SONIFICATION AND INFORMATION

CONCEPTS, INSTRUMENTS AND TECHNIQUES

David Worrall

March 2009

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

University of Canberra

CERTIFICATE OF AUTHORSHIP OF THESIS

Except where clearly acknowledged in footnotes, quotations and the bibliography, I certify that I am the sole author of the thesis submitted today entitled

Sonification: and Information: Concepts, Instruments and Techniques

I further certify that, to the best of my knowledge, the thesis contains no material previously published or written by another person except where due reference is made in the text of the thesis. The material in the thesis has not been the basis of an award of any other degree or diploma except where due reference is made in the text of the thesis. The thesis complies with University requirements for a thesis as set out in http://www.canberra.edu.au/secretariat/goldbook/forms/thesisrqmt.pdf

....

Signature of Candidate

....

Signature of chair of the supervisory panel

....

Date

DECLARATION

Some material in this thesis has previously been made public. Parts of Chapter 1 are awaiting publication in 2009. Parts of Chapter 4 appeared in papers for The International Conference for Auditory Display 2007 and The Inaugural International Conference on Music Communication Science 2007. Parts of Chapter 5 are to appear in a paper for the International Conference for Auditory Display 2009. In accordance with general software engineering practices, all of the software described in this thesis utilises tools developed by other people, too numerous to mention. References to the major tools used are clearly indicated when they are first referred to in the text.

The cover image is a composite photograph of two musical serpents. It was created by the author as a logo for *SoniPy*, the sonification software outlined in this thesis. First appearing in Europe in 1590, the serpent is an amalgamation of tuba, trombone, bassoon, French horn and oboe. Played softly, it has a firm yet mellow timbre, at medium volume it produces a robust sound and when played loudly it can produce unpleasant noises reminiscent of large animals in distress (Schmidt 1997).

Abstract

This thesis is a study of sonification and information: what they are and how they relate to each other. The pragmatic purpose of the work is to support a new generation of software tools that are can play an active role in research and practice that involves understanding information structures found in potentially vary large multivariate datasets. The theoretical component of the work involves a review of the way the concept of *information* has changed through Western culture, from the Ancient Greeks to recent collaborations between cognitive science and the philosophy of mind, with a particular emphasis on the phenomenology of immanent abstractions and how they might be supported and enhanced using sonification techniques. A new software framework is presented, together with several examples of its use in presenting sonifications of financial information, including that from a high-frequency securities-exchange trading-engine.

Acknowledgements

I take this opportunity to thank a number of people who have played a significant rôle in assisting this thesis come into being. Most immediately, my thesis supervisors Dr Mitchell Whitelaw and Professor Roger Dean, whose different approaches with respect to its motivation and content, and execution provided me the opportunity to develop more rigorous arguments than I might otherwise have done. Their attention to detail was most valuable.

I would like to thank Dr Stephen Barrass for his willingness to discuss the field of Auditory Display from a broad perspective, from his own initial findings in the mid-1990s through to his mature understanding of the field today. Michael Bylstra's assistance in testing a multitude of public domain software modules helped me to develop selection criteria for their usefulness for *SoniPy*, and his willingness to explore the intricacies of software library wrappers will, it is hoped, make a fuller realisation of the project in the public domain more expeditious.

The University of Canberra provided financial support, for which I thank them, as did the Capital Markets Cooperative Research Centre. Professor Michael Aitkin, provided a pragmatic but philosophically informed finance voice as well as positive feedback when the potential of the techniques demonstrated became apparent. Particular thanks go to Associate Professor Graham Partington, for his encouragement and support of what must have seen a very left–of–field proposal when I first suggested it to the CMCRC.

The opportunity to spend a concentrated period of time to develop this thesis came after twenty-five years of composition, teaching, research and public performance in electroacoustic and experimental music and the electronic arts more generally. I owe a debt of gratitude to the colleagues and students I was privileged to lead at the Australian Centre for the Arts and Technology at the Australian National University throughout the 1990s, for their enthusiasm for learning and their willingness to question. A special thanks goes to my colleague, collaborator and co-conspirator, the animator Stuart Ramsden. Our work together, and in particular the Design Structures course we taught, fortified the creative and technical foundations on which this thesis was built.

Last, but by no means least, there are my family and friends. You all know who you are. No words suffice. Simply, a heartfelt "thanks".

TABLE OF CONTENTS

TABLE OF FIGURES

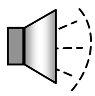
TABLE OF TABLES

TABLE OF AUDIO

THE ACCOMPANYING DVD

Audio Examples

Audio examples are essential for a full understanding of the sonification techniques discussed in the text of chapter 5. Sound examples are indicated in the outside margin of the text with a loudspeaker icon, as illustrated on the right of this text. Each of the examples folders on the disk has README textfile that summarises the files in that folder.



Some of these examples require specific sampling rates not compatible with audio CD format, so, unfortunately it is not possible to play it on a standard CD player.

The audiofiles are playable using *SoundHack*, freely available for the Macintosh OS from http://www.soundhack.com/freeware.php
Another alternative is *play* a well crafted command-line tool, available from http://www.hieper.nl/html/software.html

Whenever suitable, the csound source files (.csd) used to render the examples are provided.

