresses, how it fits in with other work done within their community, and so on. In some cases, such framing information may involve obfuscation in order to add mystery to the work or its creator, which can add to our perception of creativity. We describe a novel method for the analysis of human creativity, using grounded theory. We demonstrate the importance of grounded theory via an ethnographic study of interviews by John Tusa with contemporary artists. By exploring the type of context and background that the artists share, we have developed theories which highlight the importance of areas of framing information, such as motivation, intention, or the processes involved in creating a work. We extend this to consider the role of mystery and obfuscation in framing, by considering what artists *do not say* versus what is explicitly revealed.

1 Introduction

Film, dance, sculpture, music, theatre, architecture, photographs, and visual art are not usually presented to viewers as isolated acts of creativity. Instead, they are accompanied by contextual information such as title, summary, pictures, reviews, resume of the artist, and so on. This context enhances viewers' understanding and appreciation of the work, and enables them to make more informed judgements about the creativity involved. Computational Creativity (CC) has traditionally focused on artefact generation, to the extent that the degree of creativity judged to be in the system is often considered to be entirely dependent on characteristics of the set of artefacts it produces (for instance, see [14]). Very few systems in CC currently generate their own narrative, or framing information. Artefacts are judged either in isolation or in conjunction with a human-produced narrative, such as the name of the system and any scientific papers which describe how it works. We believe that enabling creative software to produce its own framing information is an important direction in the development of autonomously creative systems.

In the following paper, we first describe a novel approach to analysing human creativity, grounded theory (§2). The importance of this methodology is that, we argue, it can be used to derive theories of human creativity which can then be interpreted in computational terms. We then present an ethnographic study of a collection of interviews with artists by arts administrator and journalist John Tusa [16], which is based on grounded theory (§3). Having considered what artists sometimes talk about and ways in which they talk about it, we move on to consider what they *don't* talk about, in our discussion of the role of mystery and obfuscation in framing information (§4). We then discuss our findings and their implications for CC (§5), and describe related work, including proposals for a dually-creative approach to framing [2] – based upon a more informal manual analysis of human framing – and suggest where our ideas extend the literature on evaluating CC (§6). Finally, we make some proposals for future work (§7). Our key contribution is to demonstrate the value of the grounded theory (GT) methodology for CC, by performing an ethnographic analysis, based on GT, of a collection of interviews with artists. Other contributions include highlighting pertinent aspects of framing information, such as the idea that cognitive aspects play an important role, as well as an artist's desire, intention and processes, which is presented within the context of a chronological framework. Artists use metaphors and analogies to emphasise their answers, while leaving some element of mystery, such as avoiding giving too much detail and employing ambiguity productively.

2 Methodology and assumptions

Grounded theory (GT) is a research method within qualitative research which uses data to derive a theory [9]. It was developed in order to reverse the focus on verification of a theory, instead emphasising the prior stage of discovering which concepts and hypotheses are relevant to a particular area. The method consists in a set of heuristic guidelines which suggest a principled way of analysing data at increasing levels of abstraction. It is intended to be theory-neutral, with concepts and categories emerging during data-analysis. GT is a useful methodology for those CC researchers who aim to simulate aspects of human creativity, since it can be used to produce theories of creativity which are grounded in evidence which has been systematically gathered and analysed. GT has five stages:

1. Research problem: find a problem.
2. Data collection: gather a solid body of rich data.
3. Coding: label the data according to what they indicate - this can be done on a line-by-line level or by synthesizing larger amounts of data. Collect codes (annotations) of similar content, thus allowing the data to be grouped into concepts.
4. Categories: group the concepts into categories, and demonstrate relationships between concepts and categories.
5. Theory: Use the concepts and categories to formulate explanations of the subject of research.

Our starting point is that analysing examples of human creativity (using a standard methodology such as GT) and translating resulting ideas and theories into computational terms can suggest useful new directions for CC researchers. We do not make any claims regarding

[1] Computational Creativity Group, Department of Computing, Imperial College, 180 Queens Gate, London SW7 2RH, United Kingdom. Website: www.ccg.doc.ic.ac.uk. Email: apease@doc.ic.ac.uk

'true' (versus perceived) creativity or value, or the role of CC in educating the public on making judgements on creativity. Likewise, we omit discussion of ethical issues, such as whether it is ethical to build a system which 'embellishes' how it has generated its artefacts.

3 Using an approach based on grounded theory to discover types of framing information

We have performed an ethnographic study, based on GT, of artists talking about their work in interviews. Our research problem is to discover what types of framing information accompany a creative artefact. In particular, we are interested in what people say (§'s 3.2 - 3.4), how they say it (§3.5), and what they don't say (§4). We used a combined quantitative and qualitative approach to GT, based on individual words (GT is principally used to analyse qualitative data, however it can also be used for quantitative data, as discussed in [9, chap. XIII]). While in GT the resulting theory can be evaluated according to a set of criteria including fit to data, predictive and explanatory power, logical consistency, clarity and scope, due to the preliminary nature of our work, we do not carry out evaluation at this stage.

3.1 Data collection

Sir John Tusa is a British arts administrator, radio and television journalist, known for his BBC Radio 3 series *The John Tusa Interview*, in which he interviews contemporary artists. These interviews have been reproduced as two books in which he explores the processes of creativity [15, 16]. We have analysed all thirteen interviews in his most recent collection, in order to provide the starting point for a taxonomy of framing information. The interviews feature two filmmakers (Bertolucci, Egoyan), two choreographers (Cunningham, Forsythe), two sculptors (Kapoor, Whiteread), one composer (Goebbels), one theatre director (McBurney), one architect (Piano), one photographer (Rovner) and four artists (Craig-Martin, Gilbert and George, Viola), comprising twelve men and two women (Gilbert and George are interviewed together). The interviews have been transcribed and are in *The John Tusa Interview Archive* on the radio 3 webpages.[2] We used this text as our data. It contains 90,860 words: this breaks down into 20,451 words as questions from Tusa and 70,409 words as responses from the artists. In the following discussion, unless otherwise specified, all page numbers refer to this collection of interviews [16].[3]

3.2 Coding

In order to identify anchors which highlight key points of the data, we used automatic methods to identify words commonly used during both questions and answers in the interviews. We did this via the WORDLE tool,[4] written by Jonathan Feinberg, which found 193 different words which occurred frequently enough to pass its threshold.[5] We then formed concepts to describe these 193 words: *Abstract*, *Artists*, *Cognitive*, *Desire*, *Domain*, *Emotion*, *Intention*, *Life*, *Movement*, *Novelty*, *Number*, *Perception*, *Physical*, *Process*, *Qualifiers*, *Size*, *Social*, *Space*, *Time*, *Value*, *Work*, and a catch-all *Other* concept. Each word was classified as one of these concepts: for example, we classified the words "important", "good", "extraordinary", "interesting", "interested", "great" and "difficult" under *Value* and "new" and "different" under *Novelty*.

3.3 Categories

We used the concepts discovered during coding to suggest categories and then grouped the concepts appropriately and began to consider the relationships between them. We formed the categories CREATIVITY, PEOPLE, ART and PHYSICS. The concepts *Abstract*, *Qualifiers* and *Other* did not fit into any obvious category, so we omit these in the following discussion. We present our concepts and categories in the table below (represented by italics and small capitals, respectively). Each word is followed by a pair of numbers in brackets; this denotes the number of occurrences of the words in the questions and responses, respectively. For example, in the table below, the word "interesting" occurred 25 times in the questions throughout Tusa's interviews, and 60 times in the responses; hence we write "interesting (25;60)". We write the total number of occurrences across both the questions and the responses at the end of each concept. The combined total of all of the occurrences of the 193 words is 15,409.

CREATIVITY.
Value: important (27; 56), good (-; 75), extraordinary (-; 34), interesting (25; 60), interested (10; 34), great (-; 59), difficult (-; 34). TOTAL: 414.
Novelty: new (21; 75), different (25; 98). TOTAL: 219.

PEOPLE.
Emotion: feel (23; 61), felt (-; 51). TOTAL: 135.
Cognitive: think (104; 442), mind (16; -), believe (-; 34), thought (19; 67), understand (-; 42), knew (-; 36), know (49; 545), memory (12; -), remember (-; 41). TOTAL: 1,407.
Intention: mean (77; 161), try (11; 63), trying (-; 57), point (16; 73). TOTAL: 458.
Perception: see (30; 135), sense (23; 57), looking (14; 37), sound (19; 34), look (13; 84), view (14; -), experience (20; 58). TOTAL: 538.
Desire: want (38; 83), wanted (19; 70), like (54; 347), love (-; 51). TOTAL: 662.
Physical: body (12; -), physical (13; -). TOTAL: 25.
Life: living (-; 34), life (21; 86). TOTAL: 141.
Social: people (67; 250), person (16; 42), human (10; -), relationship (11; -), company (13; -), everybody (-; 41), together (12; 63), culture (10; -), audience (14; -). TOTAL: 549.

ART.
Process: order (-; 39), used (-; 47), use (-; 52), made (18; 105), making (21; 90), make (23; 159), way (52; 266), building (19; 42), process (15; -), done (17; 62), change (-; 50), become (-; 38). TOTAL: 1,105.
Domain: ballet (12; 41), film (25; 59), films (12; -), art (36; 137), music (28; 70), dance (21; 38), theatre (14; 57), show (20; 46), image (11; 44), images (19; -), word (15; -), stage (20; 39), video (16; -) classical (12; -). TOTAL: 683.
Work: pictures (-; 36), pieces (18; -), piece (25; 80), studio (-; 40), working (87; 68), work (87; -), works (15; -). TOTAL: 545.
Artists: artists (-; 39), artist (28; 53), dancers (11; -), director (10; -), dancer (12; -). TOTAL: 153.

PHYSICS.
Space: space (10; 46), room (15; 51), place (-; 43), inside (11; -), world (13; 96). TOTAL: 285.
Time: time (48; 187), long (-; 45), years (19; 92), moment (15; 62), end (13; 35), sometimes (13; 43), day (-; 64), back (42; 74), now (95; 135), never (22; 109), start (15; 38), started (12; -), early (12; -), ever (28; -), always (12; -), still (19; -). TOTAL: 1,344.
Number: one (61; 306), two (15; 59), three (11; -), many (-; 89), first (25; 91). TOTAL: 657.
Size: little (-; 88), huge (13; -). TOTAL: 101.
Movement: go (25; 115), come (27; 69), comes (13; -), still (-; 44), went (-; 50), going (54; 119), came (15; 58). TOTAL: 589.

[2] http://www.bbc.co.uk/radio3/johntusainterview/

[3] Although data in GT are typically generated by the researcher with a specific research problem in mind, in our case, since our research problem is to discover what types of framing information accompany a creative artefact, it is appropriate to use data which already exists.

[4] http://www.wordle.net/

[5] Admittedly, this is a very crude method and is not ideal but we see it as a useful starting point. The WORDLE tool omits stop words (words which are frequently used but unimportant, such as "and", or "the"), does not perform stemming and marks capitals as separate occurrences.

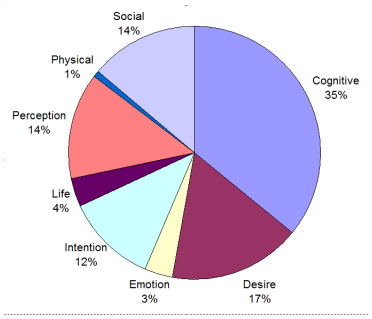


Figure 1. The breakdown of our PEOPLE category, showing concepts *Emotion*, *Cognitive*, *Intention*, *Perception*, *Desire*, *Physical* and *Life*.

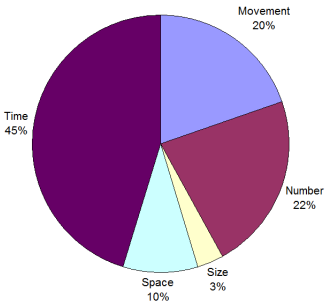


Figure 2. The breakdown of our PHYSICS category, showing concepts *Space*, *Size*, *Time*, *Number* and *Movement*.

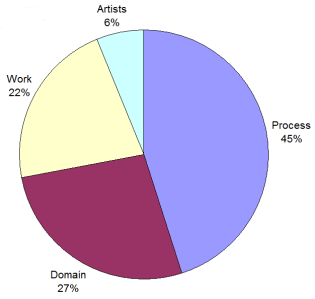


Figure 3. The breakdown of our ART category, showing concepts *Process*, *Domain*, *Work* and *Artists*.

3.4 Theory

In order to determine the relative importance of each category, we use the totals shown at the end of each category. Curiously, words associated most with creativity (which we categorised in the traditional way as the twin goals of value and novelty, [1]) only occurred 6% of the time. This is similar to the number when seen in proportion to the total (633/15,409). Within CREATIVITY, the concept *Value* accounts for 65% and *Novelty* for 35%. Overall, CREATIVITY accounts for 6% of the categories, ART for 25%, PHYSICS for 30% and PEOPLE for 39%. Figures 1 - 3 contain pie charts which display the relative importance of each concept within the latter three categories.

In figure 1, concerning the breakdown of the category PEOPLE, we see that *Cognition* is hugely important. This suggests that thinking plays a significant role in digesting an artwork; thus lending weight to our argument that framing information, rather than simple perception, is an essential component of creativity. *Desire* also accounts for a large proportion of this category. The philosopher Hobbes very strongly associated desire with motivation, which suggests that artists talk about *why* they work. *Intention*, that is, what an artist means by a specific work, forms the next largest category. *Perception* is perhaps discussed less than one would expect, forming merely 3.5% of the discussion (538/15,409). Of these, 29% concerned *Perception* in general (the words "sense" and "experience") (158/538). Of the remaining 380 words, 86% (327/380) concerned sight, and just 14% (53/380) concerned sound. This is, perhaps, a little surprising, in that the artists speaking included one composer, two filmmakers and two choreographers, to whom sound must be a fundamental part of their creations (although, of course, "seeing" can be used to convey understanding as well as referring to vision). No other sense was discussed.

Our PHYSICS category, shown in figure 2, is interesting in that nearly half of the words concern time (1344/2976): this suggests the importance of chronology in framing information. Also of interest in this category are words concerning *Size*: only two were found – "huge", which appeared exclusively in the questions and "little", which appeared exclusively in the responses.

In figure 3, concerning the breakdown of ART into the concepts *Process*, *Domain*, *Work* and *Artists*, we see that almost half of the discussion concerned processes. This indicates that artists talk about *how* they create their work, which may be in contrast to the romantic notion sometimes held of creativity as being inexplicable (this notion may derive back to ancient Greek ideas, in which a creator was seen merely as a messenger from the Muses).

3.5 Style of answers: metaphors and analogies

GT also suggests that we pay attention to styles of language employed. We found that metaphors and analogies were frequently used to convey an answer. Examples include: "The *Singing Sculpture* was like a waterfall that you watched and watched and watched!" (George, of Gilbert and George, p. 115); "... you just see these rooms full of young people completely eating the pictures off the wall" (George, p. 116); "If you learn a language, for example, if you've learned English as your mother tongue, it's very difficult to erase something like that. And ballet was my mother tongue in dance, so you can't erase it from your consciousness." (Forsythe, pp. 93-4); "I'm sort of like a cat, you know. You ever see a cat sit around and stare at things? In that sense I sit around and stare at things ...." (Forsythe, p. 103). Fully fleshed out analogies are also used, for instance, in *The Conformist*,Bernardo Bertolucci draws an analogy between film and Plato's Allegory of the Cave:

> .... Plato says, here you have a cave, you have a group of prisoners in chains sitting at the entrance of the cave, turned towards the bottom of the cave, the interior of the cave. Behind them, there is a fire. People pass with the statues, between the fire and prisoners sitting down and the cave. The fire projects the shadows of the statues on the bottom of the cave. So, I was thinking this is exactly – after all, Plato is about 500 BC! – this is cinema! The fire is the projector with the lamp and the sculpture are like the film passing and the prisoners sitting is the audience and the bottom of the cave is the screen.' (pp. 31-2)

4 The role of mystery in framing information

GT has provided methodological guidance for a theory about what artists say about their work. We should also consider what they *don't* say. An audience may not want to know the full details surrounding a creative act, and in some circumstances might prefer instead that a created artefact and the processes which went into its production are shrouded in some level of mystery. This could be for a number of reasons, including:

- When we look at a created artefact, and learn something about the creative act leading to it, if we cannot imagine how we would have come up with such an innovative idea/process, then we assign more creativity to the producer of the artefact. As a society, we value creative individuals, and hence an audience might want to be given an opportunity to bestow such status onto someone, and this opportunity could be lost if no level of mystique is maintained.
- Certain created artefacts such as paintings and musical compositions are important and interesting to audience members because

they can be interpreted in different ways by different people. Audience members might therefore prefer to be told less about an artefact and its production, so that they can exercise their imagination and incorporate the creative act into their own personal experience.

- Other created artefacts, such as proofs to theorems and many linguistic generations are intended to explicitly communicate some idea, and – especially if the idea is complex or possibly designed for humour – it might be worthwhile for the consumer to work out certain details for themselves. Hence, audiences of such artefacts might prefer to have to work to understand the results of a creative act, because they know they will learn more in this fashion.

We therefore see that it might pay dividends if artists, writers, musicians, and even scientists do not give away full details of their creative processes, leaving instead some room for conjectures about how they might have innovated in such novel ways, which encourages audience members to see the producer as more creative; fill in the gaps and make the dialogue more personal; and exercise their minds in order to fully understand and appreciate the results of a creative act.

Moreover, artists, writers, musicians and scientists know the dividends to be gained by maintaining mystery about their creativity, hence they might go to further lengths to add mystery via obfuscation, providing difficult cultural references, or by giving misleading information. In this sense, a creative act can be seen as the production of a mini-drama, with the production of an element of mystery alongside the creative act being very much an important aspect. Rather than being an irrelevant aside which gets in the way of the study of true creativity, the addition of drama can be seen as an integral part of creativity which could be simulated in software. Audiences want drama, whether within the artefact, or surrounding the creative act producing it. Hence, software developed in CC projects could aim to deliver such drama, and this might go some way towards the acceptance of the idea of software being independently creative in society. Such information could be fictional. For instance, Gilbert and George maintain the illusion of being an inseparable duo of living sculptures:

> *JT:* Will the art of Gilbert and George die when the first one of you dies?
>
> *George:* No I think if we fell under a bus today the pictures will live on, I'm sure of that.
>
> *JT:* But will the artist Gilbert and George die when the first one of you dies?
>
> *George:* We always cross the road together. So maybe we have to be careful! (p. 131)

We consider two further areas of obfuscation below.

4.1 Omitting details

It would be impossible for framing information to include all details concerning the creation of a work, or an artists' personal life (nor would it be desirable: it is possible to both over and under-explain). In [16], Rovner alludes to the notion of an appropriate amount of detail when giving framing information:

> *MR:* There was a name to the kind of water I was drinking. I was wearing very specific clothes, because I'm a very specific ... person ..... You know, I did want not to have any eggs in my sandwich at the day, like always I would never eat eggs! And I wanted the bread to be dark and not white, and many many details going on. (p. 216)

Details will always be omitted:

> *JT:* Now you're not a photographer by training, you began life as a dancer.
>
> *MR:* I began life as a baby, actually. At some point yes I was a dancer, for a few years. (p. 202)

Some details may be omitted because they may lead to an image which the artist wants to avoid. John Tusa discusses this with Kapoor:

> *JT:* ...quite a lot of the time... you wanted to avoid the Indian tag. I was rather shocked when I came across an article from 1998... which said that you're the most successful Indian artist living in the West! Nobody would say that now, so is that why in a way you can talk about the Indian influences much more openly, because you're not pigeonholed?" (p. 159).

Omitting details about technique increases the mystery and can add to the perceived creativity of an act. Consider, for instance, Cunningham's description of his technique of developing dance and music independently:

> *JT:* Why didn't this come out as a mess? That's still a question?
>
> *MC:* No. Because Cage, regardless of what anybody thinks about what he did, was very clear about structures. And these were structures in time. As he said when asked this question, 'Why do you separate the music and the dance?' once Cage replied, 'Well, you see, Merce does his thing and I do mine, and for your convenience we put it together'.
>
> *JT:* Extremely clever elliptical answer.
>
> *MC:* Yes. (p58)

An extreme example of omitting details is artists who keep their identity secret, such as the graffiti artist Banksy.

4.2 Ambiguous terms

The use of multiple meaning is inherent in artefacts in many art forms, such as poetry and visual art. This also applies to framing information. For instance, consider the title of Tracey Emin's 1995 work *Everyone I Have Ever Slept With 1963-1995*. The most obvious interpretation would be to suppose it is about sexual partners, whereas Emin took a more literal interpretation and included various family members, friends and two unborn foetuses. Michael Craig-Martin talks about deliberately misleading people:

> *JT:* Do you mind when people invest them with the symbolic overtones and read non-realistic things into them?
>
> *M C-M:* No, I love it, and I try to add as many false trails of that kind as I possibly can myself. (p. 47)

He goes on to discuss the ambiguity of a filing cabinet in one of his works, which is perceived in multiple ways, depending on the viewer. When displayed in Moscow, the viewers associated the filing cabinet with the KGB: "it's not just because a filing cabinet has a meaning, its meaning is changed by the context of what I've done and where it is." (Craig-Martin, pp. 47-8).

5 Discussion

We have used GT to suggest theories about ways in which artists talk about their work. Analysis of data such as the set of interviews we use suggests a new direction for CC: enabling creative software to generate some of its own framing information. As with human artworks, the appeal of computer creativity will be enhanced by the presence of framing of a similar nature. Few creative systems currently do this, one being an automated poetry generator currently being developed [3]. In addition to creating a poem, this system produces text which describes particular aspects of its poetry that it found appealing.

We found that cognitive aspects such as thinking and knowing play an important role in framing information, and people are interested in an artist's desire or motivation (why did she do X), intention (what did she mean by X?) and processes (how did she do X?). This is all given within a chronological framework (when was a piece started, how long did it take, and so on). Answers are brought to life via metaphors and analogies, while some element of mystery is left, for example by giving an appropriate level of detail and employing ambiguity in a productive way.

Human framing information has previously been analysed by hand, using a more informal approach [2]. Notably, the more systematic approach taken by GT, which we have outlined here, emphasized several of the concepts that were also highlighted as important by that study, such as *Intent* and *Process*.

Intent has been investigated in collage-generation systems [12]. Here, the software based its collage upon events from the news of that day with the aim of inviting the audience to consider the artwork in the context of the wider world around them. This method was later generalised to consider wider combinations of creative systems and more-closely analyse the point in the creative process at which intentionality arose [5].

Details of the creative process are valid aspects of framing information, which are relevant to both computational and human creative contexts. As discussed above (§4.1), there is a notion of an appropriate level of detail: extensive detail may be dull and the appreciation of artefacts is sometimes enhanced by the absence of information about the generative process. Furthermore, as noted in [2], the extent to which information about the process can be perfectly recalled varies between these two contexts. Human fallibility, often means that not all information can be recounted. Similarly, creative software that appeals to transient or dynamic sources, perhaps on the internet, may not be able to retrospectively recover its sources in full.

Not all aspects of framing that we have identified in the ethnographic study in §3 have a clear analogy in CC. For example, concepts within the PEOPLE analysis, such as *Emotion*, *Desire* and *Life* currently have limited meaning in the computational context. This was also noted in [2], although the authors commented how it does make sense to talk of the *career* of a software artist, namely its corpus of work and aspects such as the audience's response, to which it might refer. These difficult-to-capture aspects of the artist's background further support the proposals in [2] of a dually-creative approach to framing in the CC context. This method describes how creative software might be enhanced by the introduction of an automated story-generation system, with the responsibility of producing appropriate framing information. It was further imagined how this might be extended to allow an interactive dialogue, akin to an interview, between a computer artist and its audience. Aspects of the generated story might also feed back into the process of development of the artefact itself, in a cyclic manner. Given the artists' use of metaphor and analogy in the Tusa interviews (§3.5), tools which were able to perform these tasks (see [7, 8]) might be integrated into the storytelling aspect. Our discussions on mystery in §4 suggest that there is a valid place for both fiction and omission within framing information. Additionally, our ethnographical study demonstrated the vast variety of framing information. These both represent significant challenges for contemporary automated story-generation systems.

6 Related work

6.1 Computational accounts of types of framing information

In [2] we presented an informal approach to framing information for CC. In particular, we suggested ways in which motivation, intention and processes could be interpreted in computational terms. In this paper we have given these terms a firmer grounding in data on ways in which humans talk about their creative acts.

6.1.1 Motivation

Many creative systems currently rely upon human intervention to begin, or guide, a creative session and the extent to which the systems themselves act autonomously varies widely. In some sense, the level to which these systems could be considered self-motivating is inversely proportional to the amount of guidance they receive. However, it is possible to foresee situations where this reliance has been removed to such an extent – and the human input rendered so remote – that it is considered inconsequential to the creative process. For instance, the field of Genetic Programming [11] has resulted in software which can, itself, develop software. In the CC domain, software may eventually produce its own creative software which, in turn, produces further creative software, and so forth. In such a scenario, there could be several generations in an overall geneology of creative software. As the distance between the original human creator and the software that directly creates the artefact increases, the notion of self-motivation becomes blurred.

Beyond this, the scope for a system's motivation towards a particular generative act is broad. For example, a suitably configured system may be able to perform creative acts in numerous fields and be able to muster its effort in directions of its own choosing. With this in mind, we can make a distinction between *motivation to perform creative acts in general*, *motivation to create in a particular field* and *motivation to create specific instances*.

Our analysis suggests that, in the human context, the motivation towards a specific field is variously influenced by the life of the artist, their career and their attitudes, in particular towards their field and audience. Several of these are distinctly human in nature and it currently makes limited sense to speak of the *life* or *attitudes* of software in any real sense. By contrast, we *can* speak of the *career* of a software artist, as in the corpus of its previous output. This may be used as part of a process by which a computer system decides which area to operate within. For example, we can imagine software that chooses its field of operation based upon how successful it has previously been in that area. For instance, it could refer to external assessments of its historic output to rate how well-received it has been, focusing its future effort accordingly.

The fact that a computer has no *life* from which to draw motivation does not preclude its use as part of framing information. All those aspects missing from a computer could, alternatively, be simulated. For example, we have seen music software that aims to exhibit characteristics of well-known composers in attempts to capture their compositional style [6]. The extent to which the simulation of human motivation enhances the appeal of computer generated artefacts is, however, still unquantified. The motivation of a software creator may come from a bespoke process which has no basis in how humans are motivated. The details of such a process, and how it is executed for a given instance, would form valid framing information, specific to that software approach.

6.1.2 Intention

The aims for a particular piece are closely related to motivation, described above. A human creator will often undertake an endeavour because of a desire to achieve a particular outcome. Our ethnographic analysis suggests factors, such as attitudes to the field, which contribute to this desire. Certainly, by the fact that some output is produced, every computer generative act displays intent. The aims of the process exist and they can, therefore, be described as part of the framing. In the context of a computer generative act, we might distinguish between *a priori* intent and intentions that arise as part of the generative process. That is, the software may be pre-configured to achieve a particular goal although with some discretion regarding details of the final outcome, which will be decided during the gener-

ative process. The details of the underlying intent will depend upon the creative process applied. For example, as above, software creators might simulate aspects of human intent.

Intent has been investigated in collage-generation systems [12]. Here, the software based its collage upon events from the news of that day with the aim of inviting the audience to consider the artwork in the context of the wider world around them. This method was later generalised to consider wider combinations of creative systems and more-closely analyse the point in the creative process at which intentionality arose [5].

6.1.3 Processes

In an act of human creativity, information about the creative process may be lost due to human fallibility, memory, awareness, and so on. However, in a computational context there is an inherent ability to perfectly store and retrieve information. The majority of creative systems would have the ability to produce an audit trail, indicating the results of key decisions in the generative process. For example, an evolutionary art system might be able to provide details of the ancestry of a finished piece, showing each of the generations in between. The extent to which the generative process can be fully recounted in CC is, nevertheless, limited by the ability to fully recreate the sources of information that played into the generative process. Software may, for instance, use information from a dynamic data source in producing an artefact, and it may not be possible to recreate the whole of this source in retrospect.

One system that produces its own framing is an automated poetry generator currently being developed [3]. In addition to creating a poem, this system produces text which describes particular aspects of its poetry that it found appealing. In order to fully engage with a human audience, creative systems will need to adopt some or all of the creative responsibility in generating framing information.

Details of the creative process are valid aspects of framing information, which are relevant to both computational and human creative contexts. As discussed above, there is a notion of an appropriate level of detail: extensive detail may be dull and the appreciation of artefacts is sometimes enhanced by the absence of information about the generative process.

6.2 Computational Creativity Theory

In [4, 13], two generalisations were introduced with the aim of enabling more precise discussion of the kinds of behaviour exhibited by creative software. The first generalisation places the notion of a generative act, wherein an artefact such as a theorem, melody, artwork or poem is produced, into the broader notion of a *creative act*. During a creative act, multiple types of generative acts are undertaken which might produce framing information, F, aesthetic considerations, A, concepts, C, and exemplars, E; in addition to generative acts which lead to the invention of novel generative processes for the invention of information of types F, A, C and/or E.

The second generalisation places the notion of assessment of the aesthetic and/or utilitarian value of a generated artefact into the broader notion of the impact of a creative act, X. In particular, an assumption was introduced that in assessing the artefacts resulting from a creative act, we actually celebrate the entire creative act, which naturally includes information about the underlying methods, and the framing information, which may put X into various contexts or explain motivations, etc., generally adding value to the generated artefacts over and above their intrinsic value.

The introduction of these two generalisations enabled the FACE and IDEA descriptive models to be introduced as the first in the fledgling formalisation known as *Computational Creativity Theory*. In this paper we have extended this model by further exploring the notion of *framing*.

6.3 Methodology

Although the explicit use of grounded theory as a methodology to derive a theory from data is new to CC, Jordanous [10] uses a corpus linguistic approach on text in academic papers, in order to generate a component-based definition of creativity.

7 Future work and conclusions

Creativity is not performed in a vacuum and the human context gives an artefact meaning and value. This study highlights the value of GT in analysing human creativity and how this motivates and underpins the development of more sensible approaches to the automated generation of framing information, such as complementary story-generation. We intend to continue to develop a theory of framing information and to consider computational interpretations of our theory. We will then formalise these interpretations and develop ways of evaluating them, so that they translate into falsifiable claims that people can make about their creative systems. We expect that these will used to both guide and evaluate progress in this direction. In this way, we envisage that systems which produce artefacts such as choreography, music or visual art will also produce contextual information, which will enhance our understanding and appreciation of the work, and enable us to make more informed judgements about the creativity involved.

We intend to apply GT to other aspects of CC, such as investigating viewers' perceptions and responses to creative work. As with framing, we expect that using GT to inform our analysis will enable us to develop theories that highlight important aspects of different areas of human creativity. This data will be extremely valuable as we seek to further formalise different aspects of CC theory.

Acknowledgements

This work has been funded by EPSRC grant EP/J004049. We are very grateful to our three reviewers for their insightful comments, which have influenced this paper and will continue to be useful as we develop the project further. We would also like to thank Ross Fraser for his technical assistance.

REFERENCES

[1] M. Boden, *The Creative Mind: Myths and Mechanisms (second edition)*, Routledge, 2003.
[2] J Charnley, A. Pease, and S. Colton, 'On the notion of framing in computational creativity', in *Proceedings of the Third International Conference on Computational Creativity*, (2012).
[3] S. Colton, J. Goodwin, and Veale T., 'Full face poetry generation', in *Proceedings of the Third International Conference on Computational Creativity*, (2012).
[4] S. Colton, A. Pease, and J. Charnley, 'Computational creativity theory: The FACE and IDEA descriptive models', in *Proceedings of the Second International Conference on Computational Creativity*, (2011).
[5] M. Cook and S. Colton, 'Automated collage generation – with more intent', in *Proceedings of the Second International Conference on Computational Creativity*, (2011).
[6] D. Cope, *Computer Models of Musical Creativity*, MIT Press, Cambridge, MA, 2006.
[7] D. Gentner, K. Holyoak, and B. Kokinov, *The Analogical Mind: Perspectives from Cognitive Science*, MIT Press, Cambridge, MA, 2001.

[8] *The Cambridge Handbook of Metaphor and Thought*, ed., R. W. Gibbs Jr., Cambridge University Press, Cambridge, UK, 2008.
[9] B. G. Glaser and A. L. Strauss, *The discovery of grounded theory: strategies for qualitative research*, Aldine, Chicago, 1967.
[10] A. Jordanous, 'Defining creativity: Finding keywords for creativity using corpus linguistics techniques', in *Proceedings of the First International Conference on Computational Creativity*, (2010).
[11] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, USA, 1992.
[12] A. Krzeczkowska, J. El-Hage, S. Colton, and S. Clark, 'Automated collage generation – with intent', in *Proceedings of the First International Conference on Computational Creativity*, (2010).
[13] A. Pease and S. Colton, 'Computational creativity theory: Inspirations behind the FACE and the IDEA models', in *Proceedings of the Second International Conference on Computational Creativity*, (2011).
[14] G. Ritchie, 'Some empirical criteria for attributing creativity to a computer program', *Minds and Machines*, **17**, 67–99, (2007).
[15] J. Tusa, *On Creativity*, Methuen, London, 2003.
[16] J. Tusa, *The Janus aspect: artists in the twenty-first century*, Methuen, London, 2006.

Creativity and Conducting: Handle in the CAIRA Project

Simon Ellis • Naveen Sundar G. • Selmer Bringsjord • Alex Haig
Colin Kuebler • Joshua Taylor • Jonas Braasch • Pauline Oliveros • Doug Van Nort
elliss5@rpi.edu • govinn@rpi.edu • selmer@rpi.edu • katadh@sbcglobal.net
kueblc@rpi.edu • tayloj@rpi.edu • braasj@rpi.edu • pauline.oliveros@gmail.com • dvnt.sea@gmail.com
Department of Cognitive Science • Department of Computer Science
The Rensselaer AI & Reasoning (RAIR) Lab • Rensselaer Polytechnic Institute (RPI) • Troy NY 12180 USA

Abstract.
This paper embraces a general-intelligence-based position on machine creativity, and reports on the implementation of that position in the realm of music, in the form of an intelligent agent, Handle, an artificial conductor.

1 "High-" & "Low-Standard" Views of Creativity

In earlier work, Bringsjord, joined by Ferrucci and Bello, argues that while machines can't be *genuinely* creative, at least in the literary sphere they can nonetheless be engineered to *seem* to be creative (3). This two-part position is partly philosophical in nature (based as it is upon *a priori* reasoning), and partly engineeringish (based as it is upon producing a computational artifact capable of generating compelling short-short stories (Brutus.1)). On the philosophical side, in order for a machine to be genuinely creative ($creative_B$), it would need to pass the so-called "Lovelace Test" (LT), which means that what the machine does cannot be anticipated by the designer of this machine (3). On the engineering side, it's enough for the story-telling machine to trick human readers, in Turing-testing-style, into believing that the stories produced by this machine were produced by creative humans ($creativity_T$).

How does Cope define creativity? An explicit answer is supplied in his *Computer Models of Musical Creativity* (2005): he tells us that for him creativity is "[t]he initialization of connections between two or more multifaceted things, ideas, or phenomena hitherto not otherwise considered actively connected" (Cope 2005, 11). Immediately after giving this latitudinarian definition, Cope provides a series of examples of his brand of creativity ($creativity_C$) in action. His last example is the solving of the following puzzle:

> "I have three sons whose ages I want you to ascertain from the following clues. Stop me when you know their ages. One, the sum of their ages is thirteen. Two, the product of their ages is the same as your age. Three, my oldest-in-years son weighs sixty-one pounds."
>
> "Stop," says the second man, "I know their ages."
>
> What are their ages?

Under the assumptions that: (i) the second man is an adult, and hence—in our culture—at least 21 years of age; (ii) the second man couldn't deduce the answer after the second clue; and (iii) the second man knows his own age, it's possible to provide an outright proof that the correct answer is 2, 2, and 9. In an informal nutshell here, the reasoning runs as follows: Of the permutations of three numbers n, m, and k that sum to 13 and have a product that's at least 21, the only two that produce the same product (36) are: 1, 6, 6 and 2, 2, 9. Since in the former case there is no oldest, we are left with the latter as the only possibility. Since, using standard formalisms in logic-based AI (2), we have engineered a machine able to find and certify a formal proof of the argument just given, it's clear that a theorem-prover-based program able to solve this problem would not be $creative_B$. The reason is that the designer of such a computer program wouldn't be surprised in the least when a formal proof expressing the argument is found. In addition, such a program wouldn't be $creative_T$, for the simple reason that cracking such puzzles is precisely the kind of thing humans *expect* computers to be able to do, while humans, save for a select few trained in formal logic, have quite a bit of trouble with such puzzles.

2 Our Piagetian Conception of General Intelligence

Descartes was quite convinced that animals are mechanical machines. He felt rather differently about persons, however: He held that persons, whether of the divine variety (e.g., God, the existence of whom he famously held to be easily provable) or the human, were *more* than mere machines. Someone might complain that Descartes, coming before the likes of Turing, Church, Post, and Gödel, couldn't have had a genuine understanding of the concept of a *computing* machine, and therefore couldn't have claimed that human persons are more than such machines. But while we must admit that Descartes didn't *exactly* have in the mind the concept of a computing machine in the precise manner of, say, a universal Turing machine, what he did have in mind would subsume such modern logico-mathematical devices. For Descartes, a machine was overtly mechanical; but there is a good reason why recursion theory has been described as revolving around what is *mechanically solvable*. A Turing machine, and ditto for its equivalents, are themselves overtly mechanical.

Descartes suggested two tests to use in order to separate mere machines from human persons. The first of these directly anticipates the so-called "Turing Test." The second test is the one that anticipates a sensible suggestion for what the kernel of *general intelligence* is. He wrote:

> [We will] always have two very certain tests by which to recognize that, for all that, they were not real men. The first is, that they could never use speech or other signs as we do when placing our thoughts on record for the benefit of others. ... And the second difference is, that although machines can perform certain things as well as or perhaps better than any of us can do, they infallibly fall short in others, by which means we may discover that they did not act from knowledge, but only for the disposition of their organs. For while reason is a universal instrument which can serve for all contingencies, these organs have need of some special adaptation for every particular action. (Descartes 1911, p. 116)

We now know all too well that machines can perform certain things as well or perhaps better than any of us (witness Deep Blue and Watson, and perhaps, soon enough, say, auto-driving cars that likewise beat the pants off of human counterparts); but we also know that these machines are engineered for specific purposes that are known inside and out ahead of time. We intend Handle to mark significant progress toward the level of proficiency Descartes here refers to as a "universal instrument." This is so because, first, in general, Handle reflects Piaget's focus on *general-purpose* reasoning.

Many people, including many outside psychology and cognitive science, know that Piaget seminally — and by Bringsjord's lights, correctly — articulated and defended the view that mature, general-purpose human reasoning and decision-making consists in processes operating for the most part on formulas in the language of classical extensional logic (e.g., see (9)). Piaget also posited a sequence of cognitive stages through which humans, to varying degrees, pass; we have already referred above to Stages III and IV. How many stages are there, according to Piaget? The received answer is: four; in the fourth and final stage, *formal operations*, neurobiologically normal humans can reason accurately and quickly over formulas expressed at least in the logical system known as first-order logic, $\mathcal{L}_I$.

Judging by the cognition taken by Piaget to be stage-III or stage-IV, the basic scheme is that an agent $\mathcal{A}$ receives a problem P (expressed as a visual scene accompanied by explanatory natural language, auditory sense data, and so on), represents P in a formal language $\mathcal{L}_X$ that is a superset of the language of $\mathcal{L}_I$, producing $[P]$, and then reasons over this representation (along with background knowledge Γ) using at least a combination of some of the proof theory of $\mathcal{L}_1$ and "psychological operators." This reasoning allows the agent to obtain the solution $[S]$. We shall ignore the heterodox operations that Piaget posits in favor of highly expressive intensional logics; so we have intensional operators where Piaget spoke of psychological operators, replete with proof theories, and we will moreover view $[P]$ as a triple (ϕ, C, Q), where ϕ is a (possibly complicated) formula in the language of $\mathcal{L}_X$, C is further information that provides context for the problem, and consists of a set of $\mathcal{L}_X$ formulas, and Q is a query asking for a proof of ϕ from $C \cup \Gamma$. So:

$$[P] = (\phi, C, Q = C \cup \Gamma \vdash \phi?)$$

Our middle-ground position on machine creativity is that a worthy AI/Cog-Sci goal is to engineer a computing machine that is *creative*$_{T^+}$; a machine qualifies here if it's *creative*$_T$, *and* its internal processing conforms to Piaget's conception of general intelligence.

There are other major conceptions of general intelligence distinct from, but potentially complementary to, the explicitly Piagetic-ologicist conception we embrace. For instance, the Polyscheme (5) cognitive architecture is based on the cognitive-substrate hypothesis, which holds that there is a minimal set of core operations from which more elaborate ones blossom. The Handle project is in part based on the attempt to render both computational and precise Piaget's theory of cognitive development from Stage I to IV and beyond, by exploiting the cognitive substrate (and processing power) posited by Polyscheme for Stage I, but we have insufficient space to report this ongoing thrust at the moment.

3 Brutally Quick Overviews of CAIRA & Handle

CAIRA is a **c**reative **a**rtificially-**i**ntuitive and **r**easoning **a**gent, designed and implemented in the context of ensemble music improvisation. The CAIRA system demonstrates creative musicianship that is based on reasoning/logic *and* spontaneity; our aim, in fact, is to better understand the relationship between both modes in the creative process. CAIRA is embedded in an ensemble of musicians, but can also communicate with an individual human performer. It can analyze acoustic environments and respond by producing its own sound, or conduct music via a multi-modal display. The architecture of CAIRA is summed up in Figure 1.

Handle is a microcosmic version of the logic-based parts of CAIRA, but is also a standalone *creative*$_{T^+}$ machine conductor. While prior work on the part of Bringsjord and Ferrucci demonstrated that machine *literary* creativity can be engineered on the basis of formal logic, the use of computational formal logic to perceive and reason over *music* as it is produced in real time by groups of accomplished musicians is more demanding. Can a computing machine "understand" music and reason from that understanding to the prods issued by a great conductor, themselves issued in real time so as to improve the performance in question? While we are confident the answer is Yes, the only way to provide via engineering an existence proof of this affirmative answer is to start humbly, and gradually scale up. Accordingly, Handle is at the moment based on a single pianist playing a short piece, and specifically on the understanding and "conducting" of this playing. A streamlined description of the composite architecture is shown in the Figure 2. A screenshot of Handle in action is shown in Figure 3.[1]

The current version of Handle has two major components. The first component is an audio analysis module running within MATLAB that controls low-level audio and signal processing routines on incoming live or recorded audio. This module then passes information to Handle's musical calculus system, which runs via Common Lisp. The information is passed using a predetermined protocol and format. Handle can currently compute the tempo of live or recorded audio using the system described in (8). This is then passed on to the reasoning system, which in turn determines whether the song is being played at a tempo appropriate for the audience and context. Figure 3 shows Handle responding to a performance of the Prelude in C major from Book 1 of Bach's *The Well-Tempered Clavier* by asking for it to be replayed at a slightly faster tempo. The prelude's score, expressed in the Common Music Notation format, is shown in Figure 4. Future versions of Handle will include ability to understand scores expressed in this format; this reflects the fact that human conductors routinely reflect upon, ahead of time, the scores underlying the performances they conduct.

4 Motivations for a Musical Calculus

While considerable work has been done in modeling music at all levels, from the raw signal-processing stage to representing hierarchical structures, there is a paucity of modeling in the realm of cognitive, social, and doxastic dimensions of music. We provide a small glimpse of the foundations of our approach to constructing a *musical calculus* that can give an account of these three dimensions. Why do we need such a formalism? As we begin to examine the act of musical conducting in more detail, we begin to see why:

> Consider a simple situation in which there is a composer c, a performer p, a listener s, and a conductor h. The composition, or score, in question is *score*. The performance of the score by p is *performance*. Composer c creates *score* with the intention of inducing a single emotional effect *effect*$_1$ in the listener of the piece, s. Performer p has a belief that the composer intends the music to draw out *effect*$_1$ in s, but performer p might want his performance to have effect *effect*$_2$ on s. The conductor

[1] A video of Handle running is available at: http://www.cs.rpi.edu/~govinn/Handle_1.3_demo_video.mov

Figure 1. Overview of CAIRA

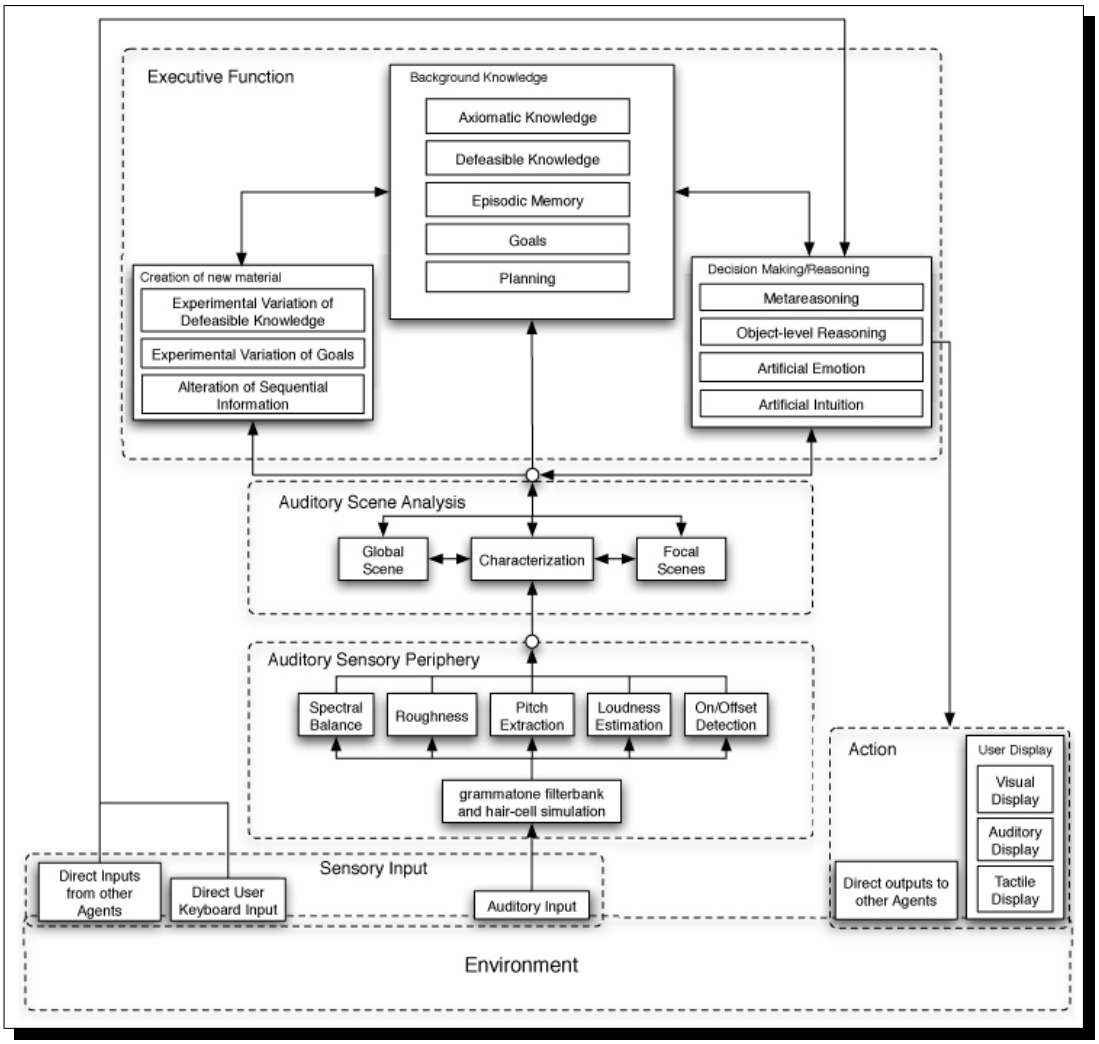


Figure 2. Handle Architecture

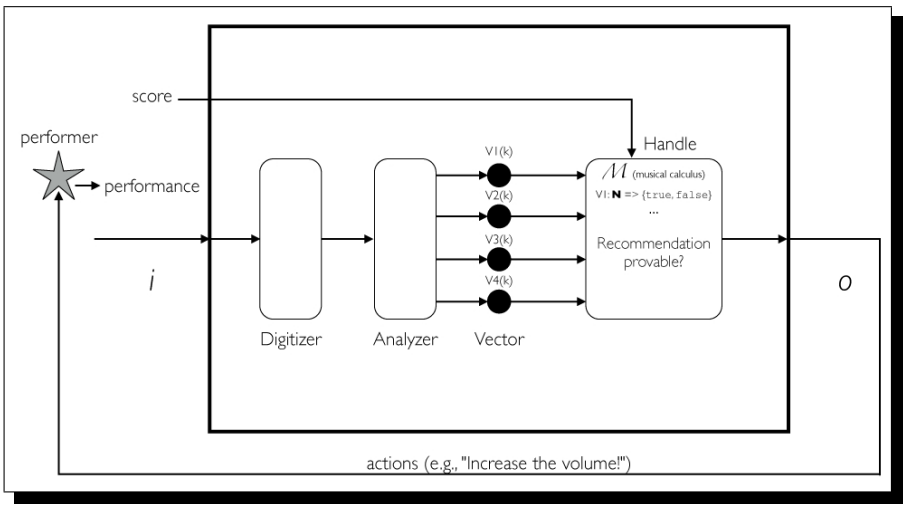


Figure 3. Sample Output from Handle

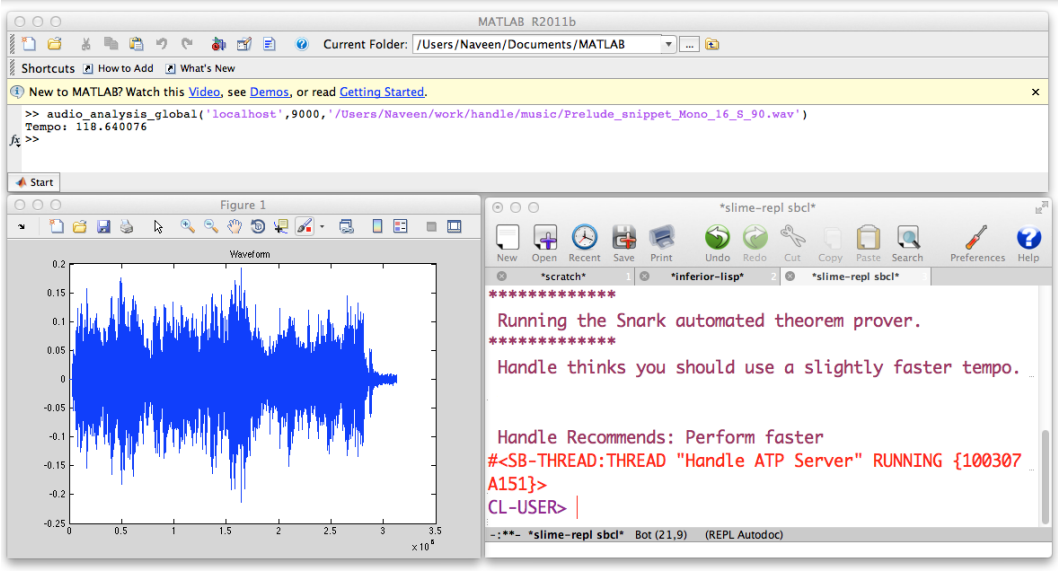


Figure 4. J.S. Bach, *The Well-Tempered Clavier*, Book 1, Prelude 1 in C major, BWV 846: Score Fragment in Common Music Notation

```
(in-package cmn)
(cmn
(size 16) (automatic-rests nil) (staff-separation 2.5)
(system
  brace
  (setf
   s1
   (staff
    bar treble (key c-major) (meter 4 4)
    quarter-rest (g4 s) (c5 s) (e5 s)
    quarter-rest (g4 s) (c5 s) (e5 s)
    bar quarter-rest (a5 s) (d5 s) (f5 s)
    quarter-rest (a5 s) (d5 s) (f5 s)
    bar quarter-rest (g4 s) (d5 s) (f5 s)
    quarter-rest (g4 s) (d5 s) (f5 s)
    bar quarter-rest (g4 s) (c5 s) (e5 s)
    quarter-rest (g4 s) (c5 s) (e5 s)
    bar quarter-rest (a5 s) (e5 s) (a5 s)
    quarter rest (a5 s) (e5 s) (a5 s)
```

h might in turn have beliefs of what the composer and the performer intend, and c might have their own intentions for the performance. Each participant in such a scenario can have further iterative beliefs: for example, the conductor believing what the performer believes the composer intended the performance should be. The conductor should also have an understanding of emotional effects and their inter-relations. For example, h should know that a melancholic effect is incompatible with a joyous effect. Such knowledge of effects should allow the conductor to dynamically alter a performance to elicit compatible effects.

Our music calculus is based on the *Cognitive Event Calculus* ($\mathcal{CEC}$), which we review briefly.

The $\mathcal{CEC}$ is a first-order modal logic. The formal syntax of the $\mathcal{CEC}$ is shown in Figure 5. The syntax specifies sorts S, signature of the function and predicate symbols f, syntax of terms t, and the syntax of sentences ϕ. We refrain from specifying a formal semantics for the calculus as we feel that the possible-worlds approach, though popular, falls short of the *tripartite analysis of knowledge* (Pappas (11)), according to which *knowledge* is a *belief* that is true and justified. The standard possible-worlds semantics for epistemic logics skips over the justification criterion for knowledge.[2] Instead of giving here a full formal semantics for our calculus based in a formalization of justification, we specify a set of inference rules that capture our informal "justification-based" semantics.

We denote that agent a knows ϕ at time t by $\mathbf{K}(a,t,\phi)$. The operators **B**, **P**, **D**, and **I** can be understood to align with belief, perception, desire, and intention, respectively. The formula $\mathbf{S}(a,b,t,\phi)$ captures declarative communication of ϕ from agent a to agent b at time t. Common-knowledge of ϕ in the system is denoted by $\mathbf{C}(t,\phi)$. Common-knowledge of some proposition ϕ holds when every agent knows ϕ, and every agent knows that every agent knows ϕ, and so on *ad infinitum*. The Moment sort is used for representing time points. We assume that time points are isomorphic with $\mathbb{N}$; and function symbols (or functors) $+,-$, relations $>,<,\geq,\leq$ are available.

The $\mathcal{CEC}$ includes the signature of the classic Event Calculus (EC) (see Mueller's (10)), and the axioms of EC are assumed to be common knowledge in the system (1). The EC is a first-order calculus that lets one reason about events that occur in time and their effects on fluents. The $\mathcal{CEC}$ is versatile: it provides a formal account of: mendacity (see Clark (6)), the false-belief task (modeled by Arkoudas and Bringsjord in (1)), and the mirror test for self-consciousness, described in (4). The latter can be consulted to read more about the calculus.

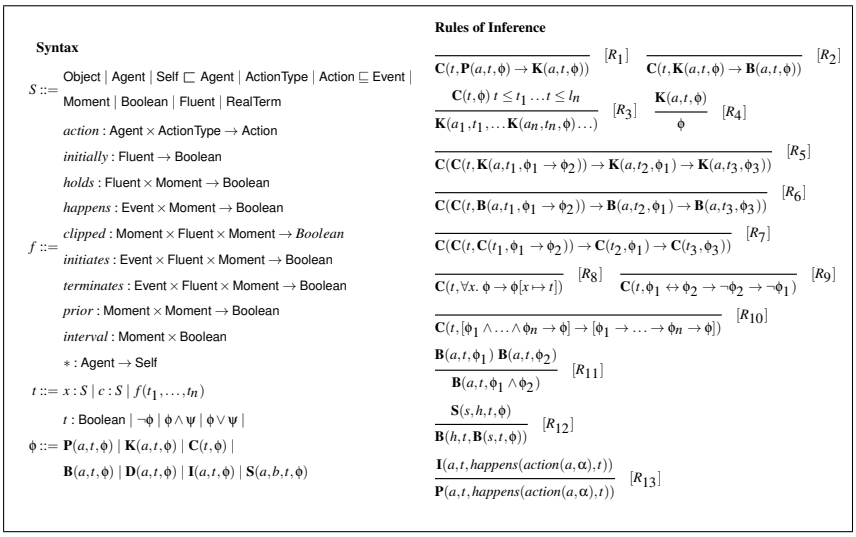
4.1 Toward a Musical Calculus

Our preliminary music calculus has at its core an EC-based hierarchical representation of the syntax and semantics of music. To our knowledge, this work represents the first attempt at modeling the hierarchical structure of music in the event calculus.

While the syntactic hierarchical structure of music has been commented upon in (13; 12), there has been very little study of the compositional or hierarchical semantics in music. Our calculus is intended to remedy this. Our representation also draws upon observations that music exhibits syntactic structure similar to that found in natural language. The alphabet of our music consists of events representing idealized notes combining information about the pitch, time, duration, and timbre of the note. This is exactly similar to the

[2] The possible worlds approach, at least in its standard form, also suffers from allowing logically omniscient agents: agents which know all logically valid sentences.

Figure 5. Cognitive Event Calculus



CHARM representation described in (13). The CHARM system allows much leeway in how such events can be combined together to form hierarchical structures. We impose some constraints that stipulate that such structures must correspond to some abstract syntax:

1. events in music must have some *syntax* with which they can combine with other events in music;
2. events in music must have *semantics* or meaning which interact with the meaning of other events to produce a composite meaning for the whole musical piece.

To this end, we use a representation inspired by the Combinatory Categorial Grammar approach to modeling meaning in natural and formal languages. (See (14) for a quick introduction to the CCG formalism.) Informally, each word in a language is assigned an expression in the typed lambda calculus. The types also specify one of two possible directions in which the lambda function can take arguments. The types allow certain parses of sentences to be ruled out. The meaning of a piece of text is one of the many functional reductions that can be carried out.

The following example illustrates this. The word 'John' has syntactic type *NP*, i.e., noun phrase, and has semantic value *john*; similarly, 'Mary' has syntactic type *NP* and semantic value *mary*. The word 'loves' is a bit more complex. It has syntactic type $(S/NP)\backslash NP$, which means that the word 'loves' combines with an *NP* on the left to give a phrase with type (S/NP). It then combines with an *NP* on the right to give a phrase of type *S*, which is of course a complete sentence. The word 'loves' has a lambda function as its semantic value; this function indicates the operations we just described. The following is a parse tree for "John loves Mary", which results in an analysis that gives us $loves(john, mary)$ as the meaning of the whole sentence at the bottom of the parse.

$$\dfrac{\dfrac{\dfrac{John}{NP : john} \quad \dfrac{loves}{(S/NP)\backslash NP : \lambda xy.loves(x,y)}}{S/NP : \lambda y.loves(john,y)} \quad \dfrac{Mary}{NP : mary}}{S : loves(john, mary)}$$

We observe that the CCG formalism can be adapted to music to enable us to handle semantically rich theories of music which *can* go beyond shallow syntactic forms in music to the deep meaning of musical pieces.

Figure 6. Signature of the Music Calculus

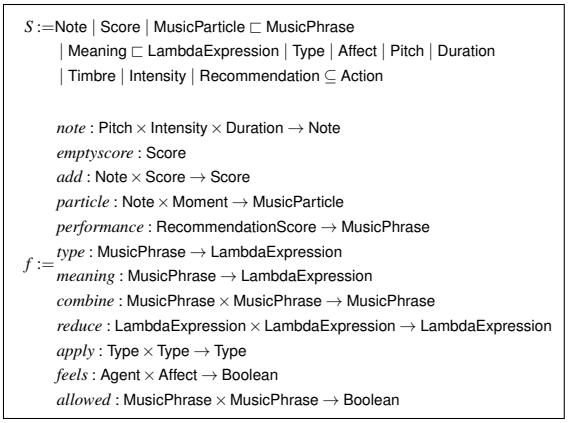


Figure 6 shows a part of the signature of our preliminary music calculus. We have self-explanatory sorts for representing different aspects of the musical universe. Note has its usual standard interpretation. A Score is a sequence of notes formed using the function symbol *add*. A MusicParticle is a note played at a particular moment and can be considered the simplest MusicPhrase. Simpler MusicPhrases combine in myriad ways to form complex MusicPhrases; this is represented using *combine*. The rendition of a Score using a Recommendation in a *performance* results in a MusicPhrase. The music phrases have meanings Meaning which form a subset of the lambda expressions; the meanings combine using *reduce*. The phrases have abstract types Type; the types combine using *apply*. Allowed combinations of music phrases are represented using *allowed*. Recommendations by the conductor are represented using objects from the sort Recommendation. Simple machinery for representing affects is achieved using the sort Affect and the predicate symbol *feels*. We model affects as a subsort of fluents in the event calculus. The way the meaning of a music phrase produces an affect is to be captured by *translates*.

With this syntactic machinery we can account for different agents interpreting a piece of music differently. What might be the meaning of a musical piece? It definitely includes affects produced in a listener. In addition to affects, the meaning can include objective prop-

erties of the music, such as its tempo, which the current version of Handle can process.

The General Problem of Conducting: The general problem of conducting can be stated as follows: Given a score *score* and the composer's intention that the listener s should feel affect a, is there a music phrase p which is the result of performing *score* with the conductor's recommendation r such that the meaning of the phrase p translates into affect a in s?

$$\begin{array}{l}\mathbf{I}(\mathsf{h},t,feels(\mathsf{s},a)) \Rightarrow \\ \exists p : \text{MusicPhrase}\ r : \text{Recommendation}. \\ (\mathbf{B}(\mathsf{h},t,performance(r,score) = p \wedge translates(meaning(p),a)))\end{array}$$

What axioms do we need to enable the conductor to determine his actions? At the minimum, we need a rule specifying combination of the music particles into music phrases. Axiom $\mathcal{A}_1$ states that two music phrases can combine if and only if their syntactic types let them combine. If they combine, the combined phrase has syntax and semantics dictated by the original pieces.

$$\boxed{\mathcal{A}_1}\ \begin{array}{l}\forall m_1\ m_2 : \text{MusicPhrase} \\ allowed(m_1,m_2) \\ \Leftrightarrow \\ \exists m : \text{MusicPhrase}.\ combine(m_1,m_2) = m \\ type(m) = apply(type(m_1),type(m_2)) \wedge \\ meaning(m) = reduce(meaning(m_1),meaning(m_2))\end{array}$$

We need knowledge capturing how the meaning of music translates into affects in agents. Before formalizing this, we need an axiom stating that musical meanings produce affects. Axiom $\mathcal{A}_2$ states that if a piece of music has some meaning, there is an event that causes an affectual response in some person. Here *start* is a defined function symbol giving us the start of a music phrase.

$$\boxed{\mathcal{A}_2}\ \begin{array}{l}\forall m : \text{MusicPhrase}\ \exists\ e : \text{Event}\ a : \text{Affect}\ t : \text{Moment} \\ initiates(e,a,t) \wedge translates(meaning(m),a) \wedge t > start(m)\end{array}$$

Axiom $\mathcal{A}_3$ states a basic property of affects: affects have to be instantiated or associated with agents.

$$\boxed{\mathcal{A}_3}\ \forall a : \text{Affect}\ \exists\ p : \text{Agent}.\ feels(p,a)$$

The *translates* predicate is supposed to capture the translation or production of affects in agents via the semantic properties of music. Upon some reflection, the reader may suspect that we have swept under this predicate symbol the hard-to-formally-model processes that operate in the production of affects. We expect that, when axiomatized, determining whether $translates(m,a)$ holds could be as hard as general-purpose deductive reasoning. Let the axioms governing *translates* be Γ. The problem of conducting can be now stated as finding an r such that:

$$\begin{array}{l}\{\mathcal{A}_1,\mathcal{A}_2,\mathcal{A}_3,\ldots\} \cup \Gamma \vdash \\ \mathbf{I}(\mathsf{h},t,feels(\mathsf{s},a)) \Rightarrow \\ \exists p : \text{MusicPhrase}\ r : \text{Recommendation}. \\ (\mathbf{B}(\mathsf{h},t,performance(r,score) = p \wedge translates(meaning(p),a)))\end{array}$$

Note how the above formulation seems to call upon Piaget's conception of general intelligence in at least two places: in determing whether *translates* holds in any arbitrary case, and in the general structure of the problem.

Acknowledgements: This project is made possible by generous sponsorship from both the NSF (grant no. 1002851) and the John Templeton Foundation. The authors would additionally like to thank the anonymous referees of this paper for their insights.

REFERENCES

[1] K. Arkoudas and S. Bringsjord, 'Toward formalizing commonsense psychology: An analysis of the false-belief task', *PRICAI 2008: Trends in Artificial Intelligence*, 17–29, (2008).

[2] S. Bringsjord, 'The logicist manifesto: At long last let logic-based AI become a field unto itself', *Journal of Applied Logic*, **6**(4), 502–525, (2008).

[3] S. Bringsjord, D. Ferrucci, and P. Bello, 'Creativity, the Turing test, and the (better) Lovelace test', *Minds and Machines*, **11**, 3–27, (2001).

[4] S. Bringsjord and N. S. Govindarajulu, 'Toward a modern geography of minds, machines and math', *SAPERE*, **Philosophy and Theory of Artificial Intelligence**, (forthcoming).

[5] Nick Cassimatis, 'Cognitive substrate for human-level intelligence', *AI Magazine*, **27**(2), 71–82, (2006).

[6] M. Clark, *Cognitive Illusions and the Lying Machine*, Ph.D. dissertation, PhD thesis, Rensselaer Polytechnic Institute (RPI), 2008.

[7] R. Descartes, *The Philosophical Works of Descartes, Volume 1. Translated by Elizabeth S. Haldane and G.R.T. Ross*, Cambridge University Press, Cambridge, UK, 1911.

[8] D. Ellis and G. Poliner, 'Identifying cover songs' with chroma features and dynamic programming beat tracking', in *Proc. Int. Conf. on Acous., Speech, and Sig. Proc. ICASSP-07*, volume 4, pp. 1429–1432. IEEE, (April 2007). A MATLAB library is available at: http://labrosa.ee.columbia.edu/projects/coversongs/.

[9] B. Inhelder and J. Piaget, *The Growth of Logical Thinking from Childhood to Adolescence*, Basic Books, New York, NY, 1958.

[10] E.T. Mueller, *Commonsense reasoning*, Morgan Kaufmann, 2006.

[11] G.S. Pappas and M. Swain, *Essays on knowledge and justification*, Cornell University Press, 1978.

[12] A.D. Patel et al., 'Language, music, syntax and the brain', *Nature neuroscience*, **6**(7), 674–681, (2003).

[13] M.T. Pearce. Notes on CHARM - a specification for the representation of musical knowledge, 2002.

[14] L. S. Zettlemoyer and M. Collins, 'Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars', in *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, volume 5, pp. 658–666, (2005).

PoeTryMe: a versatile platform for poetry generation

Hugo Gonçalo Oliveira[1]

Abstract. PoeTryMe is a platform for the automatic generation of poetry. It has a versatile architecture that provides a high level of customisation. The user can define features that go from the poem configuration and the sentence templates, to the initial seed words and generation strategy. A prototype was implemented based on PoeTryMe to generate Portuguese poetry, using natural language processing resources for this language, and patterns that denote semantic relations in human-created poetry. The possible results are illustrated by three generated poems.

1 INTRODUCTION

Natural language generation [23] is a well-established sub-field of artificial intelligence and computational linguistics. Its main goal is to develop computer programs capable of producing text that is understood by humans. Biographies [15] and weather forecasts [2] are examples of the genres of text that have been generated automatically. Another example is the generation of text with creative features, including story narratives [3], jokes [24] or poetry (see section 2).

We have seen several attempts to generate creative artifacts automatically, with the help of computer programs, and we now accept the computer as an artist. The creation of visual art and the composition of musical pieces are other fields where creative systems have been developed for.

In this paper, we present PoeTryMe, a platform designed for the automatic generation of poetry. Given a generation grammar and a set of relational triples, PoeTryMe generates grammatically correct and meaningful sentences. It has a versatile architecture that provides a high level of customisation and can be used as the base of poetry generation systems, which can be built on the top of it. In PoeTryMe, everything can be changed: the base semantics, represented as relational triples; the templates of the generated sentences, included in the generation grammars; the generation strategies, that select the lines to include in the poem; and, of course, the poem configuration.

We start this paper by referring some work on the automatic generation of poetry, including two categorisations for this kind of systems. Then, we present an overview on the architecture of PoeTryMe, followed by the description of a prototype, implemented for the generation of Portuguese poetry. While introducing the external resources used, we describe the process for acquiring line templates, and the implemented generation strategies. Following, we illustrate the possible results of PoeTryMe by presenting three generated poems. Before concluding with some cues for future work, we categorise the implemented strategies.