Code listings

The code listings provided in Appendix 4 and referred to in the text of Chapter 5. are also available .py text files on the disk

The *SoniPy* website

The *SoniPy* website is subject to ongoing modification. A reasonably recent version of this site is provided in the www_sonipy folder. The current version can be accessed at http://www.sonification.com.au/sonipy/

Chapter 1

PROLOGUE

> [S]tructure, which is the division of the whole into parts; method, which is the note-to-note procedure; form, which is the expressive content, the morphology of the continuity; and materials, the sounds and silences of the composition...(Cage 1961/1967: 36).

1.1 Motivations for the work

This thesis has its early origins in an attempt to explain a phenomenon frequently experienced when composing algorithmic music with computers. At certain times in the process, usually towards the end of a major section or the work as a whole, the algorithms are set aside and an 'adjusting' or 'tuning' is undertaken 'by ear' that might involve experimenting with rescaling or perhaps re-quantising the pitch gamut, adjusting rhythmic hierarchies, limiting or compressing the audio bandwidth, reducing reverberation in the tenor register, and so on. All these high-level 'global' actions are performed more–or–less intuitively, until the work *gels* and the internal acoustic data representations begin to function as part of a cohesive whole; in which the structure becomes secondary to the form–the morphology of the continuity. That is, an identifiable unfolding continuity.

Many questions arise in thinking about this process, which, if generalised, is not confined to computer music, or even just to music. What are the underlying principles that inform this practice and can some general methodology be drawn from them? Perhaps a new kind of music theory is needed that does not confine musical information to that defined in current didactic texts, both old and new, or in the results of the reductionist practices of laboratory-bound psychoacoustic research, interesting though they are. The embodiment is a somewhat delicate procedure: taken too far, the result is auditory mush[1]. Clearly a balance has to be orchestrated that creates a sense of cohesion but that does not blur the articulation of structural

1 A soft or soggy mass, but also, an expression that is excessively sweet and sentimental.

features. Musical orchestration is often, though not always, concerned with the mixing of individual instrumental timbres into rich, cohesive complexes; and its principles are well documented and taught in music schools everywhere. While, in his ground-breaking overview, Albert Bregman (after Helmholtz) described the basic dimensions of analytic and synthetic listening in terms of auditory stream integration and segmentation (Bregman 1994: 395-453), there is yet to be written a generalised exposition appropriate for many sonification tasks: how to synthesise auditory cohesion while maintaining a clear articulation of the separate components.

For the purpose of the current research, it was decided to put aside those parts of the process that one could easily identify as stylistic, so as to concentrate on understanding the synthesis of perceptions that afford the transfer of information structures at the expense of the cultural–in full cognisance of the dangers of such dualisms. This does not imply that the style is forgotten, simply faded into the background so as to simplify the problem at hand. In fact, on playing examples of the capital market parameter-mapping sonifications discussed in Chapter 5, a composer colleague asked, "So, why do these sonifications sound so like computer music?"

Before introducing the content of each chapter individually, a broad summary of the context of the thesis as a whole may be beneficial. To date, the most common approach to sonifying multivariate datasets has been to apply a technique often used in computer-music composition, namely, to map data dimensions to acoustic or psychoacoustic parameters, in the hope that the information content of the data will be "revealed". However, it is now recognized that such an approach has not produced the sorts of results required of sonification, namely, clear and reliable perceptions of the information. The dilemma has become known colloquially in the field as "the mapping problem". The simplest explanation for the failure of the method is related to the non-orthogonal or co-dependent nature of aural perception as it is usually parameterized. For example, under certain circumstances, an increase in loudness is also perceived as a rise in pitch. The most common "solution" when using this technique is to test empirically which of a number of fine-tunings or "tweaks" of parameter space mappings is the least problematic; perhaps in the hope that eventually, over time, a generalised model may become evident.

Rather than proffering such a generalised solution supported by extensive empirical evidence, a much less ambitious aim of the research reported in the early chapters of this thesis is to try to better define the nature of the problem, for only then will the demands on the computational tools necessary to develop such solutions be understood. A major difficulty, and one that is frequently elided, is the distinction between the concepts of information and data. In fact, the frequently used expression "data sonification" promotes that elision and in doing so, implicitly supports the idea that information can automatically "pop-out" of a sonification once an optimal parameter-mapping of the dataset is found. This thesis argues that an understanding of the historically volatile nature of the differences between data, data-embedded information, "sense data" and perception can explain why such an expectation is unrealistic; that the early attempts by phenomenologists to define such purely mental constructs leads to a tautological reduction to Platonic Ideals and all the difficulties that they imply; that the search for these mental contructs is another example of the Cartesian disembodiment 'trap' and that there is still no known basis for the reliably robust formation of such abstract mental structures.

However, I go on to argue, that the work of Polanyi and others on tacit and embodied knowledge may prove a fruitful path to explore, particularly as this approach is currently being pursued by interdisciplinary teams of philosophers and cognitive scientists, following the generally recognised failure of abstract computational models to solve "the hard problem" in machine learning research. An argument is thus advanced that the cognitive stability of aural structures such as melodies, which were of intense interest to the early phenomenologists because they are examples of apparently abstract mental structures, may be related to their origins in body actions; of speaking, singing and playing. The implication of this is that if software is to be capable of contributing to the translation of information in data into more reliable perceptual objects, it may have to be capable of simulating embodiment; a different, perhaps more difficult task, than the production of sound from acoustic or psychoacoustic parameter-mappings. With that task in mind, the thesis then proposes and reports on the development and testing of a software framework, called *SoniPy*, which affords such research.