[1] CISUC, University of Coimbra, Portugal, email: hroliv@dei.uc.pt, supported by FCT scholarship grant SFRH/BD/44955/2008, co-funded by FSE

2 RELATED WORK

The automatic generation of poetry is a complex task, as it involves several levels of language (e.g. phonetics, lexical choice, syntax and semantics) and usually demands a considerable amount of input knowledge. However, what makes this task more interesting is that some of the latter levels do not have to be strictly present.

On the one hand, writing poetic text does not have to be an extremely precise task [9], as several rules, typically present in the production of natural language, need to be broken [18]. For instance, there may not be a well-defined message. On the other hand, poetry involves a high occurrence of interdependent linguistic phenomena where rhythm, meter, rhyme and other features like alliteration and figurative language play an important role.

In this section, we present two categorisations of poetry generation systems, proposed in the literature. One of them considers the applied techniques and another the generated text.

2.1 Poetry generation techniques

Regarding the followed approaches and techniques used, poetry generation systems can be roughly grouped into four categories [8]: (i) template-based, which includes systems that just fill templates of poetry forms with words that suit syntactic and/or rhythmic constraints; (ii) generate-and-test; (iii) evolutionary; and (iv) case-based reasoning.

In generate-and-test systems, random word sequences are produced according to formal requirements, that may involve meter or other constraints. Manurung's chart system [17], WASP [9] and the generate-and-test strategy of Tra-la-Lyrics [12] are systems that fall into this category.

In Manurung's chart system, sentences are logically represented by first order predicates describing the input semantics, and charts are used to generate natural language strings that match a given stress pattern. While a chart parser analyses strings and translates them to logical forms, a chart generator translates logical forms to strings. During the generation, before adding the result of a new rule to the chart, its stress pattern is checked for compatibility with the target pattern. Only results with compatible patterns are added, ensuring that the generated text satisfies the pattern. WASP is a forward reasoning rule-based system that aims to study and test the importance of the initial vocabulary, word choice, verse pattern selection and construction heuristics, regarding the acceptance of the generated verses and complete poems. Tra-la-Lyrics [13, 12] is a system that aims to generate text based on the rhythm of a song melody, given as input. Using the sequence of strong and weak beats as a rhythmic pattern, the task of generating song lyrics is very similar to the generation of poetry. In the generate-and-test strategy, grammatical sentences are produced and then scored according to their suitability to a given meter/rhythmic pattern.

Evolutionary approaches rely on evolutionary computation techniques. POEVOLVE [16] and McGonnagall [18, 19] are examples of such approaches. POEVOLVE is a prototype that generates limericks, implemented according to a model that takes the real process of human poetry writing as a reference. In McGonnagall, the poem generation process is formulated as a state space search problem using stochastic hill-climbing search, where a state in the search space is a possible text with all its underlying representations, and a move can occur at any level of representation, from semantics to phonetics. The search model is an evolutionary algorithm encompassing evaluation and evolution.

As for case-based reasoning approaches, existing poems are retrieved, considering a target message, and then adapted to fit in the required content. Systems like ASPERA [10] and COLIBRI [5] fall into this category. They are forward reasoning rule-based systems that, given a prose description of the intended message and a rough specification of the type of poem, select the appropriate meter and stanza, generate a draft poem, request modification or validation by the user, and update their database with the information of the validated verse.

2.2 Generated poetry properties

Manurung [18] affirms that poetic text must hold all the three properties of meaningfulness, grammaticality and poeticness. More precisely, it must: (i) convey a conceptual message, which is meaningful under some interpretation; (ii) obey linguistic conventions prescribed by a given grammar and lexicon; and (iii) exhibit poetic features. An alternative categorisation for poetry generation attempts considers the latter properties and divide systems into the following: (i) word salad, which just concatenate random words together, without following grammatical rules, therefore not holding any of the properties; (ii) form-aware; and (iii) actual poetry generation systems.

In form-aware systems, the choice of words follows a pre-defined textual form, by following metrical rules. They thus hold the properties of grammaticality and poeticness. The WASP system [9], POEVOLVE [16], and the generative grammar strategy of Tra-la-Lyrics [13] fall into this category.

Actual poetry generation systems must hold the three properties. ASPERA [10] and COLIBRI [5] are examples of such systems. In both of them, words must be combined according to the syntax of the language and should make sense according to a prose message provided by the user. Also, when occurring at the end of lines, words may have additional constraints imposed by the strophic form. McGonnagall [18, 19] falls into this category as well, given that a goal state is a text that satisfies the three aforementioned properties. However, after several experimentations, Manurung et al. [19] state that it is difficult to produce both semantically coherent text in a strict agreement to a predefined meter.

There are other systems whose results exhibit poetic features, obey syntactic rules and, even though not following a well-defined and precise message, try to generate meaningful text, as they select sentences or words based on given seeds or semantic similarity. Examples of those include Wong and Chun's [25] haiku generator, Gaiku [20], and Ramakrishnan's lyrics generator [22, 21]. Wong and Chun generate haikus using a Vector Space Model (VSM), established by sentences in blogs. Candidate sentences are selected according to their semantic similarity. Gaiku generates haikus based on a lexical resource that contains similar words. Haikus are generated according to a selected theme and syntactic rules. Ramakrishnan el al. learned a model of syllable patterns from real melodies. The model was used in a system that, given a melody, generates meaningful sentences that match adequate syllabic patterns and rhyme requirements. Meaningful sentences were generated with the help of n-gram models, learnt from a text corpus. In a more recent version of the system [21], meaningful sentences are generated with the help of a knowledge base.

Furthermore, the random words strategy of Tra-la-Lyrics [13] falls in what can be considered as a fourth category, as the meter of the generated text suit the given rhythm and the text contains poetic features (rhyme), but the word order does not follow grammatical rules and there are no semantic constraints. In other words, this strategy holds the property of poeticness, but none of the others.

Recently, a novel system was presented for poetry generation [4] where, besides dealing with the three aforementioned properties, poems are generated regarding the mood for a certain day (good or bad), according to newspaper articles, and an aesthetic is produced, using a set of measures (appropriateness to the mood, flamboyance, lyricism and relevance for a selected article). The lines of the poem are collected not only from the articles, but also from short phrases mined from the Internet, and variations of the latter obtained by replacing some words with others semantically similar. Moreover, comments, supporting the choices made (e.g. mood, used sentences, aesthetic measures), are generated. While the latter contextualise the poem, the produced aesthetics may be used to evaluate the obtained results more objectively.

3 PoeTryMe

PoeTryMe is a poetry generation platform that relies on a modular architecture (see Figure 1) and thus enables the independent improvement of each module. This architecture intends to be versatile enough to provide a high level of customisation, depending on the needs of the system and ideas of the user. It is possible to define the semantics to be used, the sentence templates in the generation grammar, the generation strategy and the configuration of the poem. In this section, the modules, their inputs, and interactions are presented.

3.1 Generation Strategies

A Generation Strategy implements a method that takes advantage of the Sentence Generator to obtain lines and build up a poem. The poem is generated according to a set of seed words, used to get sentences from the Sentence Generator, and a poem template. The latter contains the poem's structure, including the number of stanzas, the number of lines per stanza and the number of syllables of each line. Figure 2 shows the representation of poem structure templates, for generating a sonnet and for a haiku. In the latter, the Portuguese word *estrofe* indicate a stanza and the *verso* indicates a line.

An instantiation of the Generation Strategy does not generate sentences. It just includes one or several heuristics to find the better sentences for each line, obtained from the Sentence Generator. Heuristics might consider features like meter, rhyme, coherence between lines or other, depending on the poem's purpose. In our prototype (see section 4), we have implemented a basic strategy, a generate-and-test strategy, and an evolutionary approach.

3.2 Sentence Generator

The Sentence Generator is the core module of PoeTryMe's architecture and is used to generate meaningful sentences with the help of:

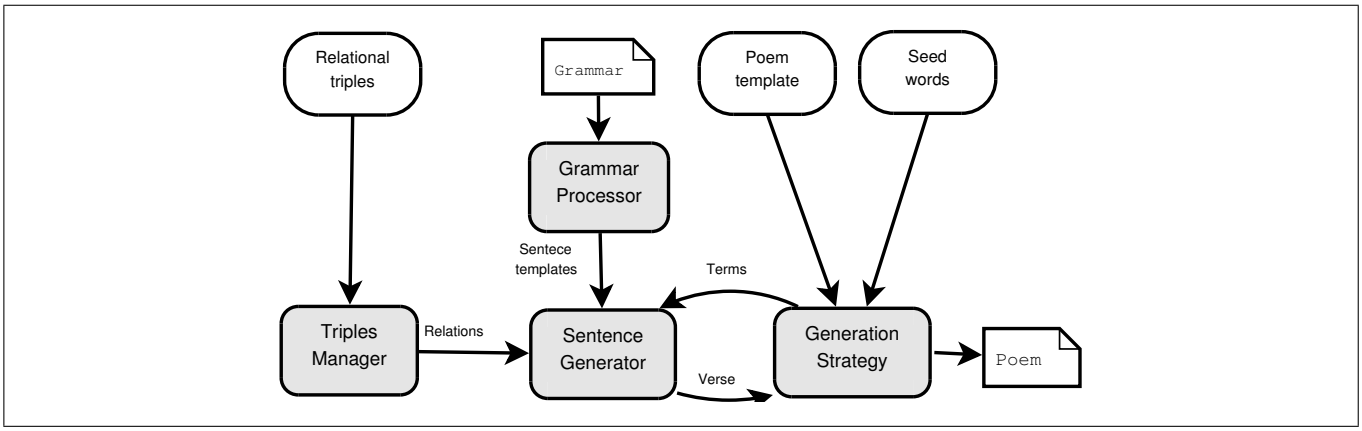


Figure 1. PoeTryMe architecture

```
#sonnet
estrofe{verso(10);verso(10);verso(10);verso(10)}
estrofe{verso(10);verso(10);verso(10);verso(10)}
estrofe{verso(10);verso(10);verso(10)}
estrofe{verso(10);verso(10);verso(10)}

---

#haiku
estrofe{verso(5);verso(7);verso(5)}
```

Figure 2. First, the structure of a sonnet, and then, the structure of a haiku.

- a semantic graph, managed by the Triples Manager, where the nodes are words and the edges are labelled according to a relation type. A tuple $t = (node_1, relation\_type, node_2)$ establishes a relational triple;
- generation grammars, processed by the Grammar Processor, which contain textual templates for the (chart) generation of grammatical sentences denoting a semantic relation.

The generation of a sentence starts with a set of seed words, used to select a subgraph from the main semantic graph. The former contains only relations involving the seed words, or connected indirectly to them from a path no longer than a predefined depth δ.

Generation proceeds by selecting a random triple in the subgraph and a random grammar rule matching its relation type. There must be a direct mapping between the relation names, in the graph, and the name of the head rules, in the grammar. After inserting the arguments of the triple in the rule body, the resulting sentence is returned.

Similarly to Manurung [17], the Grammar Processor uses a chart-parser in the opposite direction, in order to perform chart generation. The body of the rules should consist of natural language renderings of semantic relations. Besides the simple terminal tokens, that will be present in the poem without any change, the Grammar Processor supports special terminal tokens that indicate the position of the relation arguments (`<arg1>` and `<arg2>`), to be filled by the Sentence Generator.

4 POETRY GENERATION IN PORTUGUESE

This section is about the prototype implemented in the top of PoeTryMe, to generate Portuguese poetry. We present the natural language processing resources used in the prototype, list some renderings for semantic relations, included in the grammars after exploiting human-created poetry, describe the implemented generation strategies, and show three examples of generated poems.

4.1 Resources used

PEN[2] is an implementation of the Earley [6] chart-parsing algorithm that analyses sentences according to grammars given as input. These grammars are editable text files, where each line contains the name of a rule and its body. In order to differentiate rule tokens from terminals, rule names are upper case. An example of a simple and valid rule set is shown in Figure 3, where the Portuguese word `RAIZ`, meaning root, is the starting point of the grammar. We used PEN in the opposite direction, in order to perform chart generation.

```
RAIZ ::= RULE
RAIZ ::= RULE <&> OTHERRULE

RULE ::= terminal
OTHERRULE ::= otherterminal
OTHERRULE ::= otherterminal <&> OTHERRULE
```

Figure 3. PEN example rules.

CARTÃO [11] is a public lexical knowledge base for Portuguese, extracted automatically from three Portuguese dictionaries. It contains about 325,000 semantic triples, held between words, which can be used as a semantic graph. A semantic triple, represented as follows, indicates that one sense of the word in the first argument (`arg1`) is related to one sense of the word in the second (`arg2`) by means of a relation identified by `RELATION_NAME`:

[2] Available from http://code.google.com/p/pen/

`arg1 RELATION_NAME arg2`
e.g. `animal HIPERONIMO_DE cão`
(*animal* HYPERNYM_OF *dog*)

CARTÃO includes relations as synonymy, hypernymy, part-of, causation, purpose and property, amongst others. The name of the semantic relation also defines the part-of-speech of its arguments.

SilabasPT[3] is an API that performs syllabic division and stress identification for Portuguese words. It was developed to help generating text based on rhythm in the project Tra-la-Lyrics [13, 12], but it is an independent API that can be integrated in other applications.

LABEL-LEX[4] is a lexicon of Portuguese, with 1,5 million inflected word forms, automatically generated from about 120,000 lemmas. For each word form, it provides information such as the lemma, the part-of-speech and other morphological information.

4.2 Relations and renderings

Instead of creating our own grammars manually, we automatised this task by exploiting real Portuguese poetry. It is a well known fact that semantic relations can be expressed in running text by discriminating patterns, typically used to discover new relations (see, for instance, [14]). Therefore, in order to discover patterns for our grammar, we extracted all sentences in a collection of Portuguese poetry[5], where the arguments of, at least, one triple of CARTÃO co-occurred.

After replacing the arguments by terminal tokens, relative to the first and the second argument (`<arg1>` and `<arg2>`), we added the sentence as a rule in the grammar with the name of the relation. Table 1 shows examples of the relations used, example arguments, and automatically discovered patterns, used as renderings for the relations. About 700 patterns were discovered.

In order to deal with inflected words and to keep number and gender agreement in the generated sentences, before discovering the patterns, we added the number and the gender of the noun and adjective arguments to the relation name. For instance, the triple (*destino* synonym-of *futuro*) was changed to (*destino* ms-synonym-of-ms *futuro*), while the triple (*versos* part-of *quadras*) was changed to (*versos* mp-part-of-fp *quadras*). However, for the sake of clarity, we did not include this information in table 1. The number and gender information was obtained from LABEL-LEX.

4.3 Implemented generation strategies

Three different generation strategies were implemented in the prototype. While one is just used as a baseline for debugging, the others follow evolutionary approaches, as Manurung's [18] algorithm for poetry generation.

In both of the latter strategies, there is an evaluation function that scores each sentence according to the absolute difference between the number of syllables the poem line has in the template, with the number of syllables in the generated sentence – the lower the evaluation, the better the sentence is. SilabasPT is used to count the number of syllables of each sentence and identify its last stress. The final score of a poem, used only in the third strategy, is the sum of the scores of all lines plus a bonus for poems with lines in the same stanza with the same termination (rhyme). The other strategies do not score rhymes because they do not generate the poem as a whole, but just gather lines independently.

[3] Available from http://code.google.com/p/silabaspt/
[4] Available from http://label.ist.utl.pt/pt/labellex_pt.php
[5] We used `wget` to collect all the poems in the portal *Versos de Segunda*, available from http://users.isr.ist.utl.pt/~fb/VdS/

The algorithms involved in each one of the strategies are briefly described as follows:

- Basic: for each line to be filled, a random sentence is generated using the key terms;
- Generate-and-test: for each line to be filled, n random sentences are generated. The one with best score is chosen. All unused sentences are indexed and can be used if a new line needs exactly the same amount of syllables of the previously unused sentence.
- Evolutionary: an initial population of n poems is generated using the basic strategy. Then, each poem is scored according to the aforementioned evaluation function. Each new generation consists of the poems with the best evaluation, poems that are the result of crossing two random poems in the population, and newly created poems as well. When two poems are crossed, a new poem is created with lines selected from both. The best scoring poem of the last generation is returned.

4.4 Illustrative results

For illustration purposes, we present three poems obtained with the implemented prototype. In figure 4, we present a haiku, obtained with the generate-and-test strategy, using 100 generations per line, and the seed words *arte* and *paixão* (in English, art and passion), with $\delta = 1$. With more depth, the system has more word choices and thus more variations, but it is less focused on the seed words. On the other hand, using $\delta = 1$, each line will include one seed word, which is the case for the presented haiku.

The example follows the 5-7-5 syllable pattern correctly. However, the choice of words for the haiku must be done carefully, because long words prevent the generation of short lines.

```
ah paixão afecto
não tem paixão nem objecto
sem na arte modos
```

Figure 4. Example of a generated haiku.

In figure 5, we present a sonnet, this time obtained with the evolutionary approach, after 25 generations of 100 poems. In each generation, the population consisted of 40% of the best poems from the previous, 40% resulting from crossing, and 20% new. The probability of crossing, which consists of swapping two lines of two different poems, was set to 50%, and the bonus for rhymes in the end of lines to -3. Once again, we used $\delta = 1$. However, as a sonnet has fourteen lines, in order to have more variations, we used more seed words, namely: *computador*, *máquina*, *poeta*, *poesia*, *arte*, *criatividade*, *inteligência*, *artificial* (computer, machine, poet, poetry, art, creativity, intelligence, artificial).

The meter of the poem is very close to ten syllables per line. Only the third and seventh line have one additional syllable. Also, all the verses include one of the seeds and a related word. Meaning is present in each isolated verse and thus, a meaning emerges for the whole poem. However, there are no rhymes, which suggests that the bonus is not enough to generate poems with this feature.

Even so, in an attempt to force the poems to have rhymes, we generated more poems with the evolutionary approach, with similar

Type	POS	Example args.	Example rule
Synonym-of	noun,noun	*destino,futuro* *(destiny,future)*	não sei que <arg1> ou <arg2> compete á minha angústia sem leme
	adj,adj	*quebrada,rota* *(broken,ragged)*	<arg1> a espada já <arg2> a armadura
Antonym-of	adj,adj	*possível,impossível* *(possible,impossible)*	tudo é <arg1>, só eu <arg2>
Hypernym-of	noun,noun	*mágoa, dor* *(sorrow,heartache)*	e a própria <arg2> melhor fora <arg1>
Part-of	noun,noun	*versos,quadras* *(lines,blocks)*	as minhas <arg2> têm três <arg1>
Causation-of	noun,noun	morte,luto *(death,grief)*	a seca, o sol, o sal, o mar, a morna, a <arg1>, a luta, o <arg2>
	verb,noun	*dor,doer* *(pain,to_hurt)*	é <arg2> que desatina sem <arg2>
Purpose-of	noun,noun	*arma,munição* *(weapon,ammunition)*	com <arg2> sem <arg1>
	verb,noun	*taça,beber* *(cup,to_drink)*	<arg1> para <arg2> junto á perturbada intimidade
Has-quality	noun,adj	*certeza,certo* *(certainty,sure)*	eu que não tenho nenhuma <arg1> sou mais <arg2> ou menos <arg2>
Property-of	adj,noun	*letal,morte* *(letal,death)*	a <arg2> é branda e <arg1>

Table 1. Automatically discovered renderings, included in the grammars.

```
e não há deus nem preceito nem arte
um palácio de arte e plástica
as máquinas pasmadas de aparelhos
num mundo de poesias e versos

o seu macaco era duas máquinas
horaciano antes dos poetas
para as consolas dos computadores
num mundo de poesias e carmes

longas artes apografias cheias
tenho poesias como a harpa
poema em arte modelação

somos artificiais teatrais
máquinas engenhocas repetido
um poeta de líricas doiradas
```

Figure 5. Example of a generated sonnet.

settings as the previous, except for: (i) the bonus for rhymes, which was set to -10; (ii) δ was set to 2; (iii) regarding the higher value of δ, the provided seed words were only two, more precisely, they were the same as in the first presented poem (*arte* and *paixão*).

One of the resulting poems is a block of four lines, presented in figure 6. All the lines of this poem end with the same termination, but none of them agrees with the correct metrics. Actually, all the lines have more syllables than they should – one in the first and third lines, six in the second and four in the fourth. Regarding the semantics of the poem, it is less focused on the seeds, as expected, and none of them is actually used. Still, the poem contains words related to art, as *escultura*, *representação* and *composição* (sculpture, acting, composition).

As others have noticed for meaningfulness and poeticness [19], we confirmed that it is difficult to generate a poem that strictly obeys to the three properties of poetry generation without relaxing on, at least,

```
ah escultura representação
e os que dão ao diabo o movimento da convulsão
sua composição de preparação
é destino estar preso por orientação
```

Figure 6. Example of a generated block of four lines.

one of them. Moreover, the performed experiments showed that the generate-and-test strategy, with 100 or more generations of each line, result more consistently in poems with better evaluation. However, as it is, the latter strategy does not have bonus for rhymes, and they will only occur by chance, as in the poem of figure 4. On the other hand, the evolutionary approach is more complex and has parameters that should be deeper analysed, but can generate poems with rhymes in a trade-off for less precise meter.

5 CATEGORISATION

Regarding that we have implemented different strategies for generating poetry, the work presented here falls in more than one category. Although our approach uses sentence templates, only one is actually template-based (basic strategy), while the other two follow, respectively, a generate-and-test and an evolutionary approach.

As for their goals, since we use a semantic graph as input and we render information in it to natural language sentences, we can say that, if the graph is well constructed, and regarding that the grammars generate grammatically correct sentences, our system holds both the property of grammaticality and meaninfulness. Nevertheless, the latter property can be seen as "weaker" than the others, because the user only provides seed terms, and not a fixed and well-formed meaning. As for poeticness, our system supports different configurations of poems and two of the implemented strategies take the number of syllables per line into consideration. Furthermore, the evolutionary

approach has a bonus for rhymes. Therefore, according to Manurung [18], when following the generate-and-test or the evolutionary approach, our prototype can be seen as an actual poetry generation system.

6 CONCLUSIONS AND FURTHER WORK

We have presented PoeTryMe, a platform for the automatic generation of poetry, and a prototype, implemented in the top of this platform. One of the strengths of PoeTryMe is its high level of customisation. It may be used with different lexical resources and generate different poem configurations. The generation grammars may be edited and improved at will, in order to cover new linguistic constructions. Furthermore, new generation strategies may be implemented, which can originate different and interesting types of poems, according to a predefined purpose. Therefore, PoeTryMe can be used as the starting point for one (or more) poetry generation systems, eventually after taking future directions for improvement.

For instance, more generation strategies can be developed and the evolutionary strategy can be improved after testing more complex evaluation functions. Besides the number of syllables, other aspects, such as the the stress patterns, may also be considered. A strategy for generating rhymes more consistently, without a negative impact on the meter, should as well be devised.

In the implemented prototype, the lexical knowledge base used is structured on words. On the one hand, this might be a limitation, because natural language is ambiguous and several words have more than one sense. On the other hand, poetry is often vague and does not have to convey a precise message. Nevertheless, it would be interesting to compare the results of using word-based lexical resources against sense-aware resources (e.g. WordNet [7]) Also interesting would be to use a polarity lexicon (e.g. SentiWordNet [1]), in order to generate poetry with a predefined sentimental orientation (e.g. positive or negative), as others [4] have recently accomplished.

Although our prototype was created for Portuguese, the platform's architecture could be used for generating poetry in other languages. In order to do that, we would need to use external resources for the target language, including the lexical knowledge base, the syllabic division algorithm, and the morphology lexicon.

Finally, we should add that, as it happens for other creative artifacts, it is difficult to objectively evaluate the quality of a poem. Still, in the future, our results should be the target of some kind of validation and evaluation. Ideas for validation include comparing the configuration of the generated poems with similarly structured human-created poems, while evaluation might be performed based on the opinion of human subjects, which should consider aspects like the structure, meter, novelty and semantics of generated poems.

REFERENCES

[1] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani, 'SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining', in *Proceedings of the 7th International Conference on Language Resources and Evaluation*, LREC 2010, pp. 2200–2204, Valletta, Malta, (2010). ELRA.

[2] Anja Belz, 'Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models', *Natural Language Engineering*, **14**(4), 431–455, (October 2008).

[3] Selmer Bringsjord and David A. Ferrucci, *Artificial Intelligence and Literary Creativity: Inside the Mind of BRUTUS, a Storytelling Machine*, Lawrence Erlbaum Associates, Hillsdale, NJ., 1999.

[4] Simon Colton, Jacob Goodwin, and Tony Veale, 'Full FACE poetry generation', in *Proceedings of 3rd International Conference on Computational Creativity*, ICCC 2012, pp. 95–102, Dublin, Ireland, (2012).

[5] Belén Díaz-Agudo, Pablo Gervás, and Pedro A. González-Calero, 'Poetry generation in colibri', in *Proceedings of 6th European Conference on Advances in Case-Based Reasoning (ECCBR 2002)*, pp. 73–102, London, UK, (2002). Springer.

[6] Jay Earley, 'An efficient context-free parsing algorithm', *Communications of the ACM*, **6**(8), 451–455, (1970). Reprinted in Grosz et al. (1986).

[7] *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*, ed., Christiane Fellbaum, The MIT Press, May 1998.

[8] P. Gervás, 'Exploring quantitative evaluations of the creativity of automatic poets', in *Workshop on Creative Systems, Approaches to Creativity in Artificial Intelligence and Cognitive Science, 15th European Conference on Artificial Intelligence*, (2002).

[9] Pablo Gervás, 'WASP: Evaluation of different strategies for the automatic generation of spanish verse', in *Proceedings of AISB'00 Symposium on Creative & Cultural Aspects and Applications of AI & Cognitive Science*, pp. 93–100, Birmingham, UK, (2000).

[10] Pablo Gervás, 'An expert system for the composition of formal spanish poetry', *Journal of Knowledge-Based Systems*, **14**, 200–1, (2001).

[11] Hugo Gonçalo Oliveira, Leticia Antón Pérez, Hernani Costa, and Paulo Gomes, 'Uma rede léxico-semântica de grandes dimensões para o português, extraída a partir de dicionários electrónicos', *Linguamática*, **3**(2), 23–38, (December 2011).

[12] Hugo Gonçalo Oliveira, F. Amílcar Cardoso, and Francisco C. Pereira, 'Exploring different strategies for the automatic generation of song lyrics with tra-la-lyrics', in *Proceedings of 13th Portuguese Conference on Artificial Intelligence*, EPIA 2007, pp. 57–68, Guimarães, Portugal, (2007). APPIA.

[13] Hugo Gonçalo Oliveira, F. Amílcar Cardoso, and Francisco Câmara Pereira, 'Tra-la-lyrics: an approach to generate text based on rhythm', in *Proceedings of 4th International Joint Workshop on Computational Creativity*, pp. 47–55, London, UK, (2007). IJWCC 2007.

[14] Marti A. Hearst, 'Automatic acquisition of hyponyms from large text corpora', in *Proceedings of 14th Conference on Computational Linguistics*, COLING'92, pp. 539–545. ACL Press, (1992).

[15] Sanghee Kim, Harith Alani, Wendy Hall, Paul H. Lewis, David E. Millard, Nigel R. Shadbolt, and Mark J. Weal, 'Artequakt: Generating tailored biographies with automatically annotated fragments from the web', in *Proceedings of ECAI 2002 Workshop Semantic Authoring, Annotation and Knowledge Markup*, SAAKM 2002, pp. 1–6, (2002).

[16] R. P. Levy, 'A computational model of poetic creativity with neural network as measure of adaptive fitness', in *Proceedings of the ICCBR-01 Workshop on Creative Systems*, (2001).

[17] Hisar Manurung, 'A chart generator for rhythm patterned text', in *Proceedings of 1st International Workshop on Literature in Cognition and Computer*, (1999).

[18] Hisar Manurung, *An evolutionary algorithm approach to poetry generation*, Ph.D. dissertation, University of Edinburgh, 2004.

[19] Ruli Manurung, Graeme Ritchie, and Henry Thompson, 'Using genetic algorithms to create meaningful poetic text', *Journal of Experimental & Theoretical Artificial Intelligence*, **24**(1), 43–64, (2012).

[20] Yael Netzer, David Gabay, Yoav Goldberg, and Michael Elhadad, 'Gaiku: generating haiku with word associations norms', in *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, CALC '09, pp. 32–39. ACL Press, (2009).

[21] Ananth Ramakrishnan A and Sobha Lalitha Devi, 'An alternate approach towards meaningful lyric generation in tamil', in *Proceedings of the NAACL HLT 2010 Second Workshop on Computational Approaches to Linguistic Creativity*, CALC '10, pp. 31–39. ACL Press, (2010).

[22] Ananth Ramakrishnan A, Sankar Kuppan, and Sobha Lalitha Devi, 'Automatic generation of Tamil lyrics for melodies', in *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, CALC '09, pp. 40–46, Stroudsburg, PA, USA, (2009). ACL Press.

[23] Ehud Reiter and Robert Dale, *Building natural language generation systems*, Cambridge University Press, New York, NY, USA, 2000.

[24] Graeme Ritchie, Ruli Manurung, Helen Pain, Annalu Waller, Rolf Black, and Dave O'Mara, 'A practical application of computational humour', in *Proceedings of 4th International Joint Workshop on Computational Creativity*, pp. 91–98, London, UK, (2007).

[25] Martin Tsan Wong and Andy Hon Wai Chun, 'Automatic haiku generation using VSM', in *Proceeding of 7th WSEAS Int. Conf. on Applied Computer & Applied Computational Science (ACACOS '08), Hangzhou, China*, (2008).

On the Feasibility of Concept Blending in Interpreting Novel Noun Compounds

Ahmed M. H. Abdel-Fattah[1]

Abstract. This article discusses specific aspects of combining knowledge domains in a concept-based model of computational creativity. It focusses in particular on the problem of combining concepts that represent word nouns when modifier-head compounds are created. The article suggests, on a conceptual level, a method that interprets some novel noun compounds by a computationally-plausible cognitive mechanism, namely the concept blending. In order to suggest a conceptual relationship possibility between the modifier and the head nouns in such compounds, the given method utilizes an analogical relation, which is used to conceptually blend the domain representations of the constituent nouns.

1 INTRODUCTION AND MOTIVATION

It has long been interesting how human beings interpret a novel compound of known words. By a novel compound we refer to a composition that consists of two or more known words. The meanings of the words are known, but that of the novel compound itself may have never been encountered before. Generally intelligent cognitive agents like humans possess the ability to understand such meanings. Even if a compound has never been encountered before, a human can creatively suggest a sane interpretation, which might distantly differ from the meanings of the words it comprises, yet makes sense. Both the importance and the extreme difficulty of the problem of interpreting novel noun compounds are well-appreciated, as implied by the huge literature dedicated to solving and using it [3, 12, 13, 16, 22, 33, to mention just a few]. So far it is hard to find a study that presents an adequate, cognitively-based account of the problem that takes a computational creativity perspective towards solving it. Therefore, the problem still deserves contributions by looking into new solution directions. In the context of computational models of creativity and general intelligence, this eventually helps in affording cognitive agents the ability of interpreting or learning possible meanings of newly-formed combinations of known words.

1.1 A Concept-Based Model of Creativity

Our ultimate goal is to develop a cognitively-inspired, concept-based computational model of general intelligence that is based on cross-domain reasoning and accumulation of past experiences. To feasibly simulate forms of creativity, the model employes cognitive processes, such as analogy-making and concept blending, as well as ideas from nature-inspired intelligence processes, such as the ant-colony optimization (ACO) techniques [6]. Agents of this model are intended to build and manipulate their knowledge base (KB) using a knowledge representation (KR) framework that categorizes beliefs as belonging to knowledge domains (schemas, concepts, or theories). The domains, in turn, are represented in a formal language (e.g. first-order logic). In the following we discuss the intuition and the principles that stimulate such a model.

The KB of the cognitive agents can be built from organizing beliefs into knowledge concepts. The beliefs result basically from perception, and the experiences the agents acquire direct the organization process. However, not only perception is what determines the beliefs, since neither perception is necessarily an accurate interpretation of the world, nor can the agents possibly assimilate all the knowledge that results from what they perceive. When they need to make rapid, but coherent decisions, some experience-based 'mental shortcuts' enable the agents to categorize the learned knowledge by building schemas (or mental spaces), and the organization of the beliefs into knowledge concepts comes about. As a result, this affects the creation of another type of (internally organized) beliefs that do not result directly, or only, from perception, but rather from the interplay between the already available knowledge and experience. Useful beliefs of either types will keep being reinforced, establishing links and ties to other knowledge where they are of use, whereas knowledge that is not always in use will typically be less remembered. As knowledge undergoes an internal screening, depending on the history and experience of the agents, the agents may 'forget' some of the large amounts of beliefs they have acquired over time. They still can form new concepts to compensate knowledge shortage, by means of combining seemingly related or analogical concepts to create new ones.

In this article, we propose a way in which the cognitive agents in our model can combine two of the existing concepts, in order to create a third one that depends on what, and how, beliefs are organized in each of the former two. The study of a model of this kind, though difficult, is important from both a theoretical and a practical points of view, and its applications are abound. It clearly raises at least as many challenging and interesting questions as the number of the aspects that can be considered in the study. For example, the formal descriptions call several ideas from artificial intelligence and cognitive science, such as knowledge representation, belief change, and concept learning and formation. Moreover, there is no general consensus among cognitive psychologists and philosophers as to what concepts are, how they develop, or how they are represented. Many theories of concepts, whence, may need to be exposed, be they prototype theory-, theory view-, schema-, or exemplar-related (see cf. [25, 27, 29, 39] for an overview). In addition to its inherent difficulty, the latter issue is even connected with the expressiveness of the selected formal language. Limitations of various sorts prevent a complete investigation of the model in this article, but the needed principles for the current discussion are quickly addressed below.

[1] Institute of Cognitive Science, University of Osnabrück, Germany.

1.2 Model Assumptions and Basic Principles

In the model, the knowledge base is denoted by $\mathbb{K}_B$, which also stores experiences as beliefs. The beliefs, $b \in \mathbb{K}_B$, are represented by propositions using the formalism of the underlying KR framework. The model allows agents not only to store past experiences as a type of belief but to assign numeric values to such experiences as well. These values are referred to by *entrenchment values*. They serve as mnemonics of belief occurrences and rank them, somehow, according to importance and frequency. Entrenchment values depend on how recently, and how many times, have the beliefs been retrieved by the agent from $\mathbb{K}_B$ (e.g. in a new concept formation process). The assignment of an entrenchment value to each belief in the agent's KB contributes, in turn, to a total *entrenchment level* of the knowledge concepts that are linked with this particular belief (e.g. in their representations). The (overloaded) function $e_xV : \mathbb{K}_B \cup \mathbb{K}_C \rightarrow [0,1]$ is used to reflect both the entrenchment value, $0 \leq e_xV(b) \leq 1$, of a belief $b \in \mathbb{K}_B$ and the entrenchment level, $0 \leq e_xV(c) \leq 1$, of a concept $c \in \mathbb{K}_C$, where $\mathbb{K}_C$ is the knowledge base of concepts. A concept $c \in \mathbb{K}_C$ is called a HELCO if $e_xV(c) \geq \eta$ and is called a LEVCO otherwise, where $0 < \eta < 1$ is a threshold value. The concepts can either be 'innate' (i.e. built-in), with entrenchment level $e_xV(c) = 1$, or be formed as a result of a concept formation or a categorization of beliefs. In the latter case $e_xV(c) < \eta$.

The knowledge base of concepts, $\mathbb{K}_C$, functions as the *lexicon* that contains the representations of the words. Each known word is therefore represented by a concept that has an associated representation of the agent's beliefs and past experiences that are linked to that concept. The concepts that are already formed can be thought of as denoting already-known words, whereas the concepts that will be formed interpret the novel compounds.

The process of interpreting a novel compound by means of already-known nouns is equivalent, in a sense, to 'a process that creates a new concept with a low entrenchment level (i.e. a LEVCO) by conceptually blending already-existing concepts with high entrenchment levels (i.e. HELCOs)'. In other words, when HELCOs combine, a LEVCO results with an entrenchment level that depends on the entrenchment levels of the composing HELCOs. For a newly combined LEVCO, $B \in \mathbb{K}_C$, its entrenchment level $e_xV(B)$ is a function in $e_xV(S)$ and $e_xV(T)$ of the composing HELCOs, $S, T \in \mathbb{K}_C$.[2] In this way, based on the agent's background knowledge of the composing words, the model is assumed to endow its agents with the ability to construct possible meanings of newly composed sentences (i.e. word combinations). The composition of the constituent concepts is added to $\mathbb{K}_C$ as a new concept.

We think our idea to give concepts (as well as beliefs) entrenchment values makes a perfect sense to be considered in a knowledge based model of computational creativity, in particular when beliefs and experiences are what control the creation of meanings. As given by Peter Gäredenfors in [14], forced belief revisions may not give up some particular beliefs because they have a high *epistemic entrenchment*. In fact, Gäredenfors suggests that not all sentences in a belief set are of equal value for planning or problem-solving purposes. He proposes a formal tool, a binary relation, to control the contraction of beliefs by means of an ordering of their importance [15]. Moreover, in his discussion about the formal representation of epistemic states in a dynamic theory of such states, the philosopher Wolfgang Spohn sees one presentation of beliefs as more finely graded elements that come in numerical degrees [35]. Hansson also points out this exact fact in his discussion about giving up beliefs (cf. [23, Chapter 2]).

[2] No details will be given here about how these values are computed.

2 BACKGROUND AND LITERATURE

2.1 Conceptual Blending

Conceptual blending (CB) has been proposed as a powerful mechanism that facilitates the creation of new concepts by a constrained integration[3] of available knowledge. CB operates by mixing two input knowledge domains, called the "mental spaces", to form a new one that basically depends on the mapping identifications between the input domains. The new domain is called the blend, which maintains partial structures from both input domains and presumably adds an emergent structure of its own.

Three (not necessarily ordered) steps usually take place in order to generate a blend. The first is the composition (or fusion) step, which pairs selective constituents from the input spaces into the blend. In the second step, the completion (or emergence), a pattern in the blend is filled when structure projection matches long-term memory information. The actual functioning of the blend comes in the third step, the elaboration step, in which a performance of cognitive work within the blend is simulated according to its logic (cf. [8, 31]).

Figure 1 illustrates the four-space model of CB, in which two concepts, SOURCE and TARGET, represent two input spaces (the mental spaces). Common parts of the input spaces are matched by identification, where the matched parts may be seen as constituting a GENERIC space. The BLEND space has an emergent structure that arises from the blending process and consists of some matched and possibly some of the unmatched parts of the input spaces (cf. Figure 1). One of the famous blending examples is Goguen's HOUSE-BOAT and BOATHOUSE blends, which result, among others, from blending the two input spaces representing the words HOUSE and BOAT (cf. [18]).

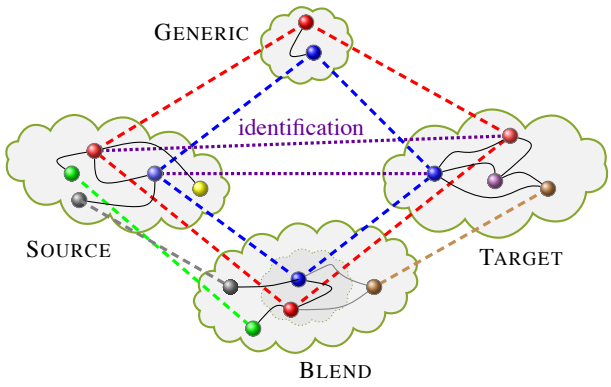


Figure 1. The four-space model of CB: common parts of the SOURCE and TARGET concepts are identified, defining a GENERIC space and a BLEND. The connecting curves within a concept reflect an internal structure.

As an important part of cognition, CB proved its importance in expressing and explaining cognitive phenomena, such as metaphor-making, counterfactual reasoning, as well as its usefulness in analogical reasoning and creating new theories [5, 8, 9, 19]. Nevertheless, there is no general computational account of blending, as a framework model, that has been proven powerful enough to cover all the examples in the literature. Combining meanings of word concepts is proposed here as a new application direction of using the ideas of CB in concept creation. We do not claim however that this is precisely

[3] Whence, CB is sometimes referred to as 'conceptual integration'.

how concepts are created in the real cognitive mind, neither do we claim that this always gives only the meaningful outputs.

According to the above discussion, we believe that the formalization of the aspects of CB is expected to produce a significant development in artificial intelligence (AI) in general and computational creativity in particular. Only few accounts have been given to formalize CB or its principles, yet they are not broad enough to suit generic computational accounts of CB (cf. [2, 18, 31, 36]). Nonetheless, CB itself still suffers from the lack of formality across its many aspects. The well-known optimality principles of CB, for instance, raise a challenge for developing such formalizations: these principles are the guideline pressures that are assumed to derive the generation of a feasible blend and distinguish good blends from bad ones [8, 30].

2.2 Conceptual Compounds

There is a general interest by cognitive scientists in analyzing how noun-noun (e.g. BOOK BOX) and adjective-noun (e.g. RED NOSE) combinations are interpreted by humans. Whether expressed in exact terms or in a metaphorical sense, several models are proposed to show how such interpretations could be performed (cf. [5, 7, 24, 37, 38] for instance). In some human languages, such as German and English, the construction of combinations involves known words, but the combination itself can range from the idiomatic or very well-known (e.g. TYPEWRITER, RAILWAY and SNOWFLAKE) to the unprecedented (e.g. CACTUS FINGER). Idiomatic combinations can also be referred to as *lexical compounds*.

Cognitive psychologists use the term conceptual combination, whereas linguists refer to word combinations as compounds or compound nominals. The term *conceptual combination* (CC) refers to the general ability that humans have of constructing a meaningful novel concept as a combination of input concepts, based on the knowledge of the meanings of the individual concepts that compose such a combination. A *compound nominal* (CN) as well refers to the resulting compound that acts as a noun itself and comprises two or more words[4], such as HIGH ENTRENCHMENT LEVEL CONCEPT.

Since in our model words are represented as concepts on a language-independent level, both terms can be used interchangeably. In any case, the process of juxtaposing two nouns is seen in our model as a creative production process not as a compositionality process, though both may be related. We count CB as a general method of elegantly mixing any two concepts. The composition denotes a newly established single *conceptual compound* that usually has a different interpretation than that of the (two) composing ones. This is why we claim that CB can feasibly be used in interpreting novel compounds: the interpretation of novel noun-noun compounds is achieved by a language-independent method that creates novel compounds by conceptually blending the corresponding concepts.

The specific problem type we are addressing here is that of interpreting unprecedented modifier-head, noun-noun compounds, i.e. previously unseen compounds that comprise exactly two already known nouns: the modifier followed by the head (e.g. COGNITIVE SCIENCE). The connection that is being made here is to issues in general intelligence, computational creativity and concept invention, but he nature of the problem of constructing the interpretation of word compounds has applications in several domains (e.g. in natural language processing (NLP) and information retrieval (IR) cf. [16]).

[4] Such words are nouns in most of the cases, but they need not be. E.g. 'get the ball rolling' can be interpreted as "INITIALIZATION".

2.3 Conceptual Challenges

In most of the cases, the meaning of a novel compound may not at all be simple to interpret by humans (not to mention to compute) because it highly depends on many factors, such as the corresponding meanings of the composing words (that do not always have unique semantic mappings), the particular uses of such meanings, the surrounding context, and an implicit relationship between the composing words. The latter *conceptual relationship* between the two composing words is considered one of the main challenges in interpreting novel compounds. The conceptual relationships that may implicitly exist between a modifier and a head in a compound are very difficult to be abstracted. As a quick example, compare what the modifier "WOUND" contributes to in "HAND WOUND", to what it contributes to in "GUN WOUND" (cf. [5, 22]).

A compound does not simply equal the sum of its parts, and its meaning is as sensitive to changes as its underlying concepts, which can themselves change over time or by knowledge revision[5]. Even a static context can highly affect the meaning of a noun-noun compound, by telling a specific anecdote from which the meaning can be inferred (e.g. COMPUTER SMILE). Also, the background knowledge and the previous experiences of one person influence the comprehension or meaning construction (e.g. a DECOMPOSING COMPOUND to a chemist may differ from that to a linguist [12]). In addition to acknowledging previous work, we quickly mention some proposals related to the deeper analyses in the literature. This helps us in further clarifying why the problem is of an inherently baffling nature and that no agreement between researchers about a ubiquitous solution has been reached.

In fact, many linguists do have the consensus that comprehension requires the presence of *relational inferences* between the concepts in a compound. For example, there are nine *recoverably deletable* predicates, given in [26], which characterize the semantic relationships between the composing nouns in a compound (see also [5]). The *abstract relations theory* also indicates a limited number of predicates to relate a modifier noun with a head noun in a modifier-head compound [13]. The *dual process model* claims that attributive and relational combination are two distinct processes resulting from comparison and integration, respectively [37], but other linguistic models raise the possibility that a single-process integration model could account for all combinations [7, 12]. A tremendous number of other works could also be mentioned (e.g. the *composite prototype model* of James Hampton, and the *constraints theory* of Fintan Costello, cf. [4]), but the final result is the same: the challenge is hard and there is no consensus.

A concept-centered approach to interpret a modifier-head compound is presented in [3], where the acquisition of implicit relationships between the modifier and the head is captured by means of their linguistic *relational possibilities*. It depends on a generation, followed by a validation, of some matching relational possibilities between both the modifier and the head in the noun-noun compound. Unlike many others, this approach is concept-centered. Unlike ours, however, it is linguistic-oriented and language-dependent (i.e. English-based), so the approach may be difficult to apply to situations where online concept creation is needed in achieving a general intelligence level. The approach we present here (cf. Section 3.2) does not yet present an account that uses such kind of relational pos-

[5] Concepts in general are relativistic notions, and are sensitive to many sources of change, e.g. think about the relativity of a concept like BIG and the changes in meaning over time of the concept COMPUTER: clerk, huge machine, PC, laptop, portable or handheld device, and so on.

sibilities using both the modifier and the head. Only the modifier plays the big role, and only an analogy-based relation (e.g. "looks-like") is implicitly assumed. We partly follow [13, 37, 38] in that relational possibilities may only be suggested by the modifier, which is the source concept in our case.

3 TRIGGERING NOUN-NOUN BLENDS

An essential assumption taken by all blending approaches is the organization of knowledge in some form of *domains*. This means that an underlying KB should provide *concepts* and *facts* in *groups*, which serve as input to the blending process. The different domains may in principle be incoherent and even mutually contradictory, but they are nonetheless interconnected in a network organized by relations like generalization, analogy, projection, and instantiation. We assume that the knowledge base of concepts, $\mathbb{K}_C$, is available at our disposal, and that within which representations of concepts $c \in \mathbb{K}_C$ already exist (we also use the term 'domains'). Inspired by the "language-of-thought" hypothesis [11], the agents in our model are thus assumed to have some available concept representations that correspond to words. One method of combining those concepts into others needs to be developed in a way that reflects, to an acceptable extent, the meaning the human beings invent in similar situations.

3.1 HDTP and the Role of Analogy Making

The underlying framework is *Heuristic-Driven Theory Projection* (HDTP), which is a powerful analogy making system for computing analogical relations between two domains (theories) axiomatized in many-sorted, first-order predicate logic (with equality). Cross-domain reasoning can also be allowed in many different ways (see [1, 20, 21, 28, 34] for more details about HDTP and an expanded elaboration of its application domains). Given an analogical relation between source and target domains, knowledge can be transferred between them via analogical transfer. HDTP applies the syntactic mechanism of anti-unification [32] to find generalizations of formulas and to propose an analogical relation (cf. Figure 2).

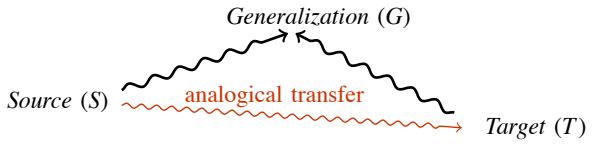


Figure 2. HDTP's overall approach to creating analogies (cf. [34]).

Analogical transfer results in *structure enrichment* of the target side, which usually corresponds to the addition of new axioms to the target theory, but may also involve the addition of new first-order symbols. There are cases in which analogical transfer is desired in order to create a new enriched domain, while keeping the original target domain unchanged. In such cases the generalization, source, target, and enriched domains are interconnected by a blend. This is clarified in the following section, where we posit a way, by which HDTP creates blends that represent the novel combinations. An example of the kind of blending, and of structure enrichment is also given below (cf. Sections 3.2 and 3.3).

The presented method of blending is inspired by the way humans create analogies [17]. The intuition here is that, while we are in the thinking process of what meaning to assign to a new modifier-head compound, we 'invent' the required meaning online, using a concept creation process: we first imagine a virtual copy of the head that is analogical to the modifier in some sense, then pick particular traits of the modifier and add them to this copy. In such a process, the newly-created word meaning can be a combination of the characteristics of the two words appearing in the compound, depending on how much in common the two words have and on our background knowledge. Using our model's terms, the beliefs that define the newly-created concept result from blending the beliefs defining the composing concepts. The resulting characteristics depend on the organized beliefs of the modifier and head concepts, on the previous experience, as well as on how may a head "look like" when it is attributed to the modifier (e.g. how may a BOX look like when it is attributed to a BOOK in the compound BOOK BOX). We emphasize again that we do not claim that our intuition explains the thinking mechanisms that take place during the actual cognitive processes. There are some inspiring reasons, however, why we are proposing that the combination can be computed in this way (e.g. the principles given in [4, 24], the discussions in [20], the developmental psychology literature in [25, 27, for instance], and the studies and experimental results of [37, 38]).

3.2 Concept Blending using HDTP

According to standard theory, a word is understood by the company of words it keeps [10] or, according to the HDTP's jargon, by the *background knowledge* an agent possesses about the words as well as about the context in which they appear. Inspired by human beings, where "a person has a repertoire of available concepts and ways of combining those concepts into higher-order concepts and into propositions" [22], we assume that our cognitive agents have already enough HELCOs, $c \in \mathbb{K}_C$, which represent the nouns they have already known (i.e. $e_x V(c) \geq \eta$).

We confine ourselves to a specific set of noun-noun composites, namely the modifier-head compounds. Although many alternative ways of paraphrasing such compounds may exist, the way their meanings are interpreted by human subjects seem to be frequently encountered (as shown in [37, 38]). The relational possibilities here can be suggested only by the modifier, i.e. the source (cf. [13]). We write a modifier-head noun-noun combination in the form B="$S\ T$", with the second noun T being the *head*. Since the first noun S functions as a *modifier* that adapts the meaning of the head, a combination "$S\ T$" in such cases is interpreted by agents as a function application $S(T)$ (because S acts, in a sense, as an operator on T that changes T's meaning [38]). Accordingly, we use an axiomatization of the operator S as the SOURCE domain for HDTP, and an axiomatization of the head T as the TARGET. In this way, HDTP can *blend* the two given nouns (as concepts) and use the blend to interpret their combination (see also [28]).

Given source and target domain representations S and T, respectively, we sketch how HDTP can be used to implement some crucial parts of our cognitively-based theory of CB for interpreting novel noun compounds. For a combination B="$S\ T$", once S and T are represented as sorted, first-order logic theories, they are provided to HDTP as SOURCE and TARGET concepts, respectively. Selecting S as the SOURCE, and not the TARGET, is based on the previous discussions and the principles of analogical reasoning [17]. This allows the transfer of knowledge, during analogical reasoning, in only one direction (and not the other) to pave the way for the "composition" and "completion" steps of CB to work (cf. Section 2.1). HDTP is applied next to the inputs, SOURCE and TARGET, and a blend results that

gives a possible interpretation of the compound, B. Some formalizations are given in Table 1, along with the corresponding illustrations of Figure 3 and the example discussion in Section 3.3.

Whenever an analogy is established, HDTP first provides an explicit generalization, G, of S and T (cf. Figure 2). G can be a base for concept creation by abstraction, and HDTP proceeds next in two phases: (1) in the *mapping phase*, S and T are compared to find structural commonalities (corresponding to the 'identification' between SOURCE and TARGET shown in Figure 1), and a generalized description is created that subsumes the matching parts of both domains, and then (2) in the *transfer phase*, unmatched knowledge in the source domain is mapped to the target domain to establish new hypotheses. It is important to note that, types of implicit relationships between the modifier and the head may be suggested and established during the transfer phase.

Table 1. Parts of suggested noun axiomatizations and their combination.

Source Axiomatization S ="SNAKE"

$$\forall x\ \exists w\ \mathit{Width}(x,w) \quad (1a)$$
$$\forall x\ \exists l\ \mathit{Length}(x,l) \quad (1b)$$
$$\forall x\ \mathit{Typical}_1(x) \rightarrow \mathit{Shape}(x,\mathit{curved}) \wedge \mathit{Skin}(x,\mathit{scaled}) \quad (1c)$$
$$\forall x \exists l\ \exists w\ \mathit{Length}(x,l) \wedge \mathit{Width}(x,w) \rightarrow l > w \quad (1d)$$

Target Axiomatization T ="GLASS"

$$\forall x\ \exists w\ \mathit{Width}(x,w) \quad (2a)$$
$$\forall x\ \exists h\ \mathit{Height}(x,h) \quad (2b)$$
$$\forall x\ \mathit{Typical}_2(x) \rightarrow \mathit{Transparent}(x) \wedge \mathit{Fragile}(x) \quad (2c)$$

Blend B ="SNAKE GLASS"

$$\forall x\ \exists w\ \mathit{Width}(x,w) \quad (3a)$$
$$\forall x\ \exists h\ \mathit{Height}(x,h) \quad (3b)$$
$$\forall x\ \mathit{Typical}(x) \rightarrow \mathit{Transparent}(x) \wedge \mathit{Fragile}(x) \quad (3c)$$
$$\forall x\ \mathit{Typical}(x) \rightarrow \mathit{Shape}(x,\mathit{curved}) \wedge \mathit{Skin}(x,\mathit{scaled}) \quad (3d)$$
$$\forall x \exists h\ \exists w\ \mathit{Height}(x,h) \wedge \mathit{Width}(x,w) \rightarrow h > w \quad (3e)$$

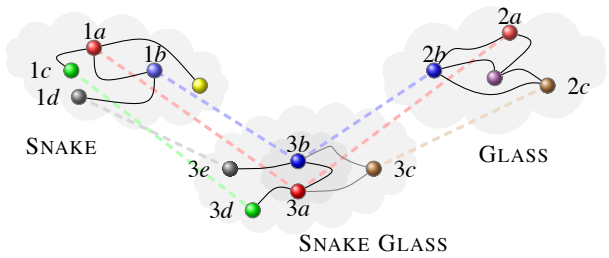


Figure 3. 'SNAKE GLASS' is a noun-noun blend, which results from the transfer phase of the blending between 'SNAKE' and 'GLASS' (cf. Table 1).

3.3 Compound Interpretation: An Example

As a specific instance, consider SNAKE GLASS, which some humans described as a "tall, very thin drinking glass" [38]. The example given here illustrates the blend of (partial formalizations of) the domains (theories) representing the source and target nouns SNAKE and GLASS, respectively (cf. Table 1). The blended domain, SNAKE GLASS, is an expansion of GLASS, the target, in which notions of 'shape' and 'skin' taken from SNAKE are added. In principle, the blended domain can be thought of as coming from enriching the first-order theory by which the target is represented with new notions taken from the source, and then importing the axioms of the source into it (cf. Figure 3).

Irrespective of whether or not other constituents are included in the representation, a formalization of the concept SNAKE should normally emphasize the existence of some *salient* SNAKE characteristics. A suggested formalization is given in Table 1, in which the common-sense emphasis is on a SNAKE having a length that is much bigger than its width, a curved body shape, and a skin that is covered in scales. The characteristics that a typical GLASS exemplar must have, among other things, are its transparency and fragility. A GLASS object also has dimensions determining its width and height. A blend of the two concepts that represent SNAKE and GLASS would, consequently, import the properties of a SNAKE that do not conflict with the GLASS representation. In particular, a blend will indicate a relation between the *dimensions* of the SNAKE GLASS. Specifically, HDTP identifies $(1a)$ and $(1b)$ with $(2a)$ and $(2b)$, and infers from $(1d)$ that one of the dimensions of a SNAKE GLASS will be much larger than the other. A SNAKE GLASS would, in addition to the non-conflicting GLASS constituents, have a curved shape, as well as other non-conflicting constituents of a SNAKE (cf. Table 1 and Figure 3).

In general, concept representation depends both on the *granularity level* (that is needed to capture the understood meaning in a specific application domain) and the background knowledge. For example in Table 1, $\mathit{Typical}_i(x)$, for $i \in \{1,2\}$, can be defined in a variety of ways, depending on how concepts are represented (and depending on previous experiences as well, i.e. on the value $e_xV(\mathit{Typical}_i(x))$).

It is worth noting that the given framework does not function in the sense that two given nouns will only (or always) produce a unique result. In fact, experiments show that humans too do not always agree on one meaning of the same given noun-noun combination, neither do they exactly follow one particular model each time they encounter a similar combination [27, 37, 38]. The framework rather enumerates alternatives ranked by the complexity of the underlying mappings. In our view, this is a desirable property because: (1) it allows possible interpretations instead of just one, and also (2) gives a space for experience to play a role in deciding whether or not a specific blend is favored over another. People also interpret novel combinations by drawing on past experience with similar combinations [12].

Without going into further details, we need to point out that every SNAKE GLASS blend is intended to be represented by a LEVCO $B_i \in \mathbb{K}_C$ with $0 < e_xV(B_i) < \eta$, such that the calculation of the value $e_xV(B_i)$ is affected by $e_xV(S)$ and $e_xV(T)$ of the source and target HELCOs. How $e_xV(c)$ values of the LEVCOs $c \in \mathbb{K}_C$ can be computed? or how implicit relationships can be retrieved during the transfer phase in the analogy-making process? are the main questions that will be considered in a later study.

4 CONCLUSION AND FUTURE WORK

Finding a meaning of a (novel) combination is a difficult creative task, yet providing a computational account that simulates human cognition is an even more difficult one. The basic challenges of the problem motivated us to contribute to solving it by presenting a computational, concept-based, cognitively-inspired, language-independent approach. The feasibility of computing a blend in the described manner exemplifies our suggestion of how this form of noun-noun combinations could be approached. On the one hand, the

use of rated experiences and levels of entrenchment for the represented concepts can help in achieving solutions to some challenges, such as when concepts get changed or externally affected. The way analogy is made use of in identifying common parts of the source and target concepts of a modifier-head compound, in generalizing them, and creating blends, can serve maintaining relational and attributive combinations at the same time. On the other hand, the implicit relational possibility that analogy provides us with between the head and the modifier still does not account on many of the different cases that can be encountered (e.g: the combination B="S T" is interpreted as "T that looks-like S" or "T that is in-the-form-of S"), but it is promising and could be improved. The method presented here may be considered as a first starting step towards the interpretation of noun-noun compounds using a new perspective. Of course, neither HDTP nor CB intend to solve the challenges altogether. The method allows, however, a feasible form of blending that respects the dual process of comparison and integration, on which famous models are based (cf. [7, 12, 24, 26, 37]).

REFERENCES

[1] Ahmed Abdel-Fattah, Tarek R. Besold, Helmar Gust, Ulf Krumnack, Martin Schmidt, Kai-Uwe Kühnberger, and Pei Wang, 'Rationality-Guided AGI as Cognitive Systems', in *Proc. of the the 34th annual meeting of the Cognitive Science Society (to appear)*, (2012).

[2] James Alexander, 'Blending in Mathematics', *Semiotica*, **2011**(187), 1–48, (2011).

[3] Cristina Butnariu and Tony Veale, 'A concept-centered approach to noun-compound interpretation', in *COLING*, eds., Donia Scott and Hans Uszkoreit, pp. 81–88, (2008).

[4] Fintan J. Costello and Mark T. Keane, 'Efficient creativity: constraint-guided conceptual combination', *Cognitive Science*, **24**(2), 299–349, (2000).

[5] S. Coulson, *Semantic Leaps: Frame-Shifting and Conceptual Blending in Meaning Construction*, Cambridge University Press, 2006.

[6] Marco Dorigo and Thomas Stützle, *Ant Colony Optimization*, Bradford Books, MIT Press, 2004.

[7] Zachary Estes, 'A tale of two similarities: comparison and integration in conceptual combination', *Cognitive Science*, **27**, 911–921, (2003).

[8] Gilles Fauconnier and Mark Turner, *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*, Basic Books, New York, 2002.

[9] Gilles Fauconnier and Mark Turner, 'Rethinking Metaphor', in *Cambridge Handbook of Metaphor and Thought*, ed., R. Gibbs, 53–66, Cambridge University Press, New York, (2008).

[10] John Rupert Firth, *Papers in linguistics 1934–51*, Oxford University Press, 1957.

[11] Jerry A. Fodor, *The Modularity of Mind: An Essay on Faculty Psychology*, Bradford Books, MIT Press, 1983.

[12] Christina L. Gagné, 'Lexical and relational influences on the processing of novel compounds', *Brain and Language*, **81**, 723 – 735, (2002).

[13] Christina L. Gagné and Edward J. Shoben, 'Influence of thematic relations on the comprehension of modifier-noun combinations', *Journal of Experimental Psychology: Learning, Memory and Cognition*, **23**(1), 71–87, (1997).

[14] Peter Gärdenfors, *Knowledge in Flux : Modeling the Dynamics of Epistemic States*, MIT Press, Cambridge, Massachusetts, 1988.

[15] Peter Gärdenfors and David Makinson, 'Revisions of knowledge systems using epistemic entrenchment', in *TARK '88: Proceedings of the 2nd conference on Theoretical aspects of reasoning about knowledge*, pp. 83–95. Morgan Kaufmann Publishers Inc., (1988).

[16] L.S. Gay and W.B. Croft, 'Interpreting nominal compounds for information retrieval', *Information Processing and Management*, **26**(1), 21–38, (1990).

[17] *The Analogical Mind: Perspectives from Cognitive Science*, eds., D. Gentner, K. Holyoak, and B. Kokinov, MIT Press, 2001.

[18] Joseph Goguen, 'Mathematical models of cognitive space and time', in *Reasoning and Cognition: Proc. of the Interdisciplinary Conference on Reasoning and Cognition*, eds., D. Andler, Y. Ogawa, M. Okada, and S. Watanabe, pp. 125–128. Keio University Press, (2006).

[19] Markus Guhe, Alison Pease, Alan Smaill, Maricarmen Martínez, Martin Schmidt, Helmar Gust, Kai-Uwe Kühnberger, and Ulf Krumnack, 'A computational account of conceptual blending in basic mathematics', *Cognitive Systems Research*, **12**(3–4), 249–265, (2011).

[20] Helmar Gust, Ulf Krumnack, Maricarmen Martínez, Ahmed Abdel-Fattah, Martin Schmidt, and Kai-Uwe Kühnberger, 'Rationality and General Intelligence', in *Artificial General Intelligence*, eds., J. Schmidhuber, K. Thorisson, and M. Looks, pp. 174–183, (2011).

[21] Helmar Gust, Kai-Uwe Kühnberger, and Ute Schmid, 'Metaphors and Heuristic-Driven Theory Projection (HDTP)', *Theor. Comput. Sci.*, **354**, 98–117, (March 2006).

[22] James A Hampton, 'Conceptual combination', In Lamberts and Shanks [25], 133–161.

[23] S.O. Hansson, *A Textbook of Belief Dynamics: Theory Change and Database Updating*, number b. 1 in Applied Logic Series, Kluwer Academic Publishers, 1999.

[24] Mark T. Keane and Fintan J. Costello, 'Setting limits on analogy: Why conceptual combination is not structural alignment', In Dedre Gentner and Kokinov [17], 172–198.

[25] *Knowledge, Concepts, and Categories*, eds., Koen Lamberts and David Shanks, MIT Press, 1997.

[26] Judith N. Levi, *The Syntax and Semantics of Complex Nominals*, Academic Press, New York, 1978.

[27] Denis Mareschal, Paul C. Quinn, and Stephen E. G. Lea, *The Making of Human Concepts*, Oxford Series in Developmental Cognitive Neuroscience, Oxford University Press, 2010.

[28] M. Martinez, T. R. Besold, Ahmed Abdel-Fattah, K.-U. Kühnberger, H. Gust, M. Schmidt, and U. Krumnack, 'Towards a domain-independent computational framework for theory blending', in *AAAI Technical Report of the AAAI Fall 2011 Symposium on Advances in Cognitive Systems*, pp. 210–217, (2011).

[29] G.L. Murphy, *The Big Book of Concepts*, Bradford Books, Mit Press, 2004.

[30] Francisco C. Pereira and Amílcar Cardoso, 'Optimality principles for conceptual blending: A first computational approach', *AISB Journal*, **1**, (2003).

[31] Francisco Câmara Pereira, *Creativity and AI: A Conceptual Blending Approach*, Applications of Cognitive Linguistics (ACL), Mouton de Gruyter, Berlin, December 2007.

[32] Gordon D. Plotkin, 'A note on inductive generalization', *Machine Intelligence*, **5**, 153–163, (1970).

[33] Mary Ellen Ryder, *Ordered chaos: the interpretation of English noun-noun compounds*, volume 123 of *Linguistics*, University of California Press, 1994.

[34] Angela Schwering, Ulf Krumnack, Kai-Uwe Kühnberger, and Helmar Gust, 'Syntactic Principles of Heuristic-Driven Theory Projection', *Journal of Cognitive Systems Research*, **10**(3), 251–269, (2009).

[35] Wolfgang Spohn, 'Ordinal conditional functions – a dynamic theory of epistemic states', in *Causation in decision, belief change, and statistics*, ed., William L. Harper, volume 2 of *Proceedings of the Irvine Conference on Probability and Causation*, pp. 105–134. Kluwer, Dordrecht, (1988).

[36] Tony Veale and Diarmuid O'Donoghue, 'Computation and Blending', *Computational Linguistics*, **11**(3–4), 253–282, (2000). Special Issue on Conceptual Blending.

[37] Edward J. Wisniewski, 'When concepts combine', *Psychonomic Bulletin & Review*, **4**(2), 167–183, (1997).

[38] Edward J. Wisniewski and Dedre Gentner, 'On the combinatorial semantics of noun pairs: Minor and major adjustments to meaning', in *Understanding Word and Sentence*, ed., G.B. Simpson, Elsevier Science Publishers B.V. (North-Holland), (1991).

[39] Stefan Wrobel, *Concept Formation and Knowledge Revision*, Kluwer, 1994.

Ontological Blending in DOL

Oliver Kutz, Till Mossakowski, Joana Hois, Mehul Bhatt, John Bateman [1]

Abstract. We introduce ontological blending as a method for combining ontologies. Compared with existing combination techniques that aim at integrating or assimilating categories and relations of thematically related ontologies, *blending* aims at creatively generating (new) categories and ontological definitions; this is done on the basis of input ontologies whose domains are thematically distinct but whose specifications share structural or logical properties. As a result, ontological blending can generate new ontologies and concepts and it allows a more flexible technique for ontology combination compared to existing methods.
Our approach to computational creativity in conceptual blending is inspired by methods rooted in cognitive science (e.g., analogical reasoning), ontological engineering, and algebraic specification. Specifically, we introduce the basic formal definitions for ontological blending, and show how the distributed ontology language DOL (currently being standardised within the OntoIOp—Ontology Integration and Interoperability—activity of ISO/TC 37/SC 3) can be used to declaratively specify blending diagrams.

1 Introduction

Well-known techniques directed towards unifying the semantic content of different ontologies, namely techniques based on matching, aligning, or connecting ontologies, are ill-suited to either re-use (proven) axioms from one ontology in another or generate new conceptual schemas from existing ontologies, as it is suggested by the general methodology of conceptual blending introduced by Fauconnier and Turner [11]: here, the blending of two thematically rather different *conceptual spaces* yields a new conceptual space with emergent structure, selectively combining parts of the given spaces whilst respecting common structural properties.[2] The 'imaginative' aspect of blending is summarised as follows [39]:

> [...] the two inputs have different (and often clashing) organising frames, and the blend has an organising frame that receives projections from each of those organising frames. The blend also has emergent structure on its own that cannot be found in any of the inputs. Sharp differences between the organising frames of the inputs offer the possibility of rich clashes. Far from blocking the construction of the network, such clashes offer challenges to the imagination. The resulting blends can turn out to be highly imaginative.

A classic example for this is the blending of the concepts *house* and *boat*, yielding as most straightforward blends the concepts of a *houseboat* and a *boathouse*, but also an *amphibious vehicle* [16].

In the almost unlimited space of possibilities for combining existing ontologies to create new ontologies with emergent structure, conceptual blending can be built on to provide a structural and logic-based approach to 'creative' ontological engineering. This endeavour primarily raises the following two challenges: (1) when combining the terminologies of two ontologies, the shared semantic structure is of particular importance to steer possible combinations. This shared semantic structure leads to the notion of base ontology, which is closely related to the notion of 'tertium comparationis' found in the classic rhetoric and poetic theories, but also in more recent cognitive theories of metaphor (see, e.g., [23]); (2) having established a shared semantic structure, there is typically still a huge number of possibilities that can capitalise on this information in the combination process: here, optimality principles for selecting useful and interesting blends take on a central position.