Chapter overview

The thesis can be broadly divided into two themes. Chapters 1–3 provide an historical overview and theoretical context, Chapter 4 is a short, somewhat speculative link to the more practically–oriented design of a software framework powerful enough to create and undertake research into sonification and information together with some technical experiments that test parts of the framework on the sonification of capital market trading data.

1.1.2 Chapter 2

Empirical research in the synthesis of auditory designs for the pragmatic communication of non-musical and non-speech acoustic representations began to emerge in the 1990. Chapter 2 reviews the field as a whole, first by examining some descriptive definitions of sonification and suggesting some small improvements. The use of discrete sounds for alerts and alarms present designers primarily with differentiation problems: between the sounds themselves and between the sounds and the environment in which they function. Though related in subtle ways, these discrete audifications do not address an opposite issue (the "mapping problem"): how can data *relations* be represented acoustically for interpretation by listeners, for the purpose of increasing their knowledge of the source from which the data was acquired. That problem can be recast as the task of creating mental 'objects' for active contemplation, rather than how to correctly elicit a timely response to a well-differentiated auditory stimulus. Somewhat between these two is the task of continuous monitoring of production and environmental processes, and so forth.

An informal browse through a number of other theses in the field was one reason behind the decision not to include another cursory overview of the physics or psychophysics of sound in this thesis. Another was the ability to reference personal material previously published that more fully covers the material. However, the most important reason was a sense that the discussion needed to move on. The physics or psychophysics is important from an analytic perspective but for it to be useful for synthesis, it needs to be in the form of inverse filters, such as that for Fletcher-Munson as informally applied in Chapter 6 (§6.9) of this thesis. There is some peripheral work currently being undertaken (Cabrera Ferguson and Schubert

2007) and it would be useful if were to be generalised. The concern of this thesis, however, was to look further forward, to try to find a basis for better mental instantiations of multivariate datasets using sonification.

The term *sonification* has passed ('been appropriated' would probably be a more accurate description) into creative practice fora, possibly in order to avoid some of the associations the term *composition* engenders in funding bodies and the public at large. This thesis attempts to maintain the distinction, not in order to promote territorial disputes, but because having such a distinction makes it easier to compare and contrast motivations and results. Research in music can be very beneficial to research in sonification, however one of the disciplines of the latter, or so it seems this author, is the need to use music's tools and findings without being seduced by the aims and functions of music itself. So Chapter 2 ends with a comparison of data sonification and data music, and iterates the principle reason why there is a frequently–expressed need for a new generation of software for sonification; tools that integrate flexible sound-synthesis engines with those for data acquisition, analysis and manipulation, in ways that afford both experiments in cognition and lucid, interpretive *soniculations* (that is, sonic articulations).

1.1.3 Chapter 3 and Appendix 1

A goal of data sonification is to use sounds to aid listeners' acquisition of knowledge about a phenomenon, so it is logical to suppose that an understanding of the essential characteristics of that acquisition process, the extraction of information, may influence the design of the software used to compose and render sonifications. Such software will need to afford the exploration of the cognitive and psychological aspects of the perception of mental objects formed through the sonification of datasets that have no analogue in the material world, and the purpose of Chapter 3 is to explore the epistemological dimensions of that task.

It is rare to find references to philosophical inquiry when reading scientific literature that reports on the results of empirical experimentation–a trend probably with its origins in the Gestaltist's desire to separate their experimental motivations from those of the 'pure' psychologists and understandable because a discussion of the validity or otherwise of the empirical techniques is more appropriate in philosophy of science arenas. Discussion of a philosophical nature is more common in fundamental science, especially on either side of a paradigm shift such as occurred in quantum physics and is currently occurring in cognitivism. Currently, sonification research is hardly settled and there are references in the literature, some, unfortunately, not very informed. Whilst not as dire, the same can be said for much published work on new media. The empiricist John Locke (1632-1704) seems particularly favoured when some degree of philosophical respectability is called for, probably, apart from his empirical leanings, because he wrote in English. Reference to Hume's refutation of some of Locke's work is as rare as Immanuel Kant's resolution. Occasional mentions of the intention to write an overview was met with enthusiasm so, having some previous experience in the field it was decided to attempt to lay out the philosophical framework as succinctly as possible.

Clearly, a complete philosophical and psychological overview is outside the scope of the current thesis, however if sonification software is to access complexly structured data, support informational enquiry, presentation and retention, in a perceptually and cognitively efficient manner, a thorough understanding of the dimensions of the problem and the contribution of others from the past, should be empowering. The approach is to use primary sources (at least English translations of them) as much as possible in order to maintain the flavour of the original enquiry, and to use sound-related examples when examples are called for–something that the original texts rarely do, and secondary sources, almost never.

The chapter begins with a discussion of some meanings of the term information, and Appendix 1, a pragmatic summary of different modes of knowledge acquisition, functions to support these definitions. Considered in this way, the trans*form*ation of in*form*ation into knowledge is an internal process–whether to an individual, a group or a community, and while there may be sonification techniques to enhance that such processes[2], they lie outside the scope of the current thesis. Most of the contents of Appendix 1 are widely understood, however it was included because such an inclusive yet succinct summary was not found elsewhere. In addition to the various forms of inference, and embodied knowledge, the inclusion of Reliablism, so apt an epistemological description of current scholarly practice, will add a less–well–known flavour.

---

[2] Such as those used to enhance learning by entraining brain beta frequencies.

The relationship between our sensing of a variegated world and the mental models we use to represent it has been a major theme in Western philosophy and the remainder of Chapter 3 provides a reasonably thorough introduction.

1.1.4 Chapter 4

Chapter 4 is a short *intermezzo* between the theoretical orientation of the epistemology of Chapter 3 and the more practical orientation of software in chapter 5. Its purpose is to address, in a discursive way, the question: “If sonification software is to meet and even anticipate the needs of sonifiers in the future, what sorts of problems will it be required to address?” In some ways Chapter 4 is the underling enquiry of the thesis as a whole but a way was not found to express it properly without reference to the epistemology of Chapter 3, which points very strongly to the inadequacy of a purely mind-oriented solution to the problem of how to sustain abstract immanent phenomenal objects of multivariate datasets for cognitive enquiry and reflection. If this inadequacy is a reality it is probably more effective if it is considered a ‘design feature’ of the human condition, rather than as a ‘bug’.

Any attempt to define the basis on which a paradigm that exploited this ‘bug’ that could be constructed for the translation of information contained in datsets into mental models that were more sustainable than those used in Parameter Mapping sonifications would require more empirical research than was appropriate in the context of the current work. The two closest models known to work, for different reasons and with different types of information are speech and music. These are powerful models. However, to function as a medium of information transfer, speech requires language and that requires community adoption. Esperanto does not seem to have been accepted, and the modernised talking drum, Morse code, probably could not sustain a ‘come-back’, enchanting though it would be. The seeming universality of music and the increasing acceptance, as evidenced in the popularity of ‘world music’ (Taylor 1997: 1), of broader range of musical paradigms than considered by Deryck (1959), are positive aspects. Because of its experimental nature, music can lead the way. The serialists exposed cognitive limitations to all but the highly trained, such as the lack of recognition of outside time temporal transformations (retrogradation for example), and imitations in the recognition of pitch inversion.

Chapter 4 begins by returning to the Greeks again; for perspective and for inspiration. While music has its limitations, Chapter 4 outlines a somewhat speculative case for the need for software to be able to address issue of embodiment, whatever that may mean as a possible way forward. While this is not taken-up in any major way in the remainder of the thesis, it was an underlying motivation for developing the framework approach to software design that is discussed in Chapter 5, as mentioned earlier.

1.1.5 Chapter 5

The need for better software tools for sonification was highlighted in the *Sonification Report*'s comprehensive review of the field (Kramer et al. 1997). Their review included some general proposals for adapting sound synthesis software to the needs of sonification research. However, over a decade later, it is evident that the current demands being made of sonifications, especially those with large or multidimensional datasets, are much greater than the capabilities afforded by music the composition and sound synthesis software that is currently in use. Chapter 5 addresses some of the technical reasons this problem exists and discusses some major contributions towards achieving the *Report*'s proposals and current sonification demands.

The chapter outlines a broader and more robust framework model that can integrate other software developer's prior work and expertise, including that which has no direct connection to sonification, by using a public-domain community-development approach. Named *SoniPy*, it integrates various already existing independent components such as those for data acquisition, storage and analysis, cognitive and perceptual mappings as well as sound synthesis and control, by encapsulating them, or control of them, as *Python* Modules within the framework. In contemporary computer science the term *framework* has a specific meaning, and that is the meaning applied here.

A website has been created that outlines the various components of *SoniPy*. It functions as a first port–of–call for sonification-related activities using the *Python* programming language, and provides an introduction to modules that have passed selection criteria testing for their use in undertaking various sonification-related tasks. While the site (at http://www.sonification.com.au/sonipy) is continually evolving at, a version is available on disk for off-line browsing.

1.1.6 Chapter 6 and Appendices 2 & 3 & 4

Chapter 6 details some experiments with capital markets data using the *SoniPy* framework. The sonification techniques employed include a new approach to the direct audification, using twenty-two years of an historical dataset, and the psychoacoustic parameter-mapping of information 'mined' from a high-frequency trading engine data. The latter work required the development of considerable data-handling capabilities in order to test the initial hypotheses. The practical experiments are preceded by a literature review of audification and prior work undertaken by others in sonifying economic, market and trading data, together with an overview of how a generic public market operates.

The sound rendering models used are as simple as possible for two reasons. Firstly, the aim is clarity not comfort, and secondly, experience has taught that hundreds of hours can be consumed trying to adjust one or more of a multitude of parameters in order to approximate a fuzzy target, only to find that the mind has adapted to the extent that what began as a clarinet-like sound ends up sounding more like a French horn, but in the interim the mind has convinced itself that it does in fact, sound more like a clarinet than it did before the whole exercise was begun.

Appendix 2 provides a succinct 'refresher' outline of some key statistical principles to make comprehension of the main text easier. Appendix 3 is the metadata specification of the high-frequency trading engine dataset and Appendix 4 contains various code listings, as detailed in the text; all of which are available, together with the sound examples, on the accompanying disk.