We believe that the principles governing ontological blending are quite distinct from the rather informal principles employed in blending phenomena in language or poetry, or the rather strict principles ruling blending in mathematics, in particular in the way formal inconsistencies are dealt with. For instance, whilst blending in poetry might be particularly inventive or imaginative when the structure of the basic categories found in the input spaces is almost completely ignored, and whilst the opposite, i.e., rather strict adherence to sort structure, is important in areas such as mathematics in order to generate meaningful blends[3], ontological blending is situated somewhere in the middle: re-arrangement and new combination of basic categories can be rather interesting, but has to be finely controlled through corresponding interfaces, often regulated by or related to choices found in foundational or upper ontologies.

We start with a discussion of alignment, matching, analogical reasoning, and conceptual blending, vis-à-vis ontological blending. The core contributions of the paper[4] can be summarised as follows; we:

- give an abstract definition of ontological blendoids capturing the basic intuitions of conceptual blending in the ontological setting;
- provide a structured approach to ontology languages, in particular to OWL-DL[5], by employing the OWL fragment of the distributed ontology language DOL for blending, namely DOL-OWL. This combines the simplicity and good tool support for OWL with the more complex blending facilities of OBJ3 [17] or Haskell [25];
- analyse the computational and representational issues that blending with ontology languages raises, and outline some of the first optimality principles for ontological blending;

[1] Research Center on Spatial Cognition (SFB/TR 8), University of Bremen, Germany. Corresponding author: okutz@informatik.uni-bremen.de

[2] The usage of the term 'conceptual space' in blending theory is not to be confused with the usage established by Gärdenfors [13].

[3] For instance when creating the theory of transfinite cardinals by blending the perfective aspect of counting up to any fixed finite number with the imperfective aspect of 'endless counting' [34].

[4] This paper elaborates on ideas first introduced in [20].

[5] In the remainder of this paper we refer to OWL-DL Version 2 by just OWL. See `http://www.w3.org/TR/owl2-overview/`

The contributions are illustrated in detail with a fully formalised example of an ontological blend, involving signs (signposts) and forests.

2 Ontology Alignment and Conceptual Blending

For a given domain, often several ontologies exist which need to be related in order to achieve coverage of the required knowledge. For instance, heterogeneous sources may provide ontological information on the same kind of data, and their information needs to be integrated with each other. Various kinds of relations between these types of ontologies have been studied in the literature, amongst them mapping and matching, alignment, coordination, transformation, translation, merging, reconciliation, and negotiation (cf. [6]). Some of these techniques, in particular matching and alignment, are typically based on statistical approaches and similarity measures [24, 10].[6]

From these techniques, alignments are most closely related to our present purpose because they can be seen as a strict, i.e., 'uncreative', version of blending. Alignments completely identify or separate information, in particular, they try to find semantically related concepts or relations from two given ontologies. They seek out commonalities between these concepts or relations by inspecting surface data, e.g., concept and relation names. However, they typically ignore their logical information, namely the axiomatisations of the ontologies. The quality of detected alignments is typically assessed by comparison to a previously defined gold-standard based on standard precision and recall methods.[7] In general, alignments are most useful for combining ontologies that specify thematically closely related domains.

The alignment operation between two ontologies was first formalised from a category-theoretic standpoint in [41], using pushouts and colimits, and further refined in [26]. A pushout links two given ontologies using a common interface theory. While the ontologies are disjointly united, the two copies of the common interface theory are identified. For example, if ontology O_1 features a concept Human, while O_2 provides Person, a corresponding concept should occur in the common interface theory and be mapped to Human and Person, respectively. The effect is that in the alignment (formalised as a pushout), Human and Person are identified. In contrast, if concepts do not appear in the common interface, they are kept apart, *even if they happen to have the same name* (cf. Bank in the example).

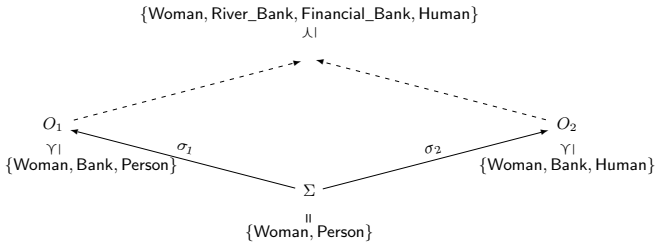


Figure 1. V-alignment: integration through interface

This construction, called V-alignments, can deal with basic alignment problems such as **synonyms** (identifying different symbols with the same meaning) and **homonyms** (separating (accidentally) identical symbols with different meaning)—see Fig. 1. Alignments, however, can support only these basic types of relations between two ontologies having thematically overlapping domains. Combinations of thematically different ontologies can easily become more complex, for instance, when dealing with **analogies** (relating different symbols based on their similar axiomatisation), **metaphors** (blending symbols from one domain into another and impose the axiomatisation of the first on the second), **pataphors** (blending and extending two domains with each other), or **conceptual blending** (blending and combining two domains for the creation of new domains). In contrast to alignments, blending thus combines two potentially thematically unrelated ontologies in a way such that new structure can emerge. Below, we define and formalise this blending operation accordingly.

In [35], conceptual blending is implemented in terms of analogy finding applied to an automatic text generation system. Particularly, for metaphorical phrasing, the tool *jMapper* compares the instances of two given input domains with each other and calculates the similarity between instances of the source and the target domain. This is based on shared properties and relationships of the domain's instances, for which thresholds can be varied. However, the *jMapper* tool does not aim at creating 'new' domains. It only works with instance definitions as input domains in a proprietary format rather than re-using standardised ontology languages.

In [25], blending is based on structural aspects of two different domains. The example of blending boat and house is here based on image schemata, namely, categories and relations from the house and boat domains are related to particular image schemata such as *container* and *surface*. The image schemata are used as an abstraction necessary for blending two domains. The boat and house example is implemented using Haskell type classes, which, however, results in rigidly blended classes for houseboat and boathouse. For instance, only a 'boat' can be an 'inhabitant' of a 'boathouse'. Any other (conceptually possible) type, such as a caretaker residing in a boathouse, contradicts this definition. Conceptual blending in general does not exhibit this kind of strong restriction.

In [16], conceptual blending is formalised categorically, focusing on the structural aspects of the blending process. In the following, we adapt this approach to ontological engineering.

3 Introducing Ontological Blending

Goguen has created the field of *algebraic semiotics* which logically formalises the structural aspects of semiotic signs, sign systems, and their mappings [15]. In his joint work with Fox Harrell [16], algebraic semiotics has been applied to user interface design and blending. Algebraic semiotics does not claim to provide a comprehensive formal theory of blending—indeed, Goguen and Harrell admit that many aspects of blending, in particular concerning the meaning of the involved notions, as well as the optimality principles for blending, cannot be captured formally. However, the structural aspects *can* be formalised and provide insights into the space of possible blends.

Goguen defines semiotic systems to be algebraic theories that can be formulated by using the algebraic specification language OBJ [17]. Moreover, a special case of a semiotic system is a *conceptual space*: it consists only of constants and relations, one sort, and axioms that define that certain relations hold on certain instances.

As we focus on standard ontology languages, namely OWL and first-order logic, we here replace the logical language OBJ. As structural aspects in the ontology language are necessary for blending, we augment these languages with structuring mechanisms known from

[6] Ontology matching and alignment based on such methods is an established field on its own having yearly competitions since 2004 (see `http://oaei.ontologymatching.org/`).

[7] See [19] for an extensive analysis. The lack of semantics involved in such an evaluation process has been clearly articulated already in [9].

algebraic specification theory [27]. This allows to translate most parts of Goguen's theory to these ontology languages. Goguen's main insight has been that semiotic systems and conceptual spaces can be related via *morphisms*, and that blending is comparable to *colimit* construction. In particular, the blending of two concepts is often a *pushout* (also called *blendoid* in this context). Some basic definitions:

An OWL **signature** consists of sets of class names, role names, and individual names. An OWL **signature morphism** between two OWL signatures consists of three mappings between the respective sets. OWL sentences over a given signature Σ are defined as in [22], e.g., subsumptions between classes, role hierarchies, and instances of classes and roles, etc. OWL models provide a domain of individuals and interpret classes as subsets, roles as binary relations, and individuals as elements of the domain. Satisfaction of sentences in a model is defined in a standard way, see [22] for details. Moreover, given a signature morphism $\sigma : \Sigma_1 \to \Sigma_2$ and a Σ_2-model M_2, the **reduct** $M_2|_\sigma$ is the Σ_1-model that interprets a symbol by first translating it along σ and then looking up the interpretation in M_2.

On top of this, we define the language DOL-OWL and its model-theoretic semantics as follows.[8] A DOL-OWL ontology O can be

- a **basic** OWL theory $\langle \Sigma, \Gamma \rangle$; Σ is a signature, Γ a set of Σ-sentences, with $\mathbf{Mod}(\langle \Sigma, \Gamma \rangle)$ containing all Σ-models satisfying Γ;
- a **translation**, written O **with** σ, (where $\sigma : \Sigma_1 \to \Sigma_2$) with $\mathbf{Mod}(O \textbf{ with } \sigma) = \{M \in \mathbf{Mod}(\Sigma_2) \mid M|_\sigma \in \mathbf{Mod}(O)\}$;
- a **union**, written O_1 **and** O_2, of ontologies over the same signature, with $\mathbf{Mod}(O_1 \textbf{ and } O_2) = \mathbf{Mod}(O_1) \cap \mathbf{Mod}(O_2)$[9];
- a **hiding**, written O **hide** σ, with $\mathbf{Mod}(O \textbf{ hide } \sigma) = \{M|_\sigma \mid M \in \mathbf{Mod}(O)\}$.

A DOL-OWL library statement can be

- an ontology definition **ontology** $O\_NAME = O$; or
- a interpretation, written **interpretation** $INT\_NAME : O_1$ to $O_2 = \sigma$.

An interpretation is **correct**, if σ is a theory morphism from O_1 to O_2, that is, for every O_2-model M_2, its reduct $M_2|_\sigma$ is an O_1-model. This definition provides a structural approach in DOL-OWL, that can be compared with instantiation of type variables in Haskell and type casting in OBJ3.

Since in some blends, not the whole theory can be mapped, Goguen [15] introduces partial signature morphisms. Here, we follow a common idea in category theory and model partial theory morphisms $\sigma : T_1 \dashrightarrow T_2$ as spans

$$T_1 \xleftarrow{\sigma_-} dom\ \sigma \xrightarrow{\sigma_+} T_2$$

of ordinary (total) theory morphisms satisfying a well-definedness condition; this has the advantage of keeping the theory simple. σ_- is the inclusion of $dom\ \sigma$ (the domain of σ) into T_1, while σ_+ is the action of the partial theory morphism. If σ_- is an isomorphism, we say that σ is total, it can then be identified with the ordinary morphism $\sigma_+ \circ \sigma_-^{-1} : T_1 \to T_2$:

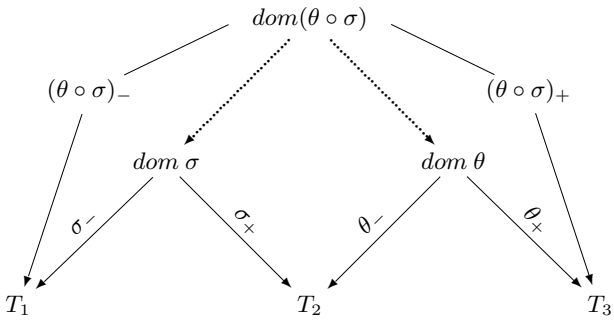
$$T_1 \xrightarrow{\sigma_-^{-1}} dom\ \sigma \xrightarrow{\sigma_+} T_2$$

[8] The definition of DOL-OWL as given here corresponds essentially to the fragment of the distributed ontology language DOL that homogeneously uses OWL modules. The full DOL language however comprises several additional features, and supports a large number of ontology languages, see [32] for a presentation of the full semantics.

[9] Unions over different signatures can be modelled using translations.

The well-definedness condition for partial theory morphisms $\sigma : T_1 \dashrightarrow T_2$ is similar to but more general than that for ordinary theory morphisms: for each T_2-model M_2, its reduct $M_2|_{\sigma_+}$ must be "somehow" a T_1-model. The "somehow" can be made precise as follows: for each T_2-model M_2, there must be a T_1-model M_1 such that $M_1|_{\sigma_-} = M_2|_{\sigma_+}$. Equivalently, $\sigma_+ : (T_1 \textbf{ hide } \sigma_-) \to T_2$ is an ordinary theory morphism (note that the models of T_1 **hide** σ_- are precisely those models that are σ_--reduct of some T_1-model).

We now recall some notions from category theory, see [1, 41] for further details. A **diagram** D consists of a graph of ontologies $(D_i)_{i \in |D|}$ and total theory morphisms $(D_m : D_i \to D_j)_{m \in D}$ among them. Partial theory morphisms can easily be dealt with: diagrams just get a little larger when spans are used. For a diagram D, a **partial sink** consists of an ontology O and a family of partial theory morphisms $(\mu_i : D_i \dashrightarrow O)_{i \in |D|}$. A **sink** is a partial sink consisting of total morphisms only. A partial sink is an epi-sink, if $f \circ (\mu_i)_- = g \circ (\mu_i)_-$ for all $i \in |D|$ implies $f = g$. A partial sink is **weakly commutative** if all emerging triangles commute weakly, i.e., for all $m : i \to j \in D$, we have that $D_m \circ \mu_i = \mu_j$ as partial morphisms. Such compositions of partial morphisms are obtained by pullback:



For total sinks, weak commutativitiy amounts to ordinary commutativity; the sink in this case is called a **co-cone**. A co-cone is a **colimit**, if it can be uniquely naturally embedded into any co-cone (hence, it can be seen as a minimal co-cone). [1] also show that colimits are epi-sinks.

We now give a general definition of ontological blending capturing the basic intuition that a blend of input ontologies shall partially preserve the structure imposed by base ontologies, but otherwise be an almost arbitrary extension or fragment of the disjoint union of the input ontologies with appropriately identified base space terms.

Definition 1 (Ontological Base Diagram) *An **ontological base diagram** is a diagram D for which the minimal nodes $(B_i)_{i \in D_{min} \subseteq |D|}$ are called **base ontologies**, the maximal nodes $(I_j)_{j \in D_{max} \subseteq |D|}$ called **input ontologies**, and where the partial theory morphisms $\mu_{ij} : B_i \dashrightarrow I_j$ are the **base morphisms**. If there are exactly two inputs I_1, I_2, and one base $\mathcal{B}$, the diagram D is called **classical** and has the shape of a V (for total morphisms) or W (for partial morphisms). In this case, $\mathcal{B}$ is also called the **tertium comparationis**.*

The basic, i.e., classical, case of an ontological base diagram with total morphisms is illustrated in the lower part of Fig. 2. In general, however, ontological blending can deal with more than one base and two input ontologies. [8], for instance, discusses the example of blending the input domains *politics*, *American culture*, and *sports*, in order to create the metaphor "He's a guy who was born on third base and thinks he hit a triple." [8, p. 172] (a criticism of George Bush).

Definition 2 (Ontological Blendoid) *Let D be a base diagram. A*

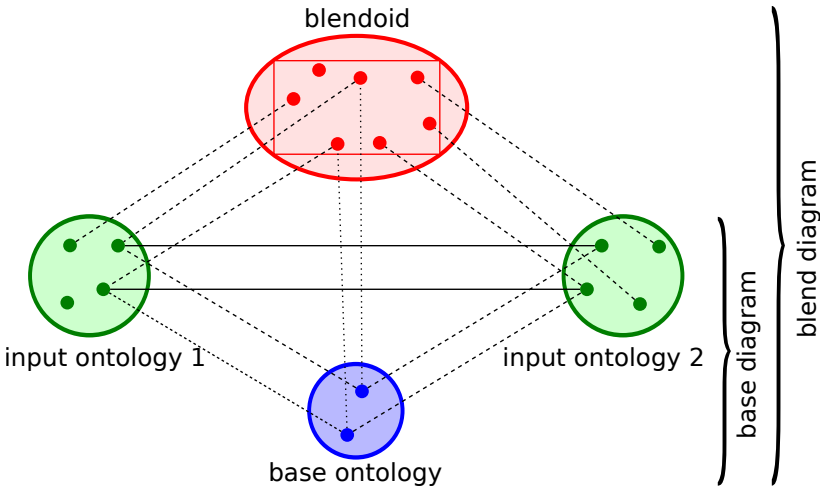


Figure 2. The basic integration network for blending: concepts in the base ontology are first refined to concepts in the input ontologies and then selectively blended into the blendoid.

blendoid $\mathfrak{B}$ *for* D *is a partial sink of* signature morphisms *over* D. *A blendoid is called*

- ***axiom-preserving****, if the signature morphisms of the partial sink are all theory morphisms;*
- ***closed****, if it is a (partial) epi-sink (which basically means that the blend is generated by the diagram), otherwise* ***open****;*
- ***total****, if the partial sink is a sink;*
- ***commutative****, if it is (weakly) commutative;*
- ***strict****, if it is a colimit (colimits are always epi-sinks, so closed).*

Here, axiom preservation, totality and commutativity can also hold to a certain degree. Consider the percentage of: signature morphisms that are theory morphisms (resp. total); and diagrams that commute.

Further note that an axiom-preserving strict blend where the base diagram has the form of a V and the base ontology is just a signature is nothing else but a V-alignment. Note that open blends might additionally import ontologies with new relevant signature.

Two crucial aspect of blends are (1) morphisms within the base diagram as well as into the blend diagram can be partial, and (2) the structure of the blend might partially violate the shared structure of the inputs ('violation of structure mapping').

In practice, open blends will typically be constructed by first generating a closed blend, and then subsequently aligning this with a new (thematically related) input ontology. In particular, this construction can be applied by aligning two different closed blends $\mathfrak{B}_1$ and $\mathfrak{B}_2$ obtained through the same base space $\mathcal{B}$ (here new signature elements can be created in the new colimit). For instance, we can align the blended ontologies for BoatHouse and HouseBoat by introducing houseboats as residents of boathouses. This **completion** by alignment or import can be seen as an analogue to the 'running of the blend' as it is discussed in conceptual blending [11].

Clearly, unless we construct a strict blendoid with a rather 'strong' base ontology, due to partiality there will always be exponentially many possibilities for the blend. Moreover, there are obviously infinitely many open blends regardless of partiality and the structure of the base. For instance, in the House and Boat blending formalised in [16], there are, in our terminology, 48 blendoids over a fixed base diagram that are axiom preserving, commutative and closed.[10]

[10] Note that this differs from the (slightly inconsistent) terminology in [16].

4 Computational and Representational Challenges

Conceptual blending has been proposed as a possible solution to get a handle on the notion of computational creativity [35]. The most sophisticated implementation to date related to blending probably is the tool described in [16] for the automated generation of poems. To create similar tools specifically dedicated to the realm of ontology, we have to address at least the following three issues:

1. The representational layer for blending needs to be specialised to ontology languages, in particular to one-sorted languages such as OWL, and languages such as Common Logic[11].
2. Given a couple (or a finite number of) ontologies, strategies are required to compute (rather than assume or handcraft) the common base ontology together with corresponding morphisms.
3. Given an ontological base diagram, techniques and heuristics are required that select interesting or useful blendoids according to genuine ontological principles. In particular, this requires new ranking and optimality principles.

We have addressed the first item already in the previous section: the language DOL-OWL allows for a structured specification of blend diagrams. Note that, more generally, mixed blend diagrams can be specified in the DOL language combining, besides several other ontology languages, first-order and OWL ontologies (see [28]). We next briefly discuss items 2. and 3.

4.1 Computing the Tertium Comparationis

To find candidates for base ontologies that could serve for the generation of ontological blendoids, much more shared semantic structure is required than the surface similarities that alignment approaches rely on. The common structural properties of the input ontologies that are encoded in the base ontology are typically of a more abstract nature. The standard example here relies on *image schemata*, such as the notion of a *container* mentioned earlier (see also [25]). Thus, in particular, foundational ontologies can support such selections. In analogical reasoning, 'structure' is (partially) mapped from a source domain to a target domain [12, 38]. Intuitively, then, the operation of

[11] See `http://common-logic.org/`

computing a base ontology can thus be seen as a bi-directional search for analogy.

We briefly discuss three promising candidates for this operation:

(1) **Ontology intersection:** [33] has studied the automatisation of theory interpretation search for formalised mathematics, implemented as part of the Heterogeneous Tool Set (HETS, see below). [29] applied these ideas to ontologies by using the ontologies' axiomatisations for finding their shared structure. Accidental naming of concept and role names is deliberately ignored and such names are treated as arbitrary symbols (i.e., any concept may be matched with any other). By computing mutual theory interpretations between the inputs, the method allows to compute a base ontology as an *intersection* of the input ontologies together with corresponding theory morphisms. While this approach can be efficiently applied to ontologies with non-trivial axiomatisations, lightweight ontologies are less applicable, e.g., 'intersecting' a smaller taxonomy with a larger one clearly results in a huge number of possible taxonomy matches [29]. In this case, the following techniques are more appropriate.

(2) **Structure-based ontology matching:** [37] address the problem that matching and alignment approaches are typically restricted to find simple correspondences between atomic entities of the ontology vocabulary. They define a number of *complex correspondence patterns* that can be used together with standard alignments in order to relate complex expressions between two input ontologies. For instance, the 'Class by Attribute Type Pattern' may be employed to claim the equivalence of the atomic concept PositiveReviewedPaper in ontology O_1 with the complex concept $\exists$hasEvaluation.Positive of O_2. Such an equivalence can be taken as an axiom of the base ontology; note, however, that it could typically not be found by intersecting the input ontologies. Giving such a library of design patterns may be seen as a variation of the idea of using image schemata.

(3) **Analogical Reasoning:** *Heuristic-driven theory projection* is a logic-based technique for analogical reasoning that can be employed for the task of computing a common generalisation of input theories. [38] establish an analogical relation between a source theory and a target theory (both first-order) by computing a common generalisation (called 'structural description'). They implement this by using anti-unification [36]. A typical example is to find a generalisation (base ontology) formalising the structural commonalities between the Rutherford atomic model and a model of the solar system. This process may be assisted by a background knowledge base (in the ontological setting, a related domain or foundational ontology). Indeed, this idea has been further developed in [30].

4.2 Selecting the Blendoids: Optimality Principles

Having a common base ontology (computed or given), there is typically a large number of possible blendoids. For example, even in the rather simple case of combining House and Boat, allowing for blendoids which only partially maintain structure (called *non-primary* blendoids in [16]), i.e., where any subset of the axioms may be propagated to the resulting blendoid, the number of possible blendoids is in the magnitude of 1000. Clearly, from an ontological viewpoint, the overwhelming majority of these candidates will be rather meaningless. A ranking therefore needs to be applied on the basis of specific ontological principles. In conceptual blending theory, a number of **optimality principles** are given in an informal and heuristic style [11]. While they provide useful guidelines for evaluating natural language blends, they do not suggest a direct algorithmic implementation, as also analysed in [16]. Moreover, the standard blending theory of [11] does not assign types, which might make sense in the case of linguistic blends where type information is often ignored. A typical example of a type mismatch in language is the operation of *personification*, e.g., turning a boat into an 'inhabitant' of the 'boathouse'. However, in the case of blending in mathematics or ontology, this loss of information is often rather unacceptable: to the opposite, a fine-grained control of type or sort information is of the utmost importance here.

Optimality principles for ontological blending will be of two kinds. (1) purely *structural/logical principles*: as introduced in Sec. 3, these will extend and refine the criteria as given in [16], namely **degree of commutativity** of the blend diagram, **type casting** (preservation of taxonomical structure), **degree of partiality** (of signature morphisms), and **degree of axiom preservation**. The relative ranking and importance of these metrics, however, will remain a case-by-case decision. In the context of OWL, typing needs to be replaced with preservation of specific axioms encoding the taxonomy. (2) *heuristic principles*: unlike the categorical modelling of alignments, blendings can often not be adequately described by a pushout operation. Some diagrams may not commute, and a more fine-grained control is required. This particularly explains why Goguen uses $3/2$ pushouts to specify blending [15]. Generalising blendoids to be $3/2$ pushouts allows for the integration of certain optimality principles in the blending process, namely an ordering of morphisms allowing to specify their quality (for instance in terms of their degree of partiality and type violation). Essentially, this introduces preference orders on possible morphisms, which can further be regulated by specific ontological principles. One candidate for regulating such preference orders, extending the purely structural optimality principles, would be adherence to the OntoClean methodology [18].

Existing Tool Support. For carrying out blending experiments using OWL, we use the DOL-OWL language and the Heterogeneous Tool Set HETS [31] which provides a prototypical implementation of the full DOL language.[12] DOL-OWL allows for writing OWL ontologies using Manchester syntax [21] (hence they can also be imported from common tools like Protégé), and DOL-OWL provides *interpretations* in the style of OBJ views that relate logical theories (here: OWL ontologies), using interpretations of theories. Interpretations are also used to build up the blending diagrams. Moreover, HETS can compute colimits of such diagrams, as well as approximations of co-limits in the case where the input ontologies live in different ontology languages [7]. These features are essential for the implementation of the example discussed next.

5 Example: Blending Forests and Signs

We briefly describe the theories of signs, forests, and their blends informally, followed by a sketch of the formal specifications of the involved ontologies and their blending.

5.1 An Informal Theory of Forests and Signs

Signs are defined as "(for information / warning) a piece of paper, wood or metal that has writing or a picture on it that gives you information, instructions, a warning, etc.: a road / traffic sign; a shop / pub sign" (taken from Oxford Advanced Learner's Dictionary). In the signage theory, signs are physical artefacts, which are defined by their colour, shape, and location, and they depict a small amount of symbols, i.e., the number of symbols on a sign may not exceed seven

[12] HETS is available under `www.dfki.de/cps/hets`. For more information on DOL and the ISO standardisation effort OntoIOp visit `http://ontolog.cim3.net/cgi-bin/wiki.pl?OntoIOp`

Figure 3. Examples for Sign (top-left), Forest (bottom-left), ForestSign (top-right), and SignForest (bottom-right) [taken from various sources]

items (which is an estimated amount of items). These symbols convey information, which may point to other objects. But also shape or colour can convey information. Signs can in principle be classified into different types of signs, such as road sign or warning sign. Forests are defined as "complex ecological systems in which trees are the dominant life form" (taken from Encyclopaedia Britannica). In the forest theory, forests are natural groups of 'soil, plant, and animal life' with a high density of trees. Here, forests have to contain at least 100 trees (which is an estimated count for simplicity). They can again be classified into subtypes, such as rainforest or tropical forest.

Blending the theories of signs and forests can result in diverse new theories. A blend *forest sign* can, for instance, describe (a) a sign pointing to a forest (by tree icons or the name of the forest), (b) a sign with the shape of a tree, or (c) a sign located in a forest. A blend *sign forest* can, for instance, (a) describe road sign clutter (a 'sign forest'), (b) describe a sign forest that consists of forest signs, or (c) identify the Sign Post Forest (see `http://www.signpostforest.com`). Fig. 3 shows examples of a sign and a forest together with the blends forest sign and 'sign forest' (road sign clutter).

Different blends are mostly based on different base ontologies. The base ontology can specify basic aspects on which the input ontologies for forests and signs agree. For instance, a base ontology can define a category (container) that consists of many entities of the same kind that are essential to determine the category's type. In detail, a sign consists of symbols that determine the sign's type while the forest consists of trees that determine the forest's type. Alternatively, a base ontology can specify that you can get lost in a certain environment. In detail, you can get physically lost in forests, i.e., you do not find your way out, and you can get mentally lost in signs, i.e., you do not see the information conveyed. Furthermore, a base ontology may specify constraints on both input ontologies, such as every forest has more trees than signs have symbols and, consequently, it is not allowed to blend forest to sign and tree to symbol in the same blendoid. Again, the base ontology specification may be guided by foundational ontologies, as described above.

5.2 Ontologies of Forest, Signage and SignForest in DOL-OWL

The two input ontologies in Fig. 4 show parts (modules) of the specifications of the Signage and Forest theory.[13] They formalise signs and forests as described in the previous section. Arrows indicate relationships between classes (i.e., the axiomatisation of the ontologies), thick lines indicate class mappings given by the theory morphisms between the base ontology, the input ontologies and the blend, light grey classes and relations are conservative extensions, which are relevant for the calculation of the colimit. The essential information in the base ontology that can lead to the signforest blendoid specifies a container class that contains objects that have a certain location. From here, partial theory morphisms are defined as interpretations in DOL-OWL that relate classes from the base ontology to classes from the input ontology (along the thick lines, cf. Fig. 4), resulting in the base diagram. Those parts of the base ontology that are not related to parts of the input ontologies are hidden by these partial theory morphisms. However, in order to calculate the colimit that creates the signforest blendoid, these hidden parts are revealed by conservatively extending the input ontologies and making the theory morphisms total, as indicated in Section 3. For example, the morphism from the base ontology to the forest ontology hides the relation hasLocation, which is not specified in the original forest ontology, but the relation then gets related to growsOn in the conservatively extended forest ontology.

Based on the interpretations from Signage and Forest, the input ontologies are blended into the blendoid SignForest by calculating the colimit of the two input ontologies resulting in a tame blendoid. In detail, Forest is identified as Forest in the blendoid. It contains the class Sign, which is mapped to Tree. The typecast of this mapping leads to a 'treeification' of signs, similar to the 'personification' of boats as inhabitants of boathouses. According to the base ontology, these 'treeified' signs have a location (hasLocation) at a certain abstract PhysicalSupport. Note that the blendoid specifies sign forests to contain at least 100 signs, whilst its conceptualisation allows a smaller amount, i.e., the resulting blendoid should be further refined.

6 Discussion and Future Work

Our work in this paper follows a research line in which blending processes are primarily controlled through mappings and their properties [14, 12, 40, 35]. By introducing blending techniques to ontology languages, we have provided a new method which allows to combine two thematically different ontologies in order to re-use axioms in other ontologies and to create a new ontology, the blendoid, describing a newly created domain. The blendoid creatively mixes information from both input ontologies on the basis of structural commonalities of the inputs and combines their axiomatisation.

Ontological blending can serve as an exploratory tool for semantic information retrieval systems (e.g., in medicine) [4]; here, ontological blending will provide the capability to automatically create blend-ontologies from multiple input ontologies that each reflect a certain domain of interest and expertise, e.g., doctors, pharmacists, nurses, each having a different *perspective* on treatment procedures and available information, but with certain shared conceptualisations. Similarly, blending serves to fulfill a creative function within design systems where multi-perspective semantics and reasoning about design concepts is essential [3].

[13] The DOL-OWL specifications is available at: `www.informatik.uni-bremen.de/~okutz/blending/blending.html`

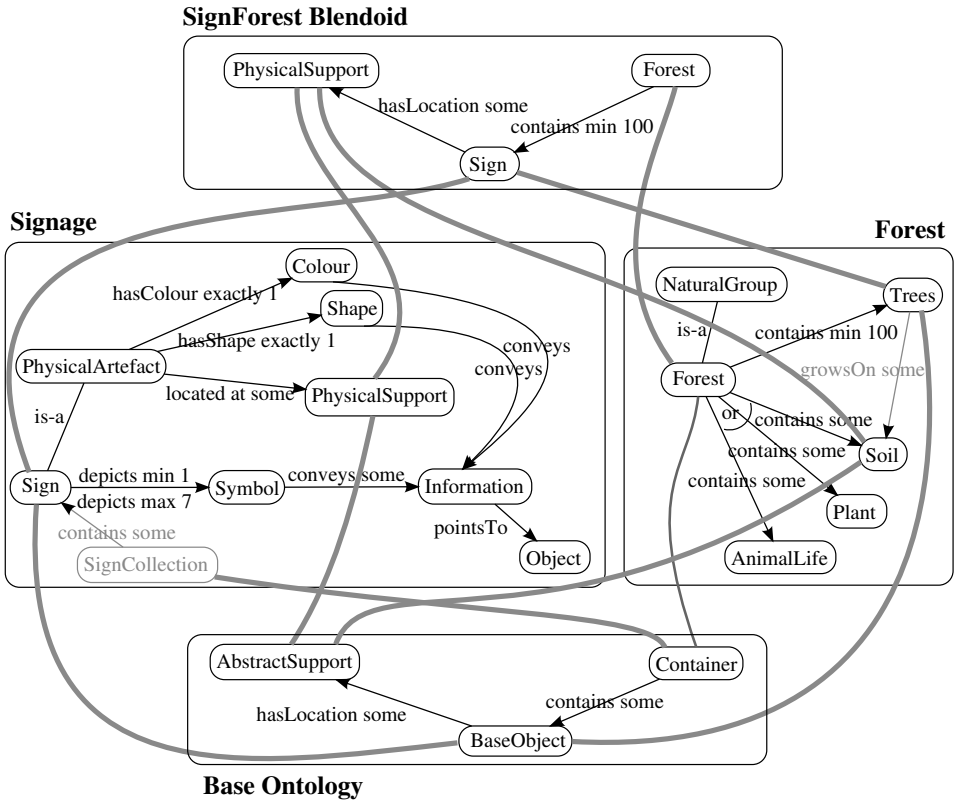


Figure 4. Blending Forest and Signage resulting in the SignForest blend

We have illustrated that the tool HETS and the DOL language [32] (here the DOL-OWL fragment discussed above) provide an excellent starting point for developing the algorithmic side of the theory further. They: (1) support various ontology language and their heterogeneous integration [27]; (2) allow to specify theory interpretations and other morphisms between ontologies [28]; (3) support the computation of colimits as well as the approximation of colimits in the heterogeneous case [7]; (4) provide (first) solutions for automatically computing a base ontology through ontology intersection [29].

However, to make ontological blending feasible in practice, all of these aspects need to be further refined, as discussed above. This concerns primarily the ontological optimality principles (e.g., for semantic completeness and related optimisation heuristics [5]) as well as means for computing common base ontologies [2]. Both issues are almost completely new research questions in ontology research, and we here gave a first analysis and partial answers to them.

Acknowledgements

Work on this paper has been supported by the DFG-funded collaborative research center SFB/TR 8 'Spatial Cognition'.