1.1.7 Chapter 7

Chapter 7 summarises the principal ideas of the thesis, draws some conclusions on what worked well, what not so well, and makes some suggestions for future similarly-motivated work as well as that which can build on the work undertaken here.

Chapter 2

AN OVERVIEW OF SONIFICATION

The aim of this overview of data sonification is to provide the reader with an understanding of the origins and conceptual issues involved in this young, interdisciplinary and quickly evolving discipline. It begins by summarising different ways sonification has been defined, the types and classifications of data that it attempts to represent with sound, and how these representations perform under the pressure of real-world usage. The need for better tools for data sonification is raised and this leads to discussion of the value of collaborative research towards this end and a reflection on the relationship between music sound and science.

There are numerous reasons why sound might be the preferred representational medium for information in particular circumstances, including the known superiority of the hearing sense to discriminate certain kinds of structures. For example, it is easy to personally verify that a purely visual comparison of spatially separated representations requires high levels of concentration and is thus very prone to error, especially over extended periods of time, while listening to the reading of such representations is much easier. The presence of auditing (*hearing of accounts* from the Latin *auditus*) has been inferred from records of Mesopotamian civilizations going back as early as 3500 BCE. To ensure that the Pharaoh was not being cheated, auditors compared the *soundness* of strictly independently scribed accounts of commodities moving in, out and remaining in warehouses (Boyd 1905). In the alternating intoning of such lists, differences can be easily identified aurally. A faster and more secure method that eliminates any ‘copy-cat’ syndrome in such alternation, is to have the scribes read the records simultaneously–a type of modulation differencing technique. While we have no evidence that these techniques were practiced in ancient times, such a suggestion does not seem unreasonable, and would represent possibly the earliest form of data sonification.

2.1 Defining and classifying sonification

A primary distinction can be made between so called *audifications*, which entail the direct amplification or filtering of existing sounds, such as is accomplished with the esophageal stethoscope, and the use of sound to convey inherently silent abstractions such as *variables* and *data*. The non-chemical[1] definition of *sonification* has evolved over the last fifteen years as its use in auditory displays has developed. For the purpose of discussing multivariate data mappings, Bly (1994: 406) described sonification as *audio representation of multivariate data*. The sonification of univariate data is also possible, and thus Scaletti proposed a more formal working definition for her investigation of auditory data representation, as

> a mapping of numerically represented relations in some domain under study to relations in an acoustic domain for the purposes of interpreting, understanding, or communicating relations in the domain under study (Scaletti 1994: 224).

In order to differentiate sonification from other uses of sound, Scaletti explicitly draws attention to two parts of her definition: a *technique* (mapping numerical data to sound) and an *intent* (to understand or communicate something about the world). Barrass, reworking Scaletti's definition, *en route* to a definition of auditory information design "the design of sounds to support an information processing activity", emphasises the idea of information (the content) over data (the medium):

> a mapping of information to perceptual relations in the acoustic domain to meet the information requirements of an information processing activity (Barrass 1997: 29-30).

Both Scaletti's and Barrass' definitions can be read to mean both the process of representing, and the resulting sonic object. The *Sonification Report* (Kramer et al. 1999) was a major effort at summarising the field to date. Its focus is on sonification as a process:

---

1 In biology, the term simply means *the production of sound waves* and is used to refer to a technique known as *sonication* [sic], in which *a suspension of cells is exposed to the disruptive effect of the energy of high-frequency sound waves* (Online Medical Dictionary: *sonification*) and (Biology Online: *sonication*). In chemistry, it refers to *the use of (often ultra-) sound waves to increase the rate of a reaction or to prepare vesicles in mixtures of surfactants and water* (ChemiCool: *sonication*). The term is also used for a process similar to the chemical one described that 'resonates' subsurface geological structures for oil extraction, See, for example, http://www.admin.mtu.edu/urel/news/media_relations/3/.

> The use of non-speech audio to convey information. More specifically, sonification is the transformation of data relations into perceived relations in an acoustic signal for the purposes of facilitating communication or interpretation.

The first sentence of this 1999 definition appears to be the most succinct and widely used. Speech audio is specifically excluded, presumably so as to discriminate sonification techniques from speech-related practices, such as text-to-speech software. While speech research is an extensive research field in and of itself, there is no reason why speech audio should *necessarily* be excluded from the definition. As Hermann argued (2002: 23), speech does have attributes which, if data-driven, could be useful for sonification purposes, and some research has suggested speech-audio displays could be used to convey non-verbal information upon which people can make useful decisions (Nesbitt and Barrass 2002). While speech and non-speech sounds share such identifiable auditory characteristics as pitch, rhythm, articulation, and rugosity, as well as some larger gestalts such as phrase and prosody, the ability to simultaneously listen to music and talk or read without confusion is well known and easily demonstrated. This is supported by cognition research that finds that the auditory cortices in the two hemispheres of the brain are relatively specialised enough to be able to exploit the temporal and spectral differences between speech and musical sounds (Zatorre, Belin and Penhume 2002). Recent research found that speech that is temporally compressed until incomprehensible as speech could significantly improve menu navigation in PDA devices (Walker, Nance and Lindsay 2006). All these reasons emphasise the importance of not excluding speech-like sounds from the definition of sonification without a thorough evaluation of the information bandwidth that would be lost by doing so.

While *the representation of data relations in sound relations* (Anderson, Sanderson and Norris 2002) is likely the most succinct definition of sonification, it avoids the intent referenced in the 1999 definition: *for the purposes of facilitating communication or interpretation.* Nor does it quite capture distinctions between, for example, data sonification and data-driven music composition. While a purpose of music is the expression of musical knowledge and broader cultural considerations, whatever they may be, between composers, performers and listeners, the purpose of sonification, as the term was originally used, is to represent data in sound in such ways that structural characteristics of the data become apparent to a listener. This distinction is emphasised by the new portmanteau expression *soniculation* (from *sonic articulation).* It is used occasionally in this thesis when such a distinction is important, without suggesting it be promoted to a term, as that would probably only add to the general confusion. In the early literature (Kramer 1994b), the term *sonification* is used as a shortened form of *data sonification* as a sub-category of *auditory display*, but the distinction, perhaps unfortunately, seems to have subsequently disappeared. All three terms are now used both nominally and verbally, as are the expressions *to sonify* and, though less common, *to audify*.

Terminology such as *auditory display* and *sonification* are best considered descriptions, rather than definitions within a strict taxonomy, because their meanings, driven by the need for finer distinctions and qualifications, have too great an inertia to be arrested by any current desires for semantic tidiness. So, this thesis uses the expression *data sonification* with the following interpretation, to clarify the object of the action and to lift the "non-speech audio" restriction:

> *Data sonification* is the acoustic representation of data for relational interpretation by listeners, for the purpose of increasing their knowledge of the source from which the data was acquired.

This, and other descriptions of data sonification, preserves the 'mystery' of the implications of the phrase *for relational interpretation by listeners*. Further, it does not intimate the fact that it is often the data *relations* that are sonified rather than the data itself. Relations are abstractions of, or from, the data and the sonification of them more explicitly implies sonifier intent. This thesis uses the expression *information sonification* to express that distinction. The term *information* adds a complexity of its own; one that has changed over time, as discussed more fully in the next chapter. When the distinction is not important to the argument, this thesis simply uses *sonification,* without a qualifier.

2.2 Classifying sonifications

Sonifications can be classified in a number of ways: distribution technology (medical, public arena, interactive) intended audience (anaesthetists, stock brokers, the visually impaired), data source (electrocardiograms, securities markets, hearing aids) data type (analog, digital, real-time, spatial, temporal) and so on. Twenty or so years since concerted research began, data sources and target applications for sonification have now become too numerous to review them in this context. They are also the easiest to locate using keyword searches on the world wide web.[2] For the current purpose, we partially follow Kramer (1994:21-29) in using a description of the type of representation that is used to present information to the auditory system.

There are two broad representational distinctions at either end of an analogic–symbolic continuum. Analogy[3] is a high-level cognitive process of transferring information from a particular subject (the analogue or Source) to another particular subject (the Target) or of establishing the relation between the Source and the Targets themselves. The purpose of analogic representation is to make the structure of the information better suited or simpler for the Target than it was in its original form. Analogic representation is more connotative than denotative. A good example of analogic sonification is a Geiger counter that produces clicks in a loudspeaker at a rate proportional to the strength of radiation in its vicinity.

By contrast, a symbolic[4] representation is a categorical sign for what is being represented. It is more denotative than connotative and thus more abstracted from the source than an analogic representation. In symbolic representation, source data is aggregated and assigned to the elements of a schema (symbols) according to a set of rules that, as a result, implicitly makes the distinction between data and information. These symbols are then displayed to the Target, who processes them according to their knowledge of the schema used. By illustration, consider a stream of noise emanating from a source. On listening to the stream, a Target recognises that some segments of it are symbols in one schema that they know, some sounds are symbols in another known schema and there are still other sounds for which they do not have a schema (other than the 'everything–I–do–not–have–a–schema–for' schema). An example would be a sound stream of English speech, spoken by

---

2 For example, in June 2008, an internet search of the phrase *stockmarket sonification* revealed over six thousand links (c.f. fifty million for "stockmarket").

3 From Gk. *analogia* 'proportion', from *ana–* 'upon, according to' + *logos* 'ratio', also 'word, speech, reckoning'. [OLED]

4 From Gk. *symbolon* syn– 'together' + stem of *ballein* 'to throw', evolves from the ''throwing things together' to 'contrasting' to 'comparing' to 'token used in comparisons to determine if something is genuine'. Hence, the 'outward sign' of something. [OLED]

someone with influenza, who, at some point says something to a dog in a language that the Target doesn't recognise. The semantic meaning of the English is understood by the Target, as are the influenza symptoms, but the dog-directed sound contains no recognisable information: it is just data. Two important features of this symbolic process are (a) a variety of display schemata can be used to symbolically represent the same information, or different information by different discrete–element–aggregation of the data, and (b) the relationships between the symbols themselves do not reflect relationships between what is being represented. For example, there is no relationship between the ideas represented by the words *loss*, *boss* and *lass*, nor the numerals of the number 1055, even though the structures of the symbols are similar.