REFERENCES

[1] J. Adámek, H. Herrlich, and G. Strecker, *Abstract and Concrete Categories*, Wiley, New York, 1990.
[2] Mehul Bhatt, Andrew Flahive, Carlo Wouters, J. Wenny Rahayu, and David Taniar, 'Move: A distributed framework for materialized ontology view extraction', *Algorithmica*, **45**(3), 457–481, (2006).
[3] Mehul Bhatt, Joana Hois, and Oliver Kutz, 'Ontological Modelling of Form and Function in Architectural Design', *Applied Ontology Journal. IOS Press*, 1–35, (2012). (in press).
[4] Mehul Bhatt, J. Wenny Rahayu, Sury Prakash Soni, and Carlo Wouters, 'Ontology driven semantic profiling and retrieval in medical information systems', *J. Web Sem.*, **7**(4), 317–331, (2009).
[5] Mehul Bhatt, Carlo Wouters, Andrew Flahive, J. Wenny Rahayu, and David Taniar, 'Semantic completeness in sub-ontology extraction using distributed methods', in *ICCSA (3)*, pp. 508–517, (2004).
[6] P. Bouquet, M. Ehrig, J. Euzenat, E. Franconi, P. Hitzler, M. Krötzsch, S. Tessaris, D. Fensel, and A. Leger. Specification of a common framework for characterizing alignment, 2005. Knowledge Web Deliverable D2.2.1.
[7] M. Codescu and T. Mossakowski, 'Heterogeneous colimits', in *Proc. of MoVaH-08*, (2008).
[8] S. Coulson, *Semantic Leaps: Frame-Shifting and Conceptual Blending in Meaning Construction*, Cambridge University Press, 2001.
[9] J. Euzenat, 'Semantic precision and recall for ontology alignment evaluation', in *Proc. of IJCAI 2007*, ed., M. M. Veloso, pp. 348–353, (2007).
[10] J. Euzenat and P. Shvaiko, *Ontology Matching*, Springer, 2007.
[11] G. Fauconnier and M. Turner, *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*, Basic Books, 2003.
[12] K. Forbus, B. Falkenhainer, and D. Gentner, 'The structure-mapping engine', *Artificial Intelligence*, **41**, 1–63, (1989).
[13] P. Gärdenfors, *Conceptual Spaces - The Geometry of Thought*, Bradford Books, MIT Press, 2000.
[14] D. Gentner, 'Structure mapping: A theoretical framework for analogy', *Cognitive Science*, **7**(2), 155–170, (1983).
[15] J. A. Goguen, 'An Introduction to Algebraic Semiotics, with Applications to User Interface Design', in *Computation for Metaphors, Analogy and Agents*, number 1562 in LNCS, 242–291, Springer, (1999).
[16] J. A. Goguen and D. F. Harrell, 'Style: A Computational and Conceptual Blending-Based Approach', in *The Structure of Style: Algorithmic Approaches to Understanding Manner and Meaning*, Springer, (2009).
[17] J. A. Goguen and G. Malcolm, *Algebraic Semantics of Imperative Pro-*

grams, MIT, 1996.
[18] N. Guarino and C. Welty, 'Evaluating ontological decisions with OntoClean', *Commun. ACM*, **45**(2), 61–65, (2002).
[19] W. R. van Hage, *Evaluating Ontology-Alignment Techniques*, Ph.D. dissertation, Vrije Universiteit Amsterdam, 2008.
[20] J. Hois, O. Kutz, T. Mossakowski, and J. Bateman, 'Towards Ontological Blending', in *Proc. of the The 14th International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA-2010)*, Varna, Bulgaria, September 8th–10th, (2010).
[21] M. Horridge and P. F. Patel-Schneider, 'Manchester Syntax for OWL 1.1', *OWLED-08*, (2008).
[22] I. Horrocks, O. Kutz, and U. Sattler, 'The Even More Irresistible $\mathcal{SROIQ}$', in *Proc. of KR*, eds., Patrick Doherty, John Mylopoulos, and Christopher A. Welty, pp. 57–67. AAAI Press, (2006).
[23] K. M. Jaszczolt, 'On Translating 'What Is Said': *Tertium Comparationis* in Contrastive Semantics and Pragmatics', in *Meaning Through Language Contrast Vol. 2*, 441–462, J. Benjamins, (2003).
[24] Y. Kalfoglou and M. Schorlemmer, 'Ontology mapping: the state of the art', *The Knowledge Engineering Review*, **18**(1), 1–31, (2003).
[25] W. Kuhn, 'Modeling the Semantics of Geographic Categories through Conceptual Integration', in *Proc. of GIScience 2002*, pp. 108–118. Springer, (2002).
[26] O. Kutz, D. Lücke, and T. Mossakowski, 'Heterogeneously Structured Ontologies—Integration, Connection, and Refinement', in *Proc. KROW 2008*, volume 90 of *CRPIT*, pp. 41–50. ACS, (2008).
[27] O. Kutz, D. Lücke, T. Mossakowski, and I. Normann, 'The OWL in the CASL—Designing Ontologies Across Logics', in *Proc. of OWLED-08*, volume 432. CEUR, (2008).
[28] O. Kutz, T. Mossakowski, and D. Lücke, 'Carnap, Goguen, and the Hyperontologies: Logical Pluralism and Heterogeneous Structuring in Ontology Design', *Logica Universalis*, **4**(2), 255–333, (2010). Special Issue on 'Is Logic Universal?'.
[29] O. Kutz and I. Normann, 'Context Discovery via Theory Interpretation', in *Workshop on Automated Reasoning about Context and Ontology Evolution, ARCOE-09 (IJCAI-09)*, (2009).
[30] M. Martinez, T. R. Besold, A. Abdel-Fattah, K.-U. Kühnberger, H. Gust, M. Schmidt, and U. Krumnack, 'Towards a Domain-Independent Computational Framework for Theory Blending', in *Proc. of the AAAI Fall 2011 Symposium on Advances in Cognitive Systems*, (2011).
[31] T. Mossakowski, C. Maeder, and K. Lüttich, 'The Heterogeneous Tool Set', in *TACAS*, volume 4424 of *LNCS*, pp. 519–522. Springer, (2007).
[32] Till Mossakowski, Christoph Lange, and Oliver Kutz, 'Three Semantics for the Core of the Distributed Ontology Language', in *7th International Conference on Formal Ontology in Information Systems (FOIS)*, ed., Michael Grüninger, Frontiers in Artificial Intelligence and Applications. IOS Press, (2012).
[33] I. Normann, *Automated Theory Interpretation*, Ph.D. dissertation, Jacobs University Bremen, 2009.
[34] R. E. Núñez, 'Creating mathematical infinities: Metaphor, blending, and the beauty of transfinite cardinals', *Journal of Pragmatics*, **37**, 1717–1741, (2005).
[35] F. C. Pereira, *Creativity and Artificial Intelligence: A Conceptual Blending Approach*, volume 4 of *Applications of Cognitive Linguistics (ACL)*, Mouton de Gruyter, Berlin, December 2007.
[36] G. D. Plotkin, 'A note on inductive generalization', *Machine Intelligence*, **5**, 153–163, (1970).
[37] D. Ritze, C. Meilicke, O. Šváb Zamazal, and H. Stuckenschmidt, 'A Pattern-based Ontology Matching Approach for Detecting Complex Correspondences', in *OM-09*, volume 551 of *CEUR*, (2009).
[38] A. Schwering, U. Krumnack, K.-U. Kühnberger, and H. Gust, 'Syntactic Principles of Heuristic-Driven Theory Projection', *Cognitive Systems Research*, **10**(3), 251–269, (2009).
[39] M. Turner, 'The Way We Imagine', in *Imaginative Minds - Proc. of the British Academy*, ed., Ilona Roth, 213–236, OUP, Oxford, (2007).
[40] T. Veale, 'Creativity as pastiche: A computational treatment of metaphoric blends, with special reference to cinematic "borrowing"', in *Proc. of Mind II: Computational Models of Creative Cognition*, (1997).
[41] A. Zimmermann, M. Krötzsch, J. Euzenat, and P. Hitzler, 'Formalizing Ontology Alignment and its Operations with Category Theory', in *Proc. of FOIS-06*, pp. 277–288, (2006).

Web-based Mathematical Problem-Solving with Codelets

Petros S. Stefaneas [1] and **Ioannis M. Vandoulakis** [2] and **Harry Foundalis** [3] and **Maricarmen Martínez** [4]

Abstract. We discuss the question of collective creative thinking conducted in Web-based proof-events in terms of notions from cognitive science, notably the notions of codelets and architecture of mind.

1 INTRODUCTION

Theorem proving is only one of possibly thousands of different cognitive activities with which a mind can be engaged. Minds most probably do not invent new architectural principles to treat each cognitive domain in a special way, because the architecture of the underlying hardware (the brain) is fixed, honed by millions of years of evolution. It has been hypothesized that just as brains are architecturally fixed, so are minds that arise as emergent properties of brains ([9], [10]). That is, there is an "architecture of mind" which is as fixed and unchanging as the architecture of brain. When a mind confronts a cognitive problem it uses "tools" from a fixed repertoire, which however are flexible enough to adapt themselves to and be useful in the solution of any problem. One such set of architectural tools of minds are the *codelets* ([9], [2]).

The purpose of this paper is to examine the feasibility of using the idea of codelets as people who actively participate in seeking and discovering proofs of theorems. To this end, after clarifying the notion of codelet, we look at some software-assisted projects for collaborative Web-based mathematical problem solving. Then we discuss why, in our view, Goguen's [4] understanding of proofs as events, enriched with the notion of codelets, provides an adequate framework for analyzing this sort of Web-based collaborative activity. To the best of our knowledge, the idea of applying notions from cognitive architectures to Web-based collaboration has not yet been explored.

2 PROBLEM SOLVING WITH CODELETS

Codelets can be conceived of as short pieces of programmed code, but in an abstract sense. In brains, codelets can be *ultimately* implemented by means of neurons; in computers, they can be short pieces of programming instructions.

The purpose of codelets is to build conceptual structures in working memory, given some input. Sometimes they can demolish structural pieces, or even whole structures. However, the bulk of their work is constructive rather than destructive. Codelets work in parallel, ignoring each other's existence. Each one has a specific and simple task to complete, and is allocated a given amount of time. If a codelet fails to finish its work within a reasonable time, it "dies" and another codelet of very similar nature makes a "fresh start", working on the same task anew.

Sometimes a codelet may spawn[5] a number of other, different codelets that are deemed useful by it in working on various aspects of the task. Thus the generator codelet becomes a higher-level "supervisor" of the sub-codelets that it generated, waiting for them to finish their sub-tasks in order to continue with its "main" task. This generates a hierarchy of codelets, in which those at a certain level have "knowledge" of the codelets they generated, but ignore both their "superior" codelets and their "peers".

Some differences between the way that codelets work and more traditional programming are the following: (1) the structures built by codelets can be both dispensable and redundant, whereas programs usually have non-redundant code, and whatever they build is never destroyed; (2) codelets that are "peers" (i.e., at the same level in the hierarchy) work in parallel, whereas programs are usually written to run sequentially; and (3) there is no "supervisor" with a total knowledge of which codelets run at any moment and what they will eventually achieve, except at a very local level, when a higher-level codelet becomes a supervisor, but only of the codelets of the immediately lower level that it spawns; instead, in traditional programs the programmer has an overall view and knowledge of which pieces of code exist and can run at any time. Thus, the system of codelets is dynamic and distributed, bound only by the constraints of its hierarchical structure.[6]

An example might clarify the question of how codelets build structures, solving problems at the same time. Let us consider the problem of visually perceiving a written piece of a sentence, and attempting to understand its meaning. Suppose the phrase (i.e., fragment of a sentence) is:

"meaning that he failed to discover it."

We can imagine a highest-level codelet the task of which is: "to understand the given phrase". This codelet spawns a number of other codelets, some of which have tasks as: "to understand one given word"; others: "to put words together in a syntactically correct structure"; and so on. Codelets of the former kind spawn other codelets assigned the task "to read a word". If the perceiving agent is a human being, then the task "to read a word" entails signaling the muscles that move the eyes to perform eye saccades and sample a few spots within the word; whereas if the perceiving agent is a program it could do something analogous but by processing letters within a string, or by processing the pixels of an image, if the above phrase was part of one. Occasionally, some

[1] National Technical University of Athens, Greece, email: petros@math.ntua.gr
[2] Hellenic Open University, Greece, email: i.vandoulakis@gmail.com
[3] Center for Research on Concepts and Cognition, Indiana University, USA, email: hfoundal@yahoo.com
[4] Universidad de los Andes, Colombia, email: m.martinez@uniandes.edu.co

[5] But note that in the system we propose in §5 the role of codelets is played by *people,* so we don't use the idea of codelets spawning other codelets.
[6] Hofstadter [9] does not assign a hierarchical structure to codelets; the idea of a hierarchy of codelets is introduced in the present article.

codelets might produce a wrong result. For instance, the "e" of "failed" might be seen as a "c", but this will not make sense in the context of the output of other codelets because there is no word like "failcd"; thus, another codelet can re-perceive the letter in a way that makes sense. The destruction of already-built structures can be seen more explicitly at the level of syntax: the reader might perceive the word "meaning" as a noun, interpreting the phrase in this sense: "it was that kind of meaning which he failed to…"; but, alas, after the word "discover" comes the pronoun "it", which either is redundant or — if this is a correctly written fragment of a sentence — destroys the perceived syntactic structure and necessitates a re-reading of the phrase. Indeed, with a fresh set of codelets working from scratch, the word "meaning" can be seen as a participle, in which case the pronoun "it" cannot refer to "meaning" but to something else prior to the given phrase. (For example, the phrase could be part of this sentence: "He claimed that the islet did not exist, actually meaning that he failed to discover it.")

Now take the case of collaborative theorem proving. A person engaged in solving a particular task toward the completion of a proof can be thought of as a codelet. The task could be proposed by a supervising codelet (some person, but with a somewhat wider view of the project), and could be taken by a pool of people who have volunteered their services and availability to the proving project, as long as they feel that the task is suitable for their abilities. Similarly, a person working on a codelet could assign sub-tasks as other codelets, of simpler nature and of an ever-narrower view, which can be taken by less qualified or less specialized proving agents. At the highest level could stand a person of qualified knowledge who gave the initial broad strokes, i.e., decided the highest-level tasks and placed them in the "codelet pool" to be undertaken by qualified agents. The tacit assumption is that perhaps in this way proofs of greater complexity can be achieved than is possible by the faculties of a single person-prover. In the rest of this paper we shall examine this idea more thoroughly.

3 WEB-BASED MATHEMATICAL PROBLEM-SOLVING

The Web may transform the way we understand mathematical proving activity. It has been used to make proving a collaborative activity, involving different people who have different backgrounds, research interests, viewpoints and expertise. This was attempted in different ways and approaches.

The Kumo proof assistant and the Tatami project was a fist such attempt that was undertaken by J.A. Goguen. The Tatami project is a Web-based distributed cooperative software system that comprises a proof assistant, called the Kumo system, a generator for documentation websites, a database, an equational proof engine, and a communication protocol to maintain truth of distributed cooperative proofs. For each proof Kumo generates a proof website (proofweb) based on user-provided sketches in a language called Duck, and assists with proofs in first-order hidden logic [5]. The understanding of mathematical proof is facilitated by the Tatami project, because it displays them as representations of their "underlying mathematics" [3].

Another approach was used in the Polymath and the Tricki Projects, initiated by Timothy Gowers in 2009[7]. He posed a mathematical problem in his blog, namely a special case of the density Hales-Jewett theorem [8], and invited the mathematical community to collaborate openly in finding an alternative, "better" proof, that could enable a deeper understanding of the theorem. The participants had the opportunity to use a rather poor arsenal of Web-tools, namely the comment function of Gowers' blog, to suggest ideas, methods and parts of proof. Alongside with this initiative, Gowers, in cooperation with Olof Sisask and Alex Frolkin, launched a Wikipedia-style project of creating a large repository of articles about the mathematical techniques that could be useful for various classes of mathematical problem-solving ([6], [15]). Thus, the Tricki project was conceived as a "treasury" of higher-order mathematical thinking, designed to support the mathematical proving practice.

In both Kumo and Polymath projects, parts of a proof can be exchanged among the members of a group and Web communication becomes an essential part of the proving activity. In addition, the Polymath project heavily relies on techniques very close to brainstorming and crowdsourcing [14]. Thus, Web-based mathematical problem-solving is strengthened by collective creative thinking, whereas the Tatami and Tricki projects serve the building up of a collective memory on (both successful and unsuccessful) proving practices.

4 PROOF-EVENTS AS A FRAMEWORK FOR WEB-BASED MATHEMATICAL PROBLEM SOLVING

Goguen [4] introduced the concept of *proof-event* in an attempt to formulate a wider viewpoint on proof, designed to incorporate traditional mathematical proofs (both constructive and non-constructive proofs), but also non-mathematical proofs (apodictic, dialectical, ontological, etc.) as well as new kinds of proving practice, such as computer proofs and proof steps.

Accordingly, the proving activity is a social process occurring in space and time and involving appropriate groups of experts, consisting of at least two persons: a *prover* (which may be a human or a machine) and an *interpreter* (who can be only a human or the mathematical community). These agents may be separated in space and time, and share possibly different *codes of communication*. Consequently, a proof-event (or generally, a sequence of proof-events) may have many different outcomes. A proof is completed when the persons involved in a proof-event conclude that they have *understood* the outcome and agree that a proof is actually given.[8]

The conceptual framework developed ([17], [13]) on the grounds of Goguen's definition of proof-event [4] enables us to approach the concept of mathematical proving activity not as an individual venture of a (possibly socially isolated) mathematician, but as an activity dependent on social and cultural underpinnings, as well as on particular groups of the academic community and their intellectual capacities.

This framework proves adequate to describe the novel form of Web-based proving as is practiced in the Kumo and Polymath projects [14]. Web-based proving appears to be a novel kind of proving practice, characterized by a change of the communication

[7] Details on the Polymath project can be found in [7].

[8] A detailed analysis of the components of proof-events is given in [16] and [17].

medium: the Web serves as both an information source (a repository of information, ideas and methods available) and a communication medium (creating global interest-based communities). Mathematical problem-solving is open to all, and communication is transformed from one-to-one or one-to-many into many-to-many.

Interactivity in Web-based proving, as practiced, for instance, in the Polymath project, enables the use of a group problem-solving technique known as *brainstorming* [12]; in particular, (asynchronous) *computer-mediated* or *Web-based (group) brainstorming* [1], by which a group tries to find a proof for a posed mathematical problem by culling a list of spontaneously generated ideas contributed by its members.

Therefore, the concept of proof-event can adequately describe such innovative forms of problem-centered proving practices and serve as a general framework of interpretation for such experiments.

5 A CODELETS-BASED MODEL FOR THE WEB-BASED PROOF-EVENTS

Web-based mathematical problem-solving is a process based primarily on the prover-interpreter interaction over the Web. During this process, an initial interpreter inserts into a *pool of unresolved issues* a list of issues that, if resolved, amount to the solution of the initial problem. For example, if the problem to be solved is the proof of a theorem, then the list of unresolved issues that are inserted into the pool can be the highest-level pieces of the proof, as envisioned by the initial interpreter. The pool is immediately available to the Web-based community of participants, who are informed of its existence by mediating "system software" and, acting as *codelets,* select pieces that they deem solvable. (Henceforth, for simplicity, the participants in the Web-based problem-solving event will be referred to as "codelets".) When codelets select an unresolved issue they do so either because they feel capable of solving it by their own means, or because they see how to decompose it further into constituent sub-issues, which are then also entered into the pool. The "system software" (henceforth: "system") keeps track of which issues are parts of which larger ones; i.e., the system knows the *hierarchy* of the problem decomposition.

As soon as a prover-codelet feels that the solution of the issue that the codelet was working on is available, informs the system of this fact. The system informs the supervisor codelet (who had inserted that issue into the pool), and the latter acts as an interpreter of the solution. If the solution is validated by the interpreter-codelet, the system is informed, so that other codelets do not attempt to assign to themselves the solution of the same issue, which appears as "solved" in the pool. When the interpreter-codelet finds that all the sub-issues of the undertaken issue are solved informs the system, and so on. Thus, each codelet acts as both a prover (for the single issue that the codelet selected from the pool), and an interpreter (of all the sub-issues that the codelet entered into the pool, after decomposing the selected issue).

In addition to parts of a proof (sub-proofs), codelets may make various other contributions, such as incomplete or even false proofs, ideas, comments, suggestions, opinions and methodology transfer rooted in past experience and expertise. These contributions are also entered into the pool, each distinguished by its type, and are conceived as directed toward the solution of a stated problem. Hence, the contributions are independent, goal-directed processes that evolve over the Web space and time and accumulate as building blocks or modules of a generated Web proof-event.

Particular contributions may turn out to be blind, i.e. to lead in due time to a recognizable deadlock situation. This may entail a change of approach towards the problem, change of methodology applied, etc.; that is, it may give rise to a new contribution and the abandonment of the unfruitful undertaking. After all, as explained in the introductory sections, some codelets might act destructively, invalidating particular sub-proofs and contributions (e.g., when they find an error in a sub-proof, or that an idea is unfruitful, etc). However, such destructions are local, whereas overall the system proceeds to a coherent solution, if one can be found.

In Web-based proof events codelets have certain specific features:

i. Each codelet, acting as a prover, knows neither who its supervising interpreter-codelet is, nor its "peer" codelets who might be working on other sub-issues of the same problem. However, when that prover-codelet becomes a supervising interpreter-codelet (due to having decomposed the issue and entered its parts into the pool), then it can keep track of which codelets work on which sub-issues. This information becomes available to it by the system.

ii. When codelets see that an issue is marked as "being worked on" (or "taken" by a codelet), they are not prevented from taking it as well. This is because some codelets may feel they can give a neat solution that might be missed by other codelets. If, eventually, two or more solutions arise for a given issue, it is up to the supervising interpreter-codelet to choose one and inform the system, which incorporates it into the overall solution.

iii. As already mentioned, the work of some codelets may turn out to be superfluous or even useless. The outputs of such codelets are not ultimately integrated into the final structure of the formal mathematical proof. Nevertheless, they cannot be considered totally irrelevant, because they might have revealed unexpected relationships with other mathematical concepts or statements or elucidate the independence of some assumption of the mathematical statement to be proved or uncover the need of a weaker or refined assumption.

iv. Particular codelets, or their derivative contributions, may vary in location and weight in a process of generation of a Web-based proof-event. A codelet may turn out to be prerequisite, refinement, simple correction or even counterexample for the contribution of another codelet. Therefore, they are arranged neither in parallel, nor in sequential order. They have a complex, graph-like structure that follows the eventual formal structure of the provisional mathematical proof.

v. Administrators do not know in advance the final outcomes of Web-based proof events, so they can't provide deterministic guidance. They are trusted by the community of the contributors in view of their reputation in the academic world. At the final stage of Web-based proof events administrators can potentially intervene, evaluate, correct, filter and integrate all kinds of contributions.

6 COLLECTIVE INTELLIGENCE AS EMERGING PROPERTY OF CODELETS IN WEB-BASED PROOF-EVENTS

Collective creative thinking and collective memory are essential components of the Web-based mathematical problem-solving. The Kumo assistant and the Polymath project are Web tools facilitating the collaboration of codelets, whereas the Tatami and the Tricki projects serve as repositories of the acquired collective memory. The image of an individual mathematical mind, which is intelligent enough to cope with hard mathematical problems, is replaced in Web-based problem-solving by the image of a "collective" mathematical mind, which is more efficient to handle difficult problems in shorter time. The new picture is vividly outlined by Nielsen [11], as an epoch-making type of "networked science".

Collective intelligence in Web-based problem-solving is characterized by openness, i.e., unrestricted sharing of ideas and intellectual property among codelets, peering of codelets and joint goal-directed action. Thus, collective intelligence can be understood as an emergent distributive property over numerous codelets of a "collective mind" that uses a set of flexible and adaptable tools from a Web-based repository in facing mathematical problems.

Such tools have a double nature: on the one hand they are objects readily available to be used for any specific purpose, i.e., they are objects "ready-to-hand" (to use Heidegger's terminology), just lying there; on the other hand, when these tools are activated (for instance, when Mathematica is used) they may initiate processes and produce contributions that even a prover might fail to reach, although they lack the intelligence of a prover. From the latter standpoint, they act as (intelligence-less) codelets, insofar as they actively work on data and follow the architecture of the Web.

7 CONCLUSION

A system for Web-based cooperation among people for the handling of proof events and mathematical problem-solving was proposed in this paper. The main advantage of this approach over the more traditional proving method is the interesting possibility that mathematical problems that are far too complex to be solved by a single person might become solvable by a community of mathematicians who cooperate following the system outlined in the present text. It is our firm belief that the limits of group thinking and cooperation among members of a community lie far beyond those of individuals, and that such limits need to be further explored.

REFERENCES

[1] A.R. Dennis and J.S. Valacich, 'Computer brainstorms: more heads are better than one', *Journal of Applied Psychology*, **78**(4): 531–537 (1993).

[2] H.E. Foundalis, *Phaeaco: a Cognitive Architecture Inspired by Bongard's Problems,* Ph.D. dissertation, Computer Science and Cognitive Science Departments, Indiana University, Bloomington Indiana, 2006.

[3] J. Goguen, K. Lin, G. Rosu, A. Mori, and B. Warinschi, 'An overview of the Tatami project', in *Cafe: An Industrial-Strength Algebraic Formal Method,* K. Futatsugi, T. Tamai and A. Nakagawa (eds), 61–78, Elsevier, (2000).

[4] J.A. Goguen. *What is a Proof?* Informal essay. University of California at San Diego. Accessed May 30, 2012 from: http://cseweb.ucsd.edu/~goguen/papers/proof.html.

[5] J.A. Goguen, 'Social and semiotic analyses for theorem prover user interface design', *Formal Aspects of Computing,* **11** 272–301, (1999). (Special Issue on User Interfaces for Theorem Provers).

[6] T. Gowers. *Tricki now fully live*. Accessed April 2, 2012: http://gowers.wordpress.com/2009/04/16/tricki-now-fully-live.

[7] T. Gowers and M. Nielsen, 'Massively collaborative mathematics', *Nature* **461**, 879–881 (October 15 2009).

[8] A. Hales and R. Jewett. 'Regularity and positional games', *Trans. Amer. Math. Soc*. **106**, 222–229, (1963).

[9] D.R. Hofstadter, *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought,* Basic Books, New York, 1995.

[10] M.L. Minsky, *The Society of Mind,* Simon and Schuster, New York, 1988.

[11] M. Nielsen, *Reinventing Discovery: The New Era of Networked Science,* Princeton University Press, Princeton, 2011. (e-Book).

[12] A. Osborn, *Applied Imagination: Principles and Procedures of Creative Problem Solving,* Charles Scribner's Sons, New York, 3rd edn, 1963.

[13] P. Stefaneas and I.M. Vandoulakis, 'Proofs as spatio-temporal processes', in *Abstracts of the 14th Congress of Logic, Methodology and Philosophy of Science,* Nancy, July 19–26, 2011, pp. 131–132.

[14] P. Stefaneas and I.M. Vandoulakis, 'The Web as a Tool for Proving' *Metaphilosophy*, (July 2012) (to appear).

[15] T. Tao. *Tricki now live. What's new*. Accessed April 2, 2012: http://terrytao.wordpress.com/2009/04/16/tricki-now-live.

[16] I.M. Vandoulakis and P. Stefaneas, 'Conceptions of proof in mathematics', in *Proceedings of the Moscow Seminar on Philosophy of Mathematics*. Proof. V.A. Bazhanov, A.N. Krichevech, V.A. Shaposhnikov (Eds), Moscow [Summary in Russian] (to appear).

[17] I.M. Vandoulakis and P. Stefaneas, 'A typology of proof-events', *International colloquium on history of mathematical sciences and symposium on nonlinear analysis in the memory of Prof. B.S. Yadav*, May 16–19, 2011, Almora, India, (to appear).

From Alan Turing's Imitation Game to Contemporary Lifestreaming Attempts

Francis Rousseaux, Karim Barkati, Alain Bonardi and Antoine Vincent[1]

Abstract. Among his various productive contributions, Alan Turing has imagined to turn the question "can machines think?" into what he called an *Imitation game,* with specific rules and play conditions ([23], [25]). Reusing the notion of dynamic continuum from *ludus* to *paidia*, as introduced by Roger Caillois in his famous study *Man, Play and Games* ([4]), we claim, with most computer scientists, that the Turing *Imitation game* is strongly *ludus*-tagged, mostly because it is not attractive and playful enough to be spontaneously played: the contrast is strong, compared to some later *paidia*-tagged game involving computing machineries, like *Interactive information and digital content browsing and retrieval.* As far as designing our interactive Artificial Intelligence systems is concerned, why should we have to choose between *ludus* and *paidia* or to deny their eternal competition? On the contrary, this paper proposes to dare to establish that irreducible concurrency between *ludus* and *paidia* as the heart of our future systems, rediscovering the importance of the Greek notion of *kairos.*

1 HAPPY BIRTHDAY DR. TURING!

During this year 2012, we shall celebrate the centenary of Alan Turing's birthday.

Apart from the recurrent scientific manifestations that pay homage or tribute to Alain Turing, such as the yearly Turing Award, 2012 will be marked up by many special events, among them scientific conferences or workshops all around the world, competitions (like the Turing Centenary Research Fellowship and Scholar Competition), and socio-political events like the amazing attempt to grant a pardon to Alan Turing, far exceeding the computer science communities.

The reason why Turing stays so famous among computer scientists not only relies on Turing's unique impact on mathematics, computing, computer science, informatics, morphogenesis, artificial intelligence, philosophy and the wider scientific world. It has something to do with the mystery of his life and the complexity of his various theories, borrowing inspiration to many different fields and crossing them boldly. For example, the present paper authors, as computer scientists involved in digital arts and interactive computer games, regularly mobilize some Turing scientific contributions, as several from their colleagues use to do so ([16], [12]), not only for technical purposes but also for cross-disciplinary connections and attempts to innovate.

This papers aims at coming back on one of the most extraordinary Turing's contributions, namely his *Imitation game,* built up "to replace the question 'can machines think?' by another, supposed to be closely related to it and expressed in relatively unambiguous words" ([23]).

The first section is dedicated to the description of some preliminary considerations about the *Imitation game* and Test, as designed in 1950 by Turing in his famous paper, concentrating on some specific considerations, supported by the sociologist Roger Caillois study about *Man, Play and Games.*

The second section describes a contemporary domain for *Imitation games* application, namely the interactive information browsing and retrieval process, analysed from a Turing Test point of view and perspective. A comparative approach with the general tracks put forward by Turing will allow us to introduce the innovative idea of Collection-centred analysis and design.

The third section will be dedicated to the development of this *Collection-centred analysis and design concept,* aiming at some specific research and applications, among them the contemporary lifestreaming attempts.

2 THE IMITATION GAME

Since the publication in 1950 of his 27 pages long paper in the 59th volume of *Mind* [23], Alan Turing ideas about Computing Machinery and Intelligence has been commented a lot, without any significant lassitude or interruption.

Some authors, usually computer scientists, have put forward some constructive criticism around Computing Machinery coming from their technical experience ([21], [1], [2], [3]), while others, usually philosophers of mind, have put forward some theoretical proposals to reframe or resituate Turing ideas about *intelligence* ([18], [9], [10]).

This section does not pretend at an exhaustive review of those contributions, nor at producing one more contribution to be considered within the permanent flow of it: we only aim at pointing out some particular aspects of Turing ideas about Computing Machinery and Intelligence that will be extended and mobilised within the next section.

2.1 Principle, framework and object

First of all we would like to redraw quickly the principle, framework and purposes of the *Imitation game* (and its Test version), such as described by Turing in his paper.

The notion of *game* relies in the heart of Turing key-concepts from the beginning of his scientific career, as it is central within the cybernetic approach ([7]): in [24], Turing will thus sketch a game typology by distinguishing *game with complete knowledge theory* from *games with incomplete one.*

[1] Ircam, Paris, France, email: {name.surname}@ircam.fr

Notice that *the Imitation game* is managed by an interrogator-oracle: C is that interrogator who tries "to determine which of the other two is the man (A) and which is the woman (B). He knows them by labels X and Y, and at the end of the game he says either 'X is A and Y is B' or 'X is B and Y is A'. The interrogator is allowed to put questions to A and B. […] It is A's object in the game to try and cause C to make the wrong identification. […] The object of the game for the third player (B) is to help the interrogator".

Notice also that there is a Test version of the *Imitation game,* characterised by the omission of B: "the game (with the player B omitted) is frequently used in practice… […]. They will then probably be willing to accept our test".

Then we can sketch this simple matrix that help to keep in mind the main configurations and objects of the Turing proposals:

	A is a Human	***A is a Computing Machinery***
A and B face C	classical game	Turing *Imitation game*
A or B faces C	*viva voce*	Turing Test

Fig.1: matrix representing the actors configurations in the Turing proposals

To go forward, we propose to use the erudite considerations of Roger Caillois in his famous book *Man, Play and Games* ([4]) written in 1967 and translated to English by Meyer Barash in 2001.

2.2 Caillois' classical study

In his study, Caillois defines *play* as a free and voluntary activity that occurs in a pure space, isolated and protected from the rest of life. *Play* is uncertain, since the outcome may not be foreseen, and it is governed by rules that provide a level playing field for all participants. In its most basic form, *play* consists of finding a response to the opponent's action — or to the play situation — that is free within the limits set by the rules.

Caillois qualifies types of *games* — according to whether competition, chance, simulation, or vertigo (being physically out of control) is dominant — and ways of playing, ranging from the unrestricted improvisation characteristic of children's play to the disciplined pursuit of solutions to gratuitously difficult puzzles. Caillois also examines the means by which *games* become part of daily life and ultimately contribute to various cultures their most characteristic customs and institutions. According to Roger Caillois and Meyer Barash, *play* is "an occasion of pure waste: waste of time, energy, ingenuity, skill, and often of money". In spite of this — or because of this — play constitutes an essential element of human social and spiritual development.

Thus is it possible to sketch a second matrix pointing out, for each *game* feature studied by Caillois (pp. 42-43 of the French edition), the main characters of Turing *Imitation game* and Test.

2.3 Caillois applied to Turing *Imitation games*

Games have to be **separate** (within space and time constraints, fixed in advance):

- As far as time is concerned: the response delays of C's interlocutors (A and B) are artificially temporised to prevent easy information towards C;
- As far as space is concerned: the physical placement of A and B is governed in such ways that direct perception is not possible for C, either visual, tactile or acoustic;
- As far as truth is concerned: the Computing Machinery A is able to simulate some mistakes, imitating the famous *Errare humanum est,* just to mask its unlikely aptitude to calculate (thus, the addition 34957+70764=105621, mentioned into the Turing paper, is false). Notice that other kinds of mistake (such as language slips) are not taken into account.

Games have to be **regulated** (submitted to some particular conventions that suspend ordinary laws):

- As Turing paper readers, we know nothing accurate about the dialogue process between C and A and/or B: Who is supposed to be interrogated first by C? Is it compulsory, for C, to alternate rigorously the different tirades? Could C concentrate on one particular protagonist by asking him/her several successive questions?
- How does the dialogue stop (as far as the universal Turing Machine is concerned, we know the importance of the stop conditions)? How to limit the *Deus Ex Machina* effect?

Games have to be **uncertain** (the process cannot be fully predictable, some inventions and initiatives being required by players):

- As far as the nature of questions/responses is concerned: How can the interrogator be convinced enough to decide between « (X is A and Y is B) or (X is B and Y is A) »? There is no precise response to that interrogation;
- Sometimes one single response tirade is enough to inform C, typically in case of practical examination, like some arithmetic instruction execution, or some particular movement of a given chess piece in a given game configuration;
- Unfortunately, this type of question does not prove that a good answer is necessarily due to a deep understanding of the respondent — rather than a lucky choice — nor that a bad response is not a mistake coming from a wrong practical application of a very correct theory;
- That is why it seems also possible for C to describe some different knowledge regions being first mapped, like sonnet writing (about Forth Bridge), arithmetic mastering (add 34957 to 70764) or chess challenging (I have K at my K1, and no other pieces. You have only K at K6 and R at R1. It is your move. What do you play?). The heuristic is there to multiply examination scopes and to diversify the interrogation domains to reduce the evaluation hazards — but this remains a very inductive and empiric method;
- At least, questions looking for a complex answer or a sophisticated demonstration (such as "What do you think of Picasso?" or "Consider the machine specified as follows… Will this machine ever answer 'Yes' to any question?") are forbidden;
- The interrogator can get around by describing a systematic structure built by *a priori* knowledge. This is the literary criticism example, where the interrogator tests the capacity of the (human or machinery?) poet to behave differently from a parrot, by evoking successively rhyme, metaphor and metonymy as creative knowledge about sonnet writing.

Games have to be **unproductive** (playing cannot create any goods or wealth):

- "I believe that in about fifty years' time it will be possible, to programme computers, […], to make them play the *Imitation game* so well that an *average* interrogator will not have more than 70 per cent chance of making the right identification after five minutes of questioning";
- To challenge that prophecy without breaking the rule and aiming at game productivity, Turing prospects towards what

he calls Learning Machine, which, according to him, has to be unpretentious, accepting ignorance, including random, fallibility and heuristic approaches. According to Turing, Computing Machineries have to train their skills, pushing the *Imitation game* towards *ludus* rather than *paidia* ([4], pp. 75-91 of the French edition).

Games have to be **fictitious** (players can easily access to the unreality feature of the game, compared with current life) and **free** (playing is not obligatory):

- In 1950, Turing admitted that Computing Machineries will have to wait for being able to attend an *Imitation game* managed by an educated interrogator, recognizing that the **fictitious** feature of *Imitation games* was too obvious, the real problem being more the lake of **addictive** available feature to be experienced by the players;
- The Turing *Imitation game* is clearly not funny enough: what could really encourage the interrogator to participate? What makes him continue to play the game? How to turn *Imitation games* into real entertainments for real *average* players?

2.4 A socio-technical analysis

Caillois places forms of *play* on a continuum from *ludus*, structured activities with explicit rules *(games),* to *paidia*, unstructured and spontaneous activities (playfulness), « although in human affairs the tendency is always to turn *paidia* into *ludus*, and that established rules are also subject to the pressures of *paidia*. It is this process of rule-forming and re-forming that may be used to account for the apparent instability of cultures ». Thus Paul Valery proposed as a definition of play: "L'ennui peut délier ce que l'entrain avait lié" (boredom can untie what enthusiasm had tied).

In general, the first manifestations of *paidia* have no name and could not have any, precisely because they are not part of any order, distinctive symbolism, or clearly differentiated life that would permit a vocabulary to consecrate their autonomy with a specific term. But as soon as conventions, techniques, and utensils emerge, the first games as such arise with them. At this point the pleasure experienced in solving a problem arbitrarily designed for this purpose also intervenes, so that reaching a solution has no other goal than personal satisfaction for its own sake ([27], [8]).

Turing has tried to form *ludus* rules to turn the *paidia* question "Can machines think?" into an other, "supposed to be closely related to it and expressed in relatively unambiguous words". He built up the *ludus* rules… but failed to turn his *free* and so *fictitious* game into an *addictive* enough one, providing enthusiasm and entertainment to players. Several contributions discuss that question, directly or indirectly ([5], [6], [11], [22], [15], [19], [28]).

We now understand enough the *Imitation game* theory to go forward. If an *Imitation game* can be turned into a Turing Test (with the player B omitted), why not adapt it to some different use cases, like interactive information browsing and retrieval through the Web, using some search engine?

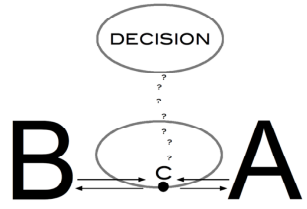


Fig.2: decision making in interactive information browsing and retrieval

The idea is to study this very common contemporary situation, that involves daily thousands of *average* users, and to describe what happens to the Turing key concepts. Turing would have dream to access such a huge panel of various practices!

3 THROUGH THE WEB

In their everyday life, thousands of people stay in front of their computer, mobile phone or tablet, to use some engine for searching information and browsing the Web. They belong to different generations, different countries, different cultures, they have different professions, but they all spend time for that, most of the time spontaneously, if not compulsively.

They enter into more or less long sessions, interacting with the search engine, and suddenly decide to get out of the session, stopping their collaboration with the Computing Machinery, that is supposed to be cooperative.

Most of the time nobody is here to investigate, checking why do the users stop collaborating at this precise moment, asking them if they are satisfied with the Machinery cooperation, elaborating some survey about what they exactly do when they communicate key-words to the searching engine or when they receive URLs lists in return to their queries.

In this section, we should like to elaborate around that phenomenon, asking some help to Turing ideas and intellectual devices, and practising by differential analysis.

3.1 Interactive information browsing

Somebody is looking for something and browses the Web, entering suddenly the interface of a given search engine. He/she put forward some keywords, just to see how the computer machinery would react to his/her provocation. The artificial system is offering back to its user an ordered list of URLs, accompanied by some surface information about the URLs content.

Now the user has a surrounded view; then, accesses to some URL and visits some associated contents; then, browses the URLs collection and chooses at a glance a new one to explore. Like in a museum, faced to an exhibition — *the screen of the machinery* —, he/she visits — *browses* — the piece of art — *the URLs contents* — of a collection. Suddenly the user C is becoming the advanced user C', mode skilled, more concerned about the current session, with more accurate concerns and projects in a better understanding situation: C' is entering some new interaction with the Machinery, C' is now different from C who he/she was. Thanks to that role he/she played when analysing the system reactions/proposals, C' has got news ideas for asking questions to the computer, choosing *better* keywords and descriptors to communicate, knowing *better* what he/she is *really* looking for.

Later on, C'' (and soon C^n) will have so much changed his/her mind that it would not be possible anymore to trace his/her initial project: because of the successive interpretation layers he/she did, but also because of the combinatorial explosion of the interpretation possibilities, mixing intuitions coming from different layers of the whole session, that still keep present to the mind of a human interpreter. The future does not rely only on the present.

At a first glance, the Machinery interrogator (C) seems to be alone in front of it, tending to personify it, like in a special kind of Turing Test (with the second player B omitted) where the player A, which tries to help the interrogator, is the cooperative Machinery

(for readability reasons we prefer to keep the letter A for the Machinery — even if it is cooperative — which normally deserves the letter B).

	Actors in presence	***Similarity with Turing approaches***
I am (C) alone in front of the Machinery A	C, A	Turing Test
I split myself into C and B~C, in front of the Machinery A	C, A, B~C	Imitation game (C'←C observes the dialogue between A and B~C)
I multiply myself in front of the Machinery A ***I evolve by building up:*** (B^n~C^n)←…←(B''~C'')←(B'~C')←(B~C)	C, A, B~C, B'~C', B''~C'',… , B^n~C^n	Vertigo of a simulacrum (The present time does not sum up the past)

Fig.3: actors' configurations within some browsing and retrieval situation

3.2 Vertigo of simulacrum

The similarity with the *Imitation game* only appears when analysing more accurately the situation: we can distinguish a third role, certainly played by the person of the interrogator C, but distinct from his/her strict interrogation role. This third role looks like the cooperative woman B one in the Imitation game, trying to support the interrogator. Let us call B~C this role, to differentiate the roles B from C, but to claim the identity of the common physical player. B~C interprets the tirades exchanged between C and B to help the up-to-date C' / C'←C (C', formerly C) to reformulate the next question of his/her interrogation session.

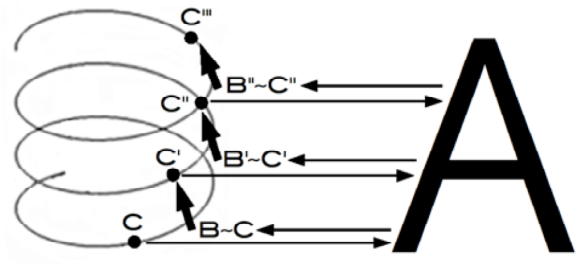


Fig.4: decision making in browsing and retrieval situation

When the Machinery A is the provocative black box, C enjoys splitting to create his/her new role B~C: C is certainly the interrogator, but he/she also learns how to become, tirades after tirades, some (B^n~C^n) such as (B^n~C^n)← …←(B''~C'')←(B'~C')←(B~C) able to better use the Machinery B. Then the dialogue is far more complex than a linear succession of tirades, where the future only depends on present times, the past having being totally absorbed by the present: the process is not at all a Markov chain, the future is a recollection of precedent states collection, not limited to the lonely present. We are faced to ***vertigo of simulacrum***, as pointed out by Caillois (page 92 of [4]).

3.3 Back to Caillois' categories

Games have to be **separate** (within constraints, fixed in advance), **fictitious** (players can easily access to the unreality feature of the game, compared with current life) and **free:**

- Because the user is changing his/her mind during the session, until meeting his/her content *search,* those interactive information browsing and retrieval use-case are not so separate from the current life, players being often unable to access to the unreality feature of the game. Their tend to play spontaneously, the Machinery being always available, and to forget the separations fixed in advance: if it could be dangerous within Serious games, this feature is however required in Virtual realities and Social games approaches;
- With the coming advent of Massive Social Networks and Life Streaming technologies and services ([20]), this tendency will probably be more and more heavy: the **separate** and **fictitious** requirements for artificial games will be more and more difficult to fulfil, addiction becoming a real risk for *average* users.

Games have to be **regulated** (submitted to some particular conventions that suspend ordinary laws):

- Remember that games have to be governed by rules, under conventions that suspend ordinary laws, and for the moment establish new legislation, which alone counts, and have to make-believe, in the meaning that they generate a special awareness of a second reality or of a free unreality, as against real life. These diverse qualities are somehow contradictory;
- As Roger Caillois wrote (turned into English by Meyer Barash): "those qualities do not prejudge the content of games. Also, the fact that the two qualities — rules and make-believe — may be related, shows that the intimate nature of the facts that they seek to define implies, perhaps requires, that the latter in their turn be subdivides. This would attempt to take account not of the qualities that are opposed to reality, but of those that are clustered in groups of games with unique, irreducible characteristics".

Games have to be **uncertain** (the process cannot be fully predictable, some inventions and initiatives being required by players) and **unproductive** (playing cannot create any goods or wealth: only property is exchanged):

- [Turing 50] put a very strong accent on Learning machines, and Turing imagines a role for the experimenters judgment, especially when he writes: "Structure of the child machine = hereditary material, Changes of the child machine = mutation, Natural selection = judgment of the experimenter. One may hope, however, that this process will be more expeditious than evolution. The survival of the fittest is a slow method for measuring advantages. The experimenter, by the exercise of intelligence, should he able to speed it up";
- But the very fact he did not succeed in designing an efficient *ludus* system made his forecast and ambition fail;
- With interactive information browsing and retrieval, back to *paidia,* the experimenter judgments can more easily be involved through machine learning processes, giving life to a real A^n ←…← A''←A'←A sequence. And of course, users being themselves involved into social communities of practice, their cooperation can amplify the machine learning complexity;
- The Learning Machine concept originally put forward by Turing becomes Persons/Machines Learning Systems, where Persons/Machines dialogues can inspire both persons and machines learning.

3.4 Analysis and perspectives

Curiously, it appears that: 1° the original Turing *Imitation game* and Test was a poor *ludus* designed for few users (Joseph Weizenbaum's friends testing ELIZA in some MIT lab in 1965?), whereas 2° the *Interactive information browsing and retrieval* activity is a great spontaneous *paidia* for many different people through the world. The temptation could be to turn back this *paidia* to some improved new-generation *ludus:* but we have learnt from Caillois how vicious is that circle. Trying to regulate the new game and organising some canonical machine learning, the risk is strong to come back to a poor *ludus* game, quickly abandoned by massive user communities.

The solution could be to use tension between *ludus* and *paidia* in the heart of our interactive systems, rather than trying to deny or reduce it. That will be the role plaid by a *Collection-centred analysis and design* concept we shall introduce in the next section of this paper. The basic idea is to consider *seriously* the activities of collecting: we claim that collectors and curators play a central archaic role in the constitution of our current usual knowledge.

4 COLLECTION-CENTRED DESIGN

Collection-centred analysis and design will be presented in this section, as an attempt to inherit from our deepest cognitive social and ancestral behaviours (human beings definitely are collectors, and collections are good places for welcoming the eternal *ludus* and *paidia* competition in the centre of our practices) towards modern ways of thinking and building our future kairos-centred AI systems, which could perfectly be characterized by recent lifestreaming attempts.

Here it is important to distinguish between figural and non-figural collections. This subtle distinction, introduced in the 1970s by Piaget and his research teams of child psychologists, brings more light to the situation. On the one hand it is certain that *non-figural* collections exist because they are completely independent of their spatial configuration. In that, they are already close to classification, of which they can only envy the formal completeness. On the other hand, there are collections we can label as *figural* because both their arrangement in space and the private properties of the collected objects determine their meaning.

4.1 Figural *vs.* non-figural collections

Because our collections seem to be nearer to order than disorder, attempting to assimilate them in classes according to predefined schemes, as in *ludus* approaches, is not so surprising: the necessary elicitation of implicit knowledge that requires class building has to do with the necessary evolution of games from *paidia* to *ludus*. At least, collections look like they are waiting for their completion within a classification order, with the aim of turning into canonical achieved structures made of objects and classes. But something is also resisting that assimilation, as artists and philosophers have always noticed.

As a matter of fact, artists and philosophers have been always fascinated by the rebellion of collections against categorical order [26], [14]. Let us mention for example Gérard Wajcman's analysis on the status of excess in collections: "Excess in a collection does not mean disorganised accumulation. There is a founding principle: for a collection to be so – even in the eyes of the collector – the number of works needs to exceed the material capacities of displaying and stocking the entire collection at home. Someone living in a studio apartment may very well have a collection: he will only need to not be able to display at least one work in his apartment. It is for this reason that the reserve is one full part of collections. Excess can also apply to memorizing abilities: for a collection to be so, the collector should be incapable of remembering all the pieces he possesses (…). In fact, he either needs to have enough pieces to reach the 'too many' and to 'forget' he had this or that one, or needs to be compelled to leave some outside his place. To put it in a nutshell, what makes a collection is that the collector should not have total power over his collection".

The process of extending a collection is potentially infinite, even if the collection is necessarily undetermined, *temporarily* finished. Practically speaking, a collection ceases to exist as something other than a commonplace correlate whenever the collector loses interest in its extension: he then stops reiterating the acquiring gesture and/or the reconstitution of the collection in an intimate dwelling comes to an end. Both acts have the same essence: in order to keep the collection in an intimate sphere, the collector re-generates the collection, working on his very logic of growth, yet unaware of it. Re-production balances the collection's heavy trends and facilitates new links among the pieces, hence setting up new similarities that will eventually influence the acquiring logic. Strangely enough, desire becomes knotted to difference. Objects enter the collection via the *being different* predicate; they only become similar later on, as being different is what they have in common, hence setting up what Jean-Claude Milner calls a *paradoxical* class.

"A private collector's scene is not his apartment but the whole world. It's important to stress that the major part of his collection in not to be found at his place, his collection is yet to come, still scattered all over the world. Any gallery or fair represents the possibility of chancing on his collection yet to come." ([26]).

Undoubtedly sensitized by those who have long considered the strange condition of collections, object-oriented software designers understood that computer modelling of collections needed the support of heterogeneous computer objects, combining private characteristics—which the objects collected are usually referred to—with characteristics that come from the activities in which these objects are collectively committed.

Curiously, the affinities between classes, collections, singularities and disorders like stack, mass, troop, jumble and other hodgepodges (the last disorders, like collections, cannot exist without a common significant space) have now changed their polarities: classes are definitely different from organizational spatial-based regimes like collections and other "disorders", which now appear to only differ from some degree.

More accurately Jean Piaget and Bärbel Inhelder [13] propose to distinguish *figural* collections from *non-figural* ones. They begin by recalling that a class requires only two categories of relations to be constituted:

- Common qualities to its members and to those of its class, and specific differences that distinguish its own members from other classes ones (comprehension);
- Relations part-whole (belongings and inclusions) determined by "all", "some" and "no one" quantifiers, applied to members of the considered class and to members of classes whose it belongs, qualified as extensions of the class.

For example, cats share in common several qualities owned by all the cats, some of them being specific and some others belonging

also to other animals. But no consideration about space never enter into such a definition: cats may be grouped or not in the space without any change concerning their class definition and properties.

Piaget then defines *figural collections* through the introduction of meaning linked to spatial or/and temporal disposal: a figural collection is a figure because of the spatial links between its elements, when non-figural collections and classes are figure-independent. Organizing knowledge has then to do with the setting of an exhibition, moving to the *paidia* side because forgetting formal, non-figural criteria.

4.2 Similarity *vs.* contiguity parsimony

The current models for information search too often assume that the function and variables defining the categorization are known in advance. In practice, however, when searching for information, experimentation plays a good part in the activity, not due to technological limits, but because the searcher does not know all the parameters of the class he wants to create. He has got special hints, but these evolve as he sees the results of his search. The procedure is dynamic, but not totally random, and this is where the collection metaphor is interesting.

Placing objects in metastable space/time always carries out the collector's experimentation. Here, the intension of the future category has an extensive figure in space/time. And this system of extension (the figure) gives as many ideas as it produces constraints. What is remarkable is that when we collect something, we always have the choice between two systems of constraints, irreducible one to the other. This artificial tension for similarity/contiguity is the only possible kind of freedom allowing us to categorize by experimentation.

This consideration shows the necessity in the design of intelligent applications to take spatial, temporal and spontaneous organization into account, having in mind the ideas brought by collections and exhibitions. As the 'natural' tendency, according to Caillois, consists in moving to formal approaches, we should insist on spatiotemporal approaches at the very beginning of application design.

5 LIFESTREAMING TENDENCIES

The collector attitude is made of *kairos [29],* in the ancient Greek meaning of opportunity, conciliating both available concurrent but irreducible approaches, similarity *vs.* contiguity, meta-playing both with *ludus* and *paidia*. This could be part of the abstract truth of games, as explored by A. Turing within his famous *Imitation game*.

At a crucial moment where service providers tend to offer us social networks timelines/aggregators and general lifestreaming tools for recollecting our whole social and personal lives[2], it is important to renew our frameworks for better innovative capacities.

[2] See for example:
- http://www.youtube.com/watch?v=mg_QZosJMGA, http://www.faveous.com/,
- http://lifestream.glifestream.aim.com/,
- http://itunes.apple.com/fr/app/life-stream-hub-reseaux-sociaux/id432768222?mt=12,
- http://www.youtube.com/watch?v=rA6czHYejWM,
- http://www.youtube.com/watch?v=px9k4hX0oLY,
- http://www.youtube.com/watch?v=oCvB3blWn1E

REFERENCES

[1] Ayse, P-S., Ilyas, C., Varol., A., *Turing Test: 50 Years Later*, Minds and Machines, 10 (4), November 2000.

[2] Blair, A-M. *Too much to know.* Yale Univ. Press, 2010.

[3] Blay, W., *Why The Turing Test is AI's Biggest Blind Alley? — From Machines and Thought: The Legacy of Alan Turing.* Mind Association Occasional Series, 1. Oxford University Press, USA, USA, 1996, pp. 53-62. ISBN 9780198235934.

[4] Caillois. R., *Man, Play and Games,* University of Illinois Press, 2001. For the French original version, see: Les jeux et les hommes — le masque et le vertige, Idées Gallimard, 1967.

[5] Colby, K.M., Hilf, F.D., Weber, S. & Kraemer, H.C., *Turing-like indistinguishability tests for the validation of a computer simulation of paranoid processes,* Artificial Intelligence 3:199-221, 1992.

[6] Copeland, B.J., (ed) *The Essential Turing.* Oxford: Clarendon Press: 488, 2004.

[7] Dupuy, J-P., *Aux origines des sciences cognitives.* La découverte, 1994.

[8] Fink, E., *Le jeu comme symbole du monde,* Editions de minuit, 1966.

[9] Hodges, A., *Alan Turing, The Enigma of Intelligence,* Random House, 1982.

[10] Lassègue, J., *Turing,* Les Belles lettres, 1998.

[11] Moor, J., *The Status and Future of the Turing Test,* Minds and Machines, 11: 77-93, 2001.

[12] Pachet, F., La Burthe, A., Zils, A., Aucouturier J.-J., *Popular music access: The Sony music browser*, Journal of the American Society for Information Science and Technology, Volume 55, issue 12, pages 1037-1044.

[13] Piaget, J., Inhelder, B., *La genèse des structures logiques élémentaires*. Neuchâtel : Delachaux et Niestlé, 1980.

[14] Pomian, K., *Collectionneurs, amateurs et curieux*. Gallimard, 1987.

[15] Preston, J., Bishop, M. (eds), *Views into the Chinese Room: New Essays on Searle and Artificial Intelligence,* Oxford & New York: Oxford University Press, 2002.

[16] Rousseaux, F., Bonardi, A., *Similarité en intension vs en extension : à la croisée de l'informatique et du théâtre.* Revue d'intelligence artificielle (RIA 05), Volume 19, N ° 1-2/2005, Paris, Hermès-Lavoisier, pages 281-288.

[17] Rousseaux, F., *La collection, un lieu privilégié pour penser ensemble singularité et synthèse*. Revue Espaces Temps, 2005.

[18] Searle, J., *Intentionality.* Cambridge University Press, 1983.

[19] Searle, J., *Minds, Brains, and Programs,* Behavioral and Brain Sciences, 3: 417-457, 1981.

[20] Snell, J., Atkins, M., Recordon, D., Messina, C., Keller, M., Steinberg, A., Dolin, R., *Activity Base, Schema (Draft),* activity-schema-01, Internet-Draft, May 27, 2011.

[21] Teuscher, C., Turing's Connectionism. *An Investigation of Neural Network Architectures.* Heidelberg: Springer-Verlag, 2002, 1-85233-475-4.

[22] Traiger, S., *Making the Right Identification in the Turing Test,* Minds and Machines, 10: 561-572, 2000.

[23] Turing, A., *Computing Machinery and Intelligence.* Mind, Vol. 59, n°36, pp. 433-460, 1950.

[24] Turing, A., *Solvable and Unsolvable Problems.* Science News 31, 1954.

[25] Turing, A., Girard, J-Y., *La machine de Turing.* Seuil Points Sciences, 1983.

[26] Wajcman, G., *Collection*. Paris : Nous, 1999.

[27] Winnicott, D., *Jeu et réalité.* Folio essais, 1971

[28] Whitby, B., *The Turing Test: AI's Biggest Blind Alley?* In: Millican, P. & Clark, A. (eds) (1996) Machines and Thought: The Legacy of Alan Turing. Mind Association Occasional Series 1, pp. 53-62. New York & Oxford: Oxford University Press, 1996.

[29] Moutsopoulos, E., *Kairos : la mise et l'enjeu,* Vrin, 1991.

Meta-morphogenesis and the Creativity of Evolution

Aaron Sloman [1]

Abstract.

Whether the mechanisms proposed by Darwin and others suffice to explain the achievements of biological evolution remains open. One problem is the difficulty of knowing exactly what needs to be explained. Evolution of information-processing capabilities and supporting mechanisms is much harder to detect than evolution of physical form, and physical behaviours in part because much goes on inside the organism, and in part because it often has abstract forms whose physical manifestations do not enable us to identify the abstractions easily. Moreover, we may not yet have the concepts required for looking at or thinking about the right things. AI should collaborate with other disciplines in attempting to identify the many important transitions in information processing capabilities, ontologies, forms of representation, mechanisms and architectures that have occurred in biological evolution, in individual development (epigenesis) and in social/cultural evolution – including processes that can modify later forms of evolution and development: meta-morphogenesis. Conjecture: The cumulative effects of successive phases of meta-morphogenesis produce enormous diversity among living information processors, explaining how evolution came to be the most creative process on the planet.

1 Life, information-processing and evolution

Research in a variety of disciplines has contributed a wealth of observations, theories and explanatory models concerned with the diversity of living organisms on many scales, from sub-microscopic creatures to very large animals, plants and fungi, though many unsolved problems remain about the processes of reproduction, development and growth in individual organisms. Many animal competences are still not replicated in machines. I suggest this is in part because of the difficulty of characterising those competences with sufficient precision and generality. Instead researchers focus on special cases inadequately analysed and their models do not "scale out". By studying many more intermediate stages in evolution and development we may achieve deeper understanding of existing biological information processing, and find clues regarding the layers of mechanisms supporting them.

Conjecture: we cannot understand specific sophisticated animal competences without understanding the creativity of biological evolution that produces not only those designs, but also many others. Studying only a few complex cases of animal cognition, for instance pursuing the (in my view hopelessly ill-defined) goal of "human-level AI" [13], may be like trying to do chemistry by studying only a few complex molecules. Likewise trying to replicate selected aspects of some competence (e.g. 3-D vision) while ignoring others may lead to grossly oversimplified models, such as AI "vision" systems that attach labels (e.g. "mug") to portions of an image but are of no use to a robot trying to pick up a mug or pour liquid out of it. Solutions need to "scale out" not just "scale up".[2]

I'll attempt to explain the conjecture, inviting collaboration on the task of identifying and analysing transitions in information processing functions and mechanisms produced by evolution, in humans and also in other species that inhabit more or less similar niches. This is the "meta-morphogenesis" project.[3] In contrast, recent fashions, fads, and factions (e.g. symbolic, neural, dynamical, embodied, or biologically inspired AI) may all turn out to be limited approaches, each able, at best, to solve only a subset of the problems.

2 Diversity of biological information-processing

Every complex organism depends on many forms of information-processing, for controlling aspects of bodily functioning, including damage detection and repair, along with growth and development of body-parts and their functions, and also for behaviours of whole individuals at various stages of development, and also new learning.

Much research has been done on transitions produced by evolution, but, as far as I know, there has not been systematic investigation of *evolutionary transitions in information-processing functions* and mechanisms and their consequences. In [12] the main transitions in information-processing mentioned are changes in forms of communication, ignoring non-communicative uses of information, e.g. in perception, motivation, decision making, learning, planning, and control of actions [18, 19], which can both evolve across generations and change during development and learning. In some species, there are also changes of the sort labelled "Representational Redescription" in [11]. There are also within-species changes in cooperative or competitive information processing, including variation between communities. Conjecture: changes in information-processing help to *speed up and diversify* processes of evolution, learning and development. For example, evolution of individual learning mechanisms, allowed products of evolution to change more rapidly, influenced by the environment.

Forms of representation and ontologies. We have known for decades that how information is represented can significantly affect uses of the information, including tradeoffs between rigour and efficiency, ease of implementation and expressive power, applicability of general inference mechanisms and complexity of searching. I suspect that similar constraints and tradeoffs, and probably many more were "discovered" long ago by biological evolution. As far as I know nobody has surveyed the tradeoffs and transitions that are relevant to uses of information in organisms. There are comparisons between the generality of logic and the

[1] School of Computer Science, University of Birmingham, UK web: http://www.cs.bham.ac.uk/~axs A longer, open access, version of this paper is freely available at http://tinyurl.com/BhamCog/12.html#1203

[2] Compare McCarthy's requirement for "elaboration tolerance"

[3] http://tinyurl.com/BhamCog/misc/m-m.html

usefulness of domain specific "analogical" representations [17, Chap 7]; and between representing structures, properties and relationships with high precision and "chunking" information into fuzzy categories, useful, for example, in learning associations, making predictions and forming explanations, each covering a range of possibilities with small variations [36]. Evolution seems to have discovered the importance of such discretisation, including meeting requirements related to learning about generalisations that hold across time and space, for instance generalisations about the properties of different kinds of matter, and generalisations about consequences of various types of action in various conditions.

Somatic and exosomatic ontologies A survey of varieties of *information contents* available to organisms would include types restricted to internal and external sensor states and effector signals, i.e. *somatic* information, and also the *exosomatic* ontologies used in organisms that evolved later, referring to objects, relationships, processes, locations, routes, and other things outside themselves. Still more sophisticated organisms can speculate about and learn about the hidden contents of the different kinds of matter found in the environment, including humans developing theories about the physics and chemistry of matter, using newly created exosomatic, theory-based (ungroundable) ontologies.[21]

Ontologies with relations Exosomatic ontologies typically locate objects, parts of objects, structures, events and processes in both space and time, so that they have spatial and temporal relationships. Information about relationships can be essential for some forms of action, e.g. direction and distance to something dangerous or something desirable, or whether helpless offspring are hidden in a tunnel or not. Spatial relations can involve different numbers of entities - X is above Y, X is between Y and Z, X is bigger than the gap between Y and Z, etc. Some objects, and some processes, have many parts with multiple relationships between them, and processes include various ways in which relationships can change, continuously or discretely. (Compare [14].) Do we know which species can acquire and use relational information, and when or how it first evolved, or how many forms it can take, including logical (Fregean) and analogical (e.g. diagrammatic, pictorial, model-based) representations? Early biological relational representations were probably molecular. Multi-strand relations involve objects with parts related to other objects with parts e.g. parts of a hand and parts of a mug. Which animals can reason about multi-strand processes?

Information about causal relationships is essential for making plans and predictions. It is not clear what sorts of causal understanding different organisms can have. Jackie Chappell and I have argued for at least two different sorts of causal knowledge (a) correlational/statistical causation (Humean) and (b) structural, mathematically explainable causation.[4] When did they evolve?

How should scalar variation be represented? A common assumption by researchers in several disciplines is that organisms and intelligent robots necessarily represent spatial structures and relationships using global metrics for length, area, volume, angle, curvature, depth speed, and other scalar features. These modes of representation first occurred in human thought only relatively recently (following Descartes' arithmetisation of geometry), so they may not be available to young children and other animals: perhaps evolution produced much older, and in some ways more powerful, ways of representing and using spatial relationships, without numerical coordinate systems? I suspect that Descartes' forebears, many animals, and pre-verbal children in our culture make use of networks of partial orderings (of distance, direction, angle, curvature, speed, size, and other properties) enhanced with semi-metrical relations refining orderings (e.g. X is at least three times as long as Y but not more than four times as long). Specifying exactly how such representations might work remains a research problem. Obviously, many animals including nest-building birds, primates, hunting mammals, and elephants understand spatial structures and affordances in ways that are far beyond the current state of computer vision/robotics. Neuroscientists and vision researchers in psychology seem to lack a theoretical framework to describe or explain such competences. One problem in such research is a tendency to confuse the ability to understand and reason about spatial relationships and processes with the ability to *simulate them*, as is done in computer game engines. Our brains cannot perform similar simulations.

Conditional control. Organisms need to be able to generate motor control signals or sequences of signals partly on the basis of information about the environment and partly under the control of goals and plans. For this, information is needed about internal states, such as energy or fluid needs, and also predicted needs, so as to initiate actions to meet anticipated requirements. Such choices depend on information about both external states and internal states (e.g. desires, preferences). So requirements and uses for information processing can vary in ways that depend on static or changing factors, some within the organism (e.g. need for a particular sort of nutrient), some in the environment (e.g. the local or remote spatial relationships between various surfaces and objects), and some of that depend on the sensory-motor morphology of the organism, e.g. whether it has an articulated body with mobile grippers, and whether it has visual, olfactory, auditory, tactile, haptic, proprioceptive or other sensors.

Precocial/Altricial tradeoffs Additional information-processing requirements depend on how individuals change in shape, size, strength and needs, which depend on what parents can do to help offspring. Many carnivores and primates are born weak and helpless and as they grow, larger, heavier and stronger, they engage in forms of movement for which new kinds of control are required, not all encoded in the genome (for example manipulation of objects that did not exist in the evolutionary history of the species [34]).

In many species, development requires use of information about the environment in setting and achieving ever more complex goals, allowing cumulative development of forms of control required by adults. This process can include play fighting, using conspecifics of similar size and competence. Contrast larvae, that, after a phase of crawling and eating, pupate and transform themselves into butterflies that apparently do not need to learn to fly, feed or mate. Information for the later phase of must somehow have been present in the caterpillar stage where it was of no use. Some of the tradeoffs between nature and nurture found in animals and likely to be relevant to future robots are discussed in [31, 5]. Not using those biological forms of representation may explain why our robots, impressive as they are in limited ways, lack the generality and flexibility of pre-verbal humans and many other animals.

On-line *vs* off-line intelligence. The simplest known organisms are surprisingly complex.[5] All require information-based control for growth and reproduction, unlike sediment layers that simply accrue whatever external physical processes provide. Informed growth requires selection of nutrients outside the organism. If not everything in the environment is suitable, microbes can use sensors that react differently to chemicals in the surrounding soup, ingesting only nutrients (except when deceived). Such organisms have information-

[4] http://tinyurl.com/BhamCog/talks/wonac/

[5] https://en.wikipedia.org/wiki/Archaea

processing needs that are highly localised in space and time: so that transient sensing and control suffice – perhaps even just a fixed set of triggers that initiate responses to different types of contact. Complex online control uses continuously sensed information, e.g. about directions, about changing gaps, about local chemical gradients, used in deciding whether to modify motor signals, e.g. so as to increase concentration of nutrients or decrease concentration of noxious substances, or towards or away from light, etc. Using direction and magnitude of changes requires more complex mechanisms than detecting presence or absence, or thresholding. Feedback control using "hill-climbing" requires access to recent values, so that new ones can be compared with old ones in order to select a change.