In data sonification, for which the Source is data and the Targets are listeners, the types of representations a sonifier might employ is dependent on the structure of the data, the kind of information needing to be extracted from it and how easily (efficiently and effectively) that information can be processed by the listeners' hearing systems: physiological, perceptual, cognitive and memoric. This description emphasises the importance, for sonifiers, of developing a thorough working knowledge of the ways hearing systems organise acoustic waves into discrete events using Targets' ability to perform such fundamental perceptual and cognitive processes as "auditory stream segregation and auditory stream integration" (Bregman 1994). Ongoing research in the fields of psychophysics and experimental psychology is directed at the empirical investigation of these processes. As is to be expected, most of this work is directed at understanding the segregation and integration of simultaneous, or near–simultaneous, components constituting auditory streams, rather than the larger or more complex scenarios of real-world situations.

2.3 Types of data representation

The data representation types described below, loosely based on de Campo's groupings (2007) are discrete, continuous and interactive. These types can be located at different points on the analogic-symbolic continuum discussed above. Discrete representations mostly function symbolically, continuous representations mostly analogically, and interactive representations, mostly analogically but in a discontinuous manner. In practice, a particular sonification may use one or more of these types, even simultaneously, to satisfy the needs of the specific application.

2.3.1 Discrete data representations

Discrete data representations are representations in which every data point (datum) is sonified with an individual auditory event. Discrete data representations are strongly symbolic and can be used as signifiers when there is no single preferred ordering of the datum. User-defined subsets of the data can be formed at will and randomly iterated over, much as a visual scene can be scanned in whatever way the viewer chooses. This flexibility of access to the data assists the user to build up a mental representation of the data space and the position of each data point in it. Whether or not a representation *appears* discrete or continuous will often depend on the interplay between data scaling and perceptual thresholds. A sequence of discrete frequencies may appear as a continuous glissando if they are closely clustered, or the close examination a continuous representation may cause it to appear discrete when it is time-stretched. Such definitional conundrums emphasise that these explanations are more descriptive than strictly taxonomic. Another example is the way in which auditory beacons (Kramer 1994; §2.3.2.1) function within an otherwise continuous data representation adds further emphasis to this point. However, when individual isolated sounds are initiated as environmental events, such as in HCI, the distinction is clearer and it is these we consider next.

2.3.1.1 Auditory warnings: Alarms and Alerts

Many animals use sounds of various kinds to warn those in their vicinity to the presence of others. Alarm signals, known in animal communication sciences as anti-predator adaptations, can be species specific, and are particularly effective when imminent danger is sensed. Boisterous alarms tend to provoke a *fright or flight* response, which, in ongoing monitoring situations, is perhaps not subtle enough: the ensuing panic is more likely to lead to a silencing of the alarm, or when that isn't possible, adaptation by cognitive filtering.

The term *alert* indicates the possibility of a more considered, attentive approach, but the distinction is by no means common practice. Patterson (1982 and 1989) experimented on alerts for aircraft navigation and produced a set of guidelines covering all aspects of warning design. The warning signals he suggests are meant to be instantly recognisable by listeners and use quite low-level intensities and slower onsets/offsets times to avoid startling the pilot. The auditory warning must impress itself upon the consciousness of the operator, yet not be so insistent that it dominates cognitive function. Patterson categorized warning signals into three priority levels: *emergency*, *abnormal* and *advisory*.

Extant guidelines for ergonomic and public safety applications of auditory warnings (McCormick and Sanders 1984) can be adapted for both general sonification purposes (Warin 2002) and more specific Human Computer Interaction (HCI) tasks, the initial purpose for which research into auditory icons and earcons was undertaken.

2.3.1.2 Auditory icons

The first detailed studies of the use of the sound capabilities of the newly emerged personal computers in the 1980s was as interface tools to operations of the machine itself. The desktop metaphor, first widely available on the Apple Macintosh computers (c. 1984), was recognized as a paradigm shift and quickly adopted by all personal computer manufacturers. In summarizing his work going back to 1988, Gaver (1994) outlines how sounds that were modelled after real world acoustics and mapped to computer events, could be used to enhance this desktop metaphor. He called these sounds *auditory icons*, for conceptual compatibility with Apple's computer Desktop Icon and Finder metaphors (folders, rubbish bin, menus etc). Gaver's reasons for using real-world sounds were based on an interpretation of James Gibson's theories of direct perception in which information about events in the world is perceived directly through patterns of energy, and understood innately, rather than through inferential representational structures in the mind. (Gibson 1966).

Building on the earlier work, Gaver developed, with others, various Finder-related tools: SonicFinder, Alternative Reality Kits (ARKs) and Environmental Audio Reminders (EARs). They extend auditory icons from sampled recordings of everyday sounds to synthesized abstract models of them that could more easily be manipulated (by pitch change, filtering etc) to represent qualitative aspects of the objects being represented. He also suggests extending the range of applicability to remote machines, such as to indicate whether a printer in another room is functioning and the frequency of the page output.

For blind users, Mynatt developed a library of auditory icons, called *Mercator*, to complement the existing widget hierarchy of an existing graphical user interface. She reasoned that "auditory icons offer the most promise for producing discriminable, intuitive mappings" on the same 'direct perception' basis as Gaver and thus argues that "[w]hat this realization means to an interface designer is that we can use sounds to remind the user of an object or concept from the user's everyday world" (Mynatt 1994).