On-line intelligence involves using information as it is acquired. *Off-line* intelligence acquires information usable later, in combination with other information, and for several different purposes. Off-line mechanisms transform sensed or sent information into new formats, stored for possible uses later, if required. Storing more abstract information can be useful because very precise details may not be relevant when one is thinking or reasoning about a situation that one is not in at the time, and also because information in a more economical and abstract form may allow more useful generalisations to be discovered, and may be simpler to combine with other forms of information.

Combining on-line and off-line intelligence. Doing something and understanding why it works requires parallel use of on-line and off-line intelligence. Some tasks, for instance mapping terrain while exploring it (SLAM) combine online and offline intelligence, as new sensor information is integrated into an multi-purpose representation of the large scale structure of the environment, where useful spatial/topological relationships and spatial contents are stored, not sensor readings. However, it is useful sometimes to store "summary sensory snapshots" for comparison with future snapshots, or to allow information to be derived from the low level details at a later time.

All this requires specific mechanisms, architectures, and forms of representation. Their uses will depend on what the environment is like and on previously evolved features of the species. We need more detailed analyses of the different functions and the mechanisms required for those functions, and how their usefulness relates to various environments and various prior design features.

Duplicate then differentiate *vs* abstraction using parameters A common pattern of change leading to more complex biological structures or behaviours starts by duplicating an already learnt or evolved specification, then allowing one, or both, copies to change, either across generations or within a lifetime. Without this a single fertilised cell could not grow into a complex organism with varied parts competences. That is also a common pattern in the development of engineering design knowledge. Another common pattern in mathematics and engineering inserts gaps into something learnt, to form a re-usable specification whose instances can take many forms that depend on the gap-fillers, e.g. algebraic structures defined in terms of types of operators and types of objects, which take different forms for different instances. This can also be a powerful form of individual learning. I suspect evolution also found ways to use it, speeding up evolution by allowing new complex sub-systems to be created by instantiating existing patterns (as opposed to duplicating old instances). This can support learning in diverse environments. It is a core feature of mathematical discovery. We need to study more biological examples.

Use of virtual machinery. Use of virtual machinery instead of physical machinery often facilitates extendability, monitoring, de-bugging, and improving designs and re-using them in new contexts. Conjecture: biological evolution "discovered" advantages of use of virtual machinery long before human engineers did, especially in self-monitoring and self-modifying systems, with many important consequences. Some virtual machines merely provide new implementations of functionality previously provided in hardware, whereas others are non-physically specified, for example, virtual machines for performing operations like forming intentions, detecting threats, evaluating strategies, extending ontologies. Describing these requires use of concepts like information, reference, error, perception, trying, avoiding, failing, planning, learning, wanting, and many more that are not definable using concepts of the physical sciences. When chess virtual machine runs we can describe what it does using concepts like pawn, threat, detect, fork, mate, plan, attempt, fail, but those descriptions cannot be translated into the language of physics, even though the chess machine is fully *implemented* physically. A translation would have to summarise all possible physical implementations using different technologies, including future ones about which we currently know nothing, so our concepts cannot presuppose their physical features [26].

Such virtual machinery is *fully implemented* in physical mechanisms (some of which may be in the environment) and cannot survive destruction of the physical computer, though a running VM can sometimes be transferred to a new computer when a physical malfunction is imminent: an option not yet feasible for biological virtual machinery. Mechanisms for supporting a class of virtual machines can enormously simplify the process of producing new instances, compared with having to evolve or grow new instances with new arrangements of physical matter. This could speed up both evolution and learning, as it speeds up engineering design.

Besides *single function* virtual machines (or application machines, e.g. a spelling checker) there are also *platform virtual machines* that support development of a wide range of additional machines implemented on the platforms, sharing the benefits of previously developed VM components with multiple uses. Platform VMs include programming language systems (e.g. a python VM) and operating systems (e.g. a linux VM). Contrary to the common notion of computation as inherently *serial* (as in a simple Turing Machine) many VMs inherently include *multiple concurrently active subsystems* interacting with one another and with things outside the machine (e.g. information stores, sensors, robot arms, displays or other networked systems).[6] Perhaps evolution of new platform VMs sped up evolution of new information-processing functionality.

These ideas raise deep unanswered questions about how specifications for different sorts of development and learning capabilities are encoded in a genome, and what needs to change in decoding processes to allow changes from mechanisms specified by their hardware (e.g. chemical implementation) to mechanisms encoded in terms of a previously evolved virtual machine.

Specifying functions rather than behaviours or mechanisms Human engineers and scientists have increasingly used virtual machinery to achieve more sophisticated design goals, driven by new engineering requirements, including the need for programs too large to fit into physical memory, the need to be able to run a program in different parts of physical memory without altering addresses for locations and the need to use novel forms of hardware. Design machines specified in terms of information processing *functions*

[6] The CogAff schema allows diverse highly concurrent VMs of varying complexity and functionality http://tinyurl.com/BhamCog/#overview

rather than their physical *structures and behaviours*, postpones the task of producing physical implementations and allows different solutions. Many computing systems are specified not in terms of the behaviours of electrons or transistors, etc., but in terms of operations on numbers, strings, arrays, lists, files, databases, images, equations, logical formulae, mathematical proofs, permissions, priorities, email addresses, and other notions relevant to providing a computing service. Programmers attempting to debug, modify, or extend such programs, normally do not think about the physical processes, but about the structures and processes in the running VM. Explaining how the program works, and what went wrong in some disaster typically involves reference to events, processes and causal interactions within the VM, or in some cases relations between VM processes and things in the environment.

Some philosophical functionalists define mental phenomena in terms of how they affect input-output mappings, e.g. [4], but this ignores designs for complex virtual machinery specified in terms of structures, processes and causal interactions in the machine, not input-output relationships – "virtual machine functionalism".

Meta-semantic competences and ontologies A semantic competence is the ability to refer to things. A *meta-semantic competence* involves being able to think about, reason about, make use of, or detect something that refers, or intends, or perceives (including possibly oneself). Such competences can take many forms. Some are shallow, while others are deep. Abilities to detect aspects of X's behaviour that indicate what X perceives, or what it intends, or whether it is annoyed or fearful, etc. can feed into decisions about how to act towards X. In the shallowest forms this can involve only evolved or learnt reactions to shallow behaviours (e.g. running, snarling), etc. Deeper meta-semantic competences include representing specific contents of percepts, intentions, preferences, beliefs, etc. of others, and possibly hypothetical reasoning about such states (what would X do if it knew that A, or desired B?). Dennett, in [7], and elsewhere, refers to this as adopting "the intentional stance", but seems to be reluctant to accept that that can involve representing what is going on inside the individual referred to. Developmental psychologists have studied "mind-reading" abilities, e.g. [2], but we still lack a comprehensive theory of the varieties of forms of semantic competence, their biological roles, which organisms have them, how they evolved, how they develop in individuals, how they can vary from one individual to another, and so on. The more sophisticated meta-semantic competences require abilities to refer to virtual machine events, states and processes. How this is done, including handling "referential opacity" is still a matter of debate: some researchers emphasis use of special logics (modal logics), while others (rightly!) emphasise architectural support for meta-semantic reasoning.

Re-usable protocols Recent history of computing included development of many specifications of re-usable protocols including networking protocols, protocols for communication with peripheral devices (screens, sensors, keyboards, etc.) and protocols for inter-process communication (among many others). Use of DNA and a set of transcription mechanisms can be viewed as a biological version of a multi-function protocol. There may be many others worth looking for, perhaps not shared universally, but perhaps shared between species with a common heritage, or between different functions within individuals or within a species. I conjecture that the advantages of use of VMs for specifying new functionality, for debugging, for modifying, extending, analysing processes were "discovered" by evolution long before human engineers. This suggests that much mental functioning cannot be understood as brain functioning, and research into minds and brains, what they do, and how they work, needs to be informed by what can be achieved by VMs whose relationship to the physical machinery of the brain may be very complex and indirect. How and when this first occurred, and how specifications for virtual implementations are encoded in genomes are unanswered questions. Some new biological competences initially developed using VMs might later use more efficient, but more inflexible, physical implementations. Sometimes the reverse might occur: competences implemented in brain mechanisms are later be replaced by VMs that provide more flexibility, more extendability, and more diversity of use [5].

Self-monitoring at a VM level. Programs that monitor and modify running systems (including themselves) can benefit from focusing on VM structures and processes as well as the underlying physical machinery. I suspect biological evolution found many uses for VMs long before there were humans on the planet. If machines or animals can introspect enough to find out that they create and manipulate non-physical entities, that could lead them to invent muddled philosophical theories about minds and bodies, as human philosophers have done [26, 28].

Representing the actual and the possible (i.e. affordances). Information-processing functions so far described involved acquiring, transforming, storing, combining, deriving, and using information about what is or has been the case, or what can be predicted: types of *factual* information. Some organisms can also represent and use information that is not about what exists but rather about what is, was, or will be *possible*. This may require new architectures, forms of representation, and mechanisms. The ability to acquire and use short-term information about possibilities for and restrictions on physical action, and restrictions on action was referred to by Gibson [8] as the ability to perceive and use "affordances", where the affordances can be either positive (enabling or helping) or negative (preventing, hindering or obstructing). There are many more ways of detecting, reasoning about, producing, or using possibilities for change in the environment or restrictions on possibilities [20, 30], included in competences of particular individuals or particular types of organism. These include representing proto-affordances (possibilities and constraints involving physical objects), vicarious affordances (for other agents - including predators, prey, collaborators, offspring, etc.), epistemic affordances, deliberative affordances, and others described in [30, 25]. For organisms with meta-semantic competences (summarised above) types of affordance that can arise will be much greater than for animals that can represent or reason only about physical/spatial possibilities.

Yet more complexity in the ontology used, the forms of representation, and the information processing arises from the need not only to represent what actually exists, at any time, but also what is and is not possible, what the constraints on possibilities are, and how those possibilities and constraints can depend on other possibilities.

People can use information without being able to answer questions about it, e.g. human syntactic competences. So tests for meta-semantic competences in young children can be misleading if the tests require explicit meta-knowledge.[7] When and how all these information-processing capabilities arose in biological organisms is not known. There are many intermediate cases between the simplest uses of grippers and the competences of human engineers. We may not be able to understand the latter without understanding more about

[7] One of the forms of "representational redescription" discussed in [11] is the transition from having a competence to being able to articulate its features.

the intermediate capabilities on which they depend.

Motivational and deliberative competences Organisms have changing needs that influence behaviours. Some changes directly trigger reactions that can reverse, or make use of the change: for instance shivering can be triggered by mechanisms detecting a drop in temperature. Evolution discovers some conditions under which such "reactive" responses are beneficial, and encodes genetic information producing the mechanisms in new individuals. But evolving reactions to needs can be very slow. It can take many generations for arrival of a new danger or a new form of food making a new response useful to lead to evolved behavioural reactions. Instead, between the mechanisms that detect needs and the mechanisms that produce behaviours, evolution interposed mechanisms that select goals triggered by detected needs, which in turn trigger planning mechanisms to select actions to achieve the goals [15]. Much AI research has been concerned with ways of achieving this. From a biological standpoint, the use of such mechanisms provides opportunities for novel evolutionary or development processes concerned with (a) selecting new goals, (b) finding plans for achieving them and (c) using plans to control actions. Many variants of these patterns are relevant to the meta-morphogenesis project. A type of evolution that generates new kinds of rewards is described in [16]. Another possibility is adding mechanisms that generate goals not because they will satisfy some need or provide some reward, but merely because there are currently no important tasks in progress, and an opportunity for generating a certain sort of goal has been detected. In [24] it is argued that reflex triggering of such goals along with mechanisms for achieving goals, will sometimes cause useful new things to be learnt, even if achieving the goal has no reward value. Failing to achieve goals often provides more valuable learning than succeeding.

Factorisation of the link between needs and actions introduces modularity of design, allowing opportunities for separate types of improvement, with benefits shared between different needs – perhaps permitting evolution and/or learning to be speeded up through sharing of benefits.

"Peep-hole" vs "Multi-window" perception and action. Although it would take up too much space to explain fully here, there is a distinction between architectures in which there is limited processing of perceptual input and the results of the processing are transmitted to various "more central" mechanisms (e.g. goal formation, or planning subsystems), which I call "peep-hole" perception, and architectures using "multi-window" perception in which perceptual subsystems do several layers of processing at different levels of abstraction in parallel, using close collaboration with the layers and with more central mechanisms (e.g. parsing, searching for known structures, interpreting). Multi window perceptual processing is crudely illustrated in this figure http://tinyurl.com/BhamCog//crp/fig9.6.gif Likewise a distinction can be made between peep-hole and multi-window *action control* subsystems. For example a multi-window action could include, in football, concurrently running towards a goal, dribbling the ball, getting into position to shoot, avoiding a defender and eventually shooting at the goal. Linguistic production, whether spoken, handwritten, or signed always has multiple levels of processing and reference. (Compare Anscombe's analysis of intention in [1].)

The use of multi-window perception and action allows a wider range of information processing at different levels of abstraction to be done concurrently with sensory inputs and motor outputs, permitting more powerful and effective perception and action subsystems to evolve or be developed. I conjecture that the multi-window solutions are used by far more species than have been noticed by researchers, and are also well developed in pre-verbal human children, though yet more development occurs later.

Transitions in representational requirements. Even in this overview of a tiny subset of evolutionary processes we find requirements for different information structures: binary on/off structures in a detector, scalar values varying over time used in homeostatic and "hill-climbing" control processes, information about spatial and topological relationships between surfaces and regions that are not currently being sensed, that are needed for planning routes, and information about possibilities for change, constraints on change, and consequences of possible changes, needed for selecting and controlling actions manipulating physical structures, along with use of meta-semantic information about information users and information-bearing structures. These requirements are related to old philosophical problems, e.g. How is information about possibilities and impossibilities be represented? Can young children, or non-human animals, make use of modal logics, and if not what are the alternatives?

Often it is not obvious how a particular type of information will be most usefully represented for a particular type of organism. Many researchers, whether studying animal cognition or attempting to design intelligent robots, assume that the representation of spatial structures and relationships must use something like global 3-D coordinate systems, forgetting that such forms of representation were a relatively late discovery in human culture. Humans made tools, machines, houses, temples, pyramids, aqueducts and other things requiring a deep understanding of spatial structures and processes before geometry had been arithmetized by Descartes, so it is possible that they were using some other form of representation.

Re-representation and systematisation. The main motive that originally got me into AI was the hope of showing that Immanuel Kant's theories about the nature of mathematical knowledge [10], were superior to the opinions of most other philosophers, including Hume, Mill, Russell, and Wittgenstein. I hoped to show this by building a robot that started off, like infants and toddlers discovering things about spatial structures and motions empirically and later finding ways of reorganising some of the information acquired into theories that allowed it to *prove* things instead of discovering them empirically, e.g. using diagrammatic proofs of the sort used in Euclidean geometry [23]. This task proved far more difficult than I initially hoped, in part because of the great difficulty of giving robots animal-like abilities to perceive, understand, and use information about structures and motions in the environment, in order to predict or explain their behaviours, as suggested by Craik [6]. Perhaps something like the processes Karmiloff-Smith labelled varieties of "Representational Redescription" [11], are needed, though there's more than re-description going on, since architectural changes are also required. I suspect these mathematical competences in humans build on precursors found not only in pre-verbal children, but also in other animals with powerful spatial reasoning capabilities required for using complex affordances, as in such as some nest-building birds.[8] This remains an important task for the Meta-morphogenesis project, which may enhance research in AI and psychology on learning and creativity.

Empirical learning *vs* working things out Many forms of learning investigated in AI, robotics and psychology make use of mechanisms for deriving taxonomies and empirical generalisations

[8] See also http://tinyurl.com/BhamCog/talks/#toddler

from collections of examples. The evidence used may come from the experiences of an individual (animal or robot) exploring an environment, finding out what can and cannot be done in it, and what the consequences are, or they may make use of data-mining techniques applied to much larger externally supplied sample sets.

Humans, and many other species, are clearly capable of discovering useful empirically supported patterns, for example linking actions, circumstances and consequences. However, human mathematical knowledge shows that humans are also capable of a different kind of learning – by *working things out*. Collecting empirical generalisations may eventually trigger a switch to another process, which instead of merely using more data to extend known generalisations, takes what is already known and attempts to find a "generative basis" for it. A special case is switching from pattern-based language use to syntax-based language use, a common transition in child development. Syntax-based competences use generative rules and compositional semantics that allow new, richer forms of communication, and also new richer forms of thinking and reasoning – one type of "representational redescription".

I conjecture that the linguistic case is a special development of a more general biological capability, that evolved earlier and in more species, which allows a collection of useful empirical generalisations to be replaced by something more economical and more powerful: a generative specification of the domain. The creation of Euclid's elements appears to have been the result of a collective process of this sort, but that collective cultural process could not have happened without the individual discoveries of new more powerful generative representations of information previously acquired empirically piecemeal [27].

In simple cases the new generative (e.g. axiomatic) representation may be discovered by data-mining processes. However in the more interesting cases it is not sufficient to look for patterns in the observed cases. Instead it is necessary to *extend the ontology used*, so as to include postulated entities that have not been experienced but are invoked as part of the process of explaining the cases that have been experienced. The infinitely small points and infinitely thin, straight and long lines, of Euclidean geometry are examples of such ontological extension required to create a system with greater generative power. This process of reorganisation of knowledge into a new, more powerful, generative form, seems to be closely related to the hypothesis in [6] that some animals can create models that they use to predict the results of novel actions, instead of having to learn empirically which ones work and which ones don't, possibly with fatal costs. The ability of human scientists to come up with new theories that explain old observations, making use of ontological extensions that refer to unobservable entities (e.g. atoms, sub-atomic particles, valences, gravity, genes, and many more) also illustrates this kind of process replacing empirical generalisations with a generative theory.

I suspect that similar transformations that have mostly gone unnoticed also occur in young human children, discovering what could be called "toddler theorems". (See http://tinyurl.com/TodTh) Such transformations could occur, both in humans and some other species, without individuals being aware of what has happened – like children unaware that their linguistic knowledge has been reorganised. Later, as meta-semantic competences develop, individuals may come to realise that they have different kinds of knowledge, some of it empirical, derived from experience, and some generated by a theory. Later still, individuals may attempt to make that new knowledge explicit in the form of a communicable theory

These conjectures about different bases for knowledge about the world are closely related to the main ideas of [11], but came from a very different research programme based on the idea of using AI techniques to solve problems in philosophy of mathematics [23]. I suspect this is closely related to Kant's theories about the nature of mathematical knowledge [10]. Such discoveries are very different in kind from the statistics-based forms of learning (e.g. Bayesian learning) that now dominate much research. The mathematical reasoning shows what *can* be or *must* be the case (given certain assumptions) not what is highly probable: e.g. working out that the angles of a triangle must add up to a straight line, or that 13 identical cubes cannot be arranged in rectangular array other than a 13x1 array, is very different from finding that stones thrown up normally come down: the latter discovery involved no mathematical necessity (until Newtonian mechanics was developed). At present I don't think there are any good theories about either the biological basis of such knowledge or how to provide it for robots.

Enduring particulars For many species the only environmental information relevant to control decisions is information about the *types* of entity in the immediate environment. E.g. is this a place that provides shelter or food? Is that a dangerous predator? Is this conspecific friendly or aggressive? For a variety of different reasons it became useful to be able to re-identify particular individuals, places, and objects at different times (e.g. is this the tool I have already tested, or do I need to test it before using it?). However, as philosophers have noted there are enormous complications regarding tracking individuals across space and time (e.g. is it the same river after the water has been replenished; is this adult the same individual as that remembered child?). This is not the place to go into details (compare [32]), but analysis of the many types of particular and the means of referring to or re-identifying them and the purposes that can serve, can give clues regarding evolutionary and developmental transitions that have so far not been studied empirically and also have not been addressed in robot projects except in a piecemeal, *ad hoc* fashion, with much brittleness.

Meta-management. As information-based controlling processes become more complex, across evolutionary or developmental time-scales, the need arises for them also to be controlled, in ways that can depend on a variety of factors, including the changing needs of individual organisms, their bodily structure, the types of sensorymotor systems they have, their developing competences, and the constraints and affordances encountered in their environments, some of which will depend on other organisms. New forms of control of controlling process are also examples of meta-morphogenesis.

Evolving new mechanisms for turning on each new kind of functionality, without harmfully disrupting other functions, is less useful than using a pre-existing, extendable, mechanism for handing control from one subsystem to another.[9] This can also support centralisation of major decisions, to ensure that all relevant available information is taken into account, instead of simply allowing strongly activated sub-systems to usurp control. Using scalar strength measures, like scalar evaluation functions in search, loses too much information relevant to comparing alternatives.

"Hard-wired", implicit control mechanisms, implemented using only direct links between and within sub-systems, can be replaced by newly evolved or developed *separate and explicit* control functions (e.g. selecting what to do next, how to do it, monitoring progress, evaluating progress, using unexpected information to re-evaluate priorities, etc., as in the meta-management functions described in [3, 35]). Such new control regimes may allow new kinds of functionality

[9] Compare the invention of a procedure call stack for computing systems.

to be added more simply and used when relevant, thereby expanding the opportunities (affordances) for evolution and learning.

From internal languages to communicative languages For some people languages are by definition a means of intentional communication between whole agents. But that ignores the vast amount and variety of types of *internal* information processing using structured forms of representation of varying complexity with compositional semantics e.g. to encode learnt generalisations, perception of complex structures, intentions to perform complex actions, questions, predictions, explanations, and plans – in both non-human animals and pre-verbal children. Philosophers and psychologists who have never thought about how to design a working animal usually never notice the requirements. As argued in [18, 19, 22], there is a natural correspondence between the contents of internal plans and behaviours controlled by the plans. I suggest that a series of evolutionary transitions allowed actions to become communications, initially involuntarily, then later voluntarily, then enhanced to facilitate communication (e.g. for cooperation) and then, using the duplicate and differentiate evolutionary strategy) "hived off" as a means of communication, which evolved into sophisticated sign languages. Later additional requirements (communication at night, and while using hands) might have led to evolution of vocal accompaniments that finally became spoken language. This conjecture has deep implications regarding structures of human and animal brains and minds that need to be explored as part of this project. Further variations in functions and mechanisms both across generations, between contemporary individuals, and between stages of development within an individual would include:
– genetically specified forms of communication (possibly specified in a generic way that can be instantiated differently by different individuals or groups).
– involuntary vs intentional forms of communication. It seems unlikely that the "begging" for food actions of fledglings and young mammals are intentional (in various meanings of that word). In other cases there are different kinds of intentionality and different levels of self-awareness when communication happens.
– other variations include whether there is explicit teaching of means of communication by older individuals (Compare [11])

Varieties of meta-morphogenesis Some examples of evolutionary meta-morphogenesis seem to be restricted to humans. We have a collection of mechanisms (closely related to some of the themes in [11]) that allow humans (a) to acquire novel capabilities by various processes of learning and exploration, including trial and error, (b) to become aware that we have acquired such a new competence or knowledge, (c) find a way to express its content, (d) decide to help someone else (e.g. offspring or members of the same social group) to acquire the competence – through a mixture of demonstrations, verbal explanations, criticisms of incomplete understanding and suggestions for improvement, and (d) to provide cultural artefacts for disseminating the knowledge.

Some previous results of information-processing morphogenesis can alter current processes of morphogenesis, for instance when learning extends abilities to learn, or evolution extends evolvability, or evolution changes abilities to learn, or new learning abilities support new evolutionary processes. Where morphogenesis produces new types of learning or development and new sorts of evolvability, that can be labelled "meta-morphogenesis". A deep explanatory theory will need to characterise the "evolutionary affordances" (generalising Gibson's notion [8]) made use of. In particular, evolved cognitive abilities may provide new affordance detectors, such as mate-selectors, accelerating evolution as agricultural breeding has done. Evolution starts off blind, but can produce new affordance detectors that influence subsequent evolution.

If every new development opens up N new possibilities for development the set of possible trajectories grows exponentially, though only a subset will actually be realised. Nevertheless, the cumulative effects of successive phases of meta-morphogenesis seems to have produced enormous diversity of physical forms, behaviours, and less obviously, types of biological information processing (including many forms of learning, perceiving, wanting, deciding, reasoning, and acting intentionally) making evolution the most creative process on our planet. The diversity may be essential for evolution of (e.g.) mathematicians, scientists, and engineers.

3 Conclusion

I have tried to present a variety of transitions in kinds of information processing that seem to have occurred in the evolutionary history of humans and other species. This is merely a taster, which may tempt more researchers to join the attempt to build a systematic overview of varieties of ways in which information processing changed during biological evolution, with a view to implementing the ideas in future computational experiments. This will require much computationally-guided empirical research seeking information about social, developmental, epigenetic, genetic and environmental transitions and their interactions.

In his 1952 paper Turing showed how, in principle, sub-microscopic molecular processes in a developing organism might produce striking large scale features of the morphology of a fully grown plant or animal. This is a claim that if individual growth occurs in a physical universe whose building blocks permit certain sorts of spatio-temporal rearrangements, complex and varied structures can be produced as a consequence of relatively simple processes.

Darwin proposed that variations in structures and behaviours of individual organisms produced by small random changes in the materials used for reproduction could be accumulated over many generations by mechanisms of natural selection so as to produce striking large scale differences of form and behaviour. This is a claim that if the physical universe supports building blocks and mechanisms that can be used by reproductive processes, then the observed enormous diversity of forms of life can be produced by a common process.

Partly inspired by Turing's 1952 paper on morphogenesis, I have tried to show that there are probably more biological mechanisms that produce changes in forms of information processing than have hitherto been studied, in part because the richness of biological information processing has not been investigated as a topic in its own right, though some small steps in this direction were taken by [9], and others. Moreover it seems that the collection of such mechanisms is not fixed: there are mechanisms for producing new morphogenesis mechanisms. These can be labelled meta-morphogenesis mechanisms. The cross-disciplinary study of meta-morphogenesis in biological information processing systems promises to be rich and deep, and may also give important clues as to gaps in current AI research.

Can it all be done using computers as we know them now? We need open minds on this. We may find that some of the mechanisms required cannot be implemented using conventional computers. It may be turn out that some of the mechanisms found only in animal brains are required for some of the types of meta-morphogenesis. After all, long before there were neural systems and computers, there

were chemical information processing systems; and even in modern organisms the actual construction of a brain does not (in the early stages) use a brain but is controlled by chemical processes in the embryo.

Biological evolution depends on far more than just the simple idea of natural selection proposed by Darwin. As organisms became more complex several different kinds of mechanism arose that are able to produce changes that are not possible with the bare minimum mechanisms of natural selection, although they depend on that bare minimum. This is not a new idea. A well known example is the use of cognition in adults to influence breeding, for instance by mate selection and selective feeding and nurturing of offspring when food is scarce. My suggestion is that we need a massive effort focusing specifically on examples of *transitions in information processing* to accelerate our understanding.

There must be many more important transitions in types of biological information processing than we have so far noticed. Investigating them will require multi-disciplinary collaboration, including experimental tests of the ideas by attempting to build new machines that use the proposed mechanisms. In the process, we'll learn more about the creativity of biological evolution, and perhaps also learn how to enhance the creativity of human designed systems. This research will be essential if we are to complete the Human Genome project.[10][11]

ACKNOWLEDGEMENTS

I am grateful to Alison Sloman for useful discussions of evolution and to Stuart Wray for comments on an early draft of this paper and his diagram summarising the draft.[12] The EU CogX project helped to shape my thinking.

REFERENCES

[1] G.E.M. Anscombe, *Intention*, Blackwell, 1957.
[2] I. Apperly, *Mindreaders: The Cognitive Basis of "Theory of Mind"*, Psychology Press, London, 2010.
[3] L.P. Beaudoin, *Goal processing in autonomous agents*, Ph.D. dissertation, School of Computer Science, The University of Birmingham, Birmingham, UK, 1994.
[4] N. Block. What is functionalism?, 1996. (Originally in The Encyclopedia of Philosophy Supplement, Macmillan, 1996).
[5] J. Chappell and A. Sloman, 'Natural and artificial meta-configured altricial information-processing systems', *International Journal of Unconventional Computing*, **3**(3), 211–239, (2007).
[6] K. Craik, *The Nature of Explanation*, Cambridge University Press, London, New York, 1943.
[7] D.C. Dennett, *The Intentional Stance*, MIT Press, Cambridge, MA, 1987.
[8] J.J. Gibson, *The Ecological Approach to Visual Perception*, Houghton Mifflin, Boston, MA, 1979.
[9] E. Jablonka and M. J. Lamb, *Evolution in Four Dimensions: Genetic, Epigenetic, Behavioral, and Symbolic Variation in the History of Life*, MIT Press, Cambridge MA, 2005.
[10] I. Kant, *Critique of Pure Reason*, Macmillan, London, 1781. Translated (1929) by Norman Kemp Smith.
[11] A. Karmiloff-Smith, *Beyond Modularity: A Developmental Perspective on Cognitive Science*, MIT Press, Cambridge, MA, 1992.

[10] Instead of regarding evolution as a "blind watchmaker", we can think of it as a blind theorem prover, unwittingly finding proofs of "theorems" about what sorts of information-using systems are possible in a physical world. The proofs are evolutionary and developmental trajectories. The transitions discussed here can be regarded as powerful inference rules.

[11] The concept of "information" used here is not Shannon's (purely syntactic) notion but the much older notion of "semantic content", explained more fully in [29] http://tinyurl.com/BhamCog/09.html#905

[12] See: http://www.cs.bham.ac.uk/~axs/fig/wray-m-m-label-small.jpg

[12] J. Maynard Smith and E. Szathmáry, *The Major Transitions in Evolution*, Oxford University Press, Oxford, England:, 1995.
[13] John McCarthy, 'From Here to Human-Level AI', volume 171, pp. 1174–1182. Elsevier, (2007). doi:10.1016/j.artint.2007.10.009, originally KR96.
[14] M. L. Minsky, 'Steps towards artificial intelligence', in *Computers and Thought*, eds., E.A. Feigenbaum and J. Feldman, 406–450, McGraw-Hill, New York, (1963).
[15] M. Scheutz and B.S. Logan, 'Affective vs. deliberative agent control', in *Proceedings Symposium on Emotion, cognition and affective computing AISB01 Convention*, ed., *et al.* C. Johnson, York, (2001).
[16] S. Singh, R.L. Lewis, and A.G. Barto, 'Where Do Rewards Come From?', in *Proceedings of the 31th Annual Conference of the Cognitive Science Society*, eds., N.A. Taatgen and H. van Rijn, pp. 2601–2606, Austin, TX, USA, (2009). Cognitive Science Society.
[17] A. Sloman, 'Interactions between philosophy and AI: The role of intuition and non-logical reasoning in intelligence', in *Proc 2nd IJCAI*, pp. 209–226, London, (1971). William Kaufmann.
[18] A. Sloman, 'What About Their Internal Languages?', *Behavioral and Brain Sciences*, **1**(4), 515, (1978).
[19] A. Sloman, 'The primacy of non-communicative language', in *The analysis of Meaning: Informatics 5, ASLIB/BCS Conf, Oxford, March 1979*, eds., M. MacCafferty and K. Gray, pp. 1–15, London, (1979). Aslib.
[20] A. Sloman, 'Actual possibilities', in *Principles of Knowledge Representation and Reasoning: Proc. 5th Int. Conf. (KR '96)*, eds., L.C. Aiello and S.C. Shapiro, pp. 627–638, Boston, MA, (1996). Morgan Kaufmann Publishers.
[21] A. Sloman. Why symbol-grounding is both impossible and unnecessary, and why theory-tethering is more powerful anyway., 2007.
[22] A. Sloman. Evolution of minds and languages. What evolved first and develops first in children: Languages for communicating, or languages for thinking (Generalised Languages: GLs)?, 2008.
[23] A. Sloman, 'The Well-Designed Young Mathematician', *Artificial Intelligence*, **172**(18), 2015–2034, (2008).
[24] A. Sloman, 'Architecture-Based Motivation vs Reward-Based Motivation', *Newsletter on Philosophy and Computers*, **09**(1), 10–13, (2009).
[25] A. Sloman, 'Some Requirements for Human-like Robots: Why the recent over-emphasis on embodiment has held up progress', in *Creating Brain-like Intelligence*, eds., B. Sendhoff, E. Koerner, O. Sporns, H. Ritter, and K. Doya, 248–277, Springer-Verlag, Berlin, (2009).
[26] A. Sloman, 'How Virtual Machinery Can Bridge the "Explanatory Gap", In Natural and Artificial Systems', in *Proceedings SAB 2010, LNAI 6226*, eds., S. Doncieux and et al., pp. 13–24, Heidelberg, (August 2010). Springer.
[27] A. Sloman, 'If Learning Maths Requires a Teacher, Where did the First Teachers Come From?', in *Proc. Int. Symp. on Mathematical Practice and Cognition, AISB 2010 Convention*, eds., Alison Pease, Markus Guhe, and Alan Smaill, pp. 30–39, De Montfort University, Leicester, (2010).
[28] A. Sloman, 'Phenomenal and Access Consciousness and the "Hard" Problem: A View from the Designer Stance', *Int. J. Of Machine Consciousness*, **2**(1), 117–169, (2010).
[29] A. Sloman, 'What's information, for an organism or intelligent machine? How can a machine or organism mean?', in *Information and Computation*, eds., G. Dodig-Crnkovic and M. Burgin, 393–438, World Scientific, New Jersey, (2011).
[30] A. Sloman. What's vision for, and how does it work? From Marr (and earlier) to Gibson and Beyond, Sep 2011.
[31] A. Sloman and J. Chappell, 'The Altricial-Precocial Spectrum for Robots', in *Proceedings IJCAI'05*, pp. 1187–1192, Edinburgh, (2005). IJCAI.
[32] P. F. Strawson, *Individuals: An essay in descriptive metaphysics*, Methuen, London, 1959.
[33] A. M. Turing, 'The Chemical Basis Of Morphogenesis', *Phil. Trans. R. Soc. London B 237*, **237**, 37–72, (1952).
[34] A A S Weir, J Chappell, and A Kacelnik, 'Shaping of hooks in New Caledonian crows', *Science*, **297**, 981, (2002).
[35] I.P. Wright, *Emotional agents*, Ph.D. dissertation, School of Computer Science, The University of Birmingham, 1977. http://www.cs.bham.ac.uk/research/cogaff/.
[36] L. A. Zadeh, 'A New Direction in AI: Toward a Computational Theory of Perceptions', *AI Magazine*, **22**(1), 73–84, (2001).