Real-world sounds easily convey simple messages that are cognitively processed without much learning if the sounds are easy to identify. Yet sound in an interface that "seems cute and clever at first may grow tiresome after a few exposures" (Gaver and Smith 1990). In evaluating *Mercator*, Mynatt also observed a number of limitations, including the difficulty users had in identifying cues, especially those of short duration; the need for high (spectral) quality of sound and the need to evaluate auditory icons in a complete set for maximum dissimilarity while maintaining ease of identification. Both Gaver and Mynatt raise the question of whether or not the concept of auditory icons breaks down with virtual objects that do not have a clear counterpoint in the everyday world. Such objects are known phenomenologically as *mental* or *immanent* objects (Husserl 1927)[5]. Mynatt concludes, at the same time as cautioning that the participant sample size was too small for formal statistical analysis, that while non-sighted users could successfully navigate the system, "the overall test results indicated that the goal of designing intuitive auditory icons was not satisfied in the prototype interface." There were suggestions, by Blattner, Sumikawa and Greenberg (1989), that the problems in memorizing a large number of such icons was due to there being no structural links between them. In addition, auditory icons can be homonymic - different physical events give rise to similar sounds. Later research (Sikora, Roberts and Murray 1995; Roberts and Sikora

5 This issue is addressed more extensively in the next chapter.

1997; Bussemakers and de Haan 2000) shows that, when compared with earcons and visual icons, test participants found that auditory icons were the least pleasant and least appropriate for use in computer interfaces.

2.3.1.3 Earcons

One alternative to using real-world sounds to reflect various computer activities, is to use structured sequences of synthetic tones called *earcons*, which are "non-verbal audio messages that are used in the computer/user interface to provide information to the user about some computer object, operation or interaction" (Blattner, Sumikawa and Greenberg 1989). Earcons are made by transforming a tone's psychophysical parameters–pitch, loudness, duration and timbre–into structured, non-verbal 'message' combinations. Blattner, Papp and Glinert. (1994) found the use of musical timbres proved more effective than simple tones, however gross differences were needed for effective distinction to occur.

Brewster, Wright and Edwards (1994) undertook a number of experiments investigating the comprehensibility of earcons under various parametric transformations. They created motifs for a set of simple operations, such as *open*, *close*, *file* and *program*. A compound earcon was then created that gave a sound for *open file* or *close program* by simply concatenating the two motifs. They experimentally tested the recall and recognition of different types of earcons and reported that 80% recall accuracy could be achieved with careful design of the sounds and that earcons "are better for presenting information than unstructured bursts of sound …and high levels of recognition can be achieved by careful use of pitch, rhythm and timbre. "

One of the most powerful features of earcons is that they can be combined to produce compound messages. So, symbolic sound elements (motifs) and their transformations can be used hierarchically to represent events or objects. A drawback, however, is that the deeper the structural hierarchy being represented, the longer a sequentially structured earcon takes to play, thus interfering with the speed at which a user can interact with the computer. Playing earcons more rapidly would risk recognition errors, though this could presumably be controlled for. Brewster, Wright and Edwards (1993) experimented with a different approach, namely to play the earcons at the same rate but present the information in parallel, taking the time of only a single earcon to do so. Their results indicated that parallel and serial earcons were recognised equally as well. This is an important finding because it indicated that cognitive processing of an auditory display for known informational content may provide greater immutability than psychoacoustic conflict might otherwise indicate[6]. In both cases, recognition rates improved with repeated usage, which is the case in most HCI applications. Interestingly, more *training* produced greater improvements for parallel earcons than serial ones, 90% recognition rates being easily achieved. Furthermore, given their symbolic structures are similar to the semantic aspects of music, the hypothesis that musicians would perform better than non-musicians was not verified: there were no differences in performance between the two groups, except that some of the non-musicians took slightly longer to learn the system. Results from several experimenters had confirmed that correct identification of earcons decreased markedly if they occurred concurrently. McGookin (2004) undertook a detailed investigation of this phenomenon and produced a set of design principles including the use of spatial separation and a 300 ms onset-time offset. His findings indicate that, even when using these guidelines, the accuracy of concurrent earcon identification still decreased from 90% to 30% when the number of earcons increased from one to four. However these guidelines did improve the accuracy with which individual earcons were identified. Overall, earcon studies indicate that they can be an effective means of communicating hierarchical structures but that the number of them that can be usefully identified is quite limited when they are used concurrently.

2.3.1.4 Speech noises

NoiseSpeech is made by digitally processing sounds so that they have some of the acoustic properties of speech (Dean 2005). It is made either by applying the formant structures of speech to noise or other sounds, or by distorting speech sounds such that they no longer form identifiable phoneme sequences. The 'hybrid' sounds that results from this process encapsulate some of the affective qualities of human speech, while removing the semantic content. Empirical experimental evidence suggests that most listeners cluster the sonic

---

[6] This suggestion was is tested in Experiment 5 in chapter 5.

characteristics of NoiseSpeech with speech rather than those musical instrument or environmental sounds (Dean and Bailes 2006; 2009).

Less abstracted than NoiseSpeech, *spearcons* (speech earcons) are spoken phrases that have been time-compressed until they are not recognizable as speech (Walker, Nance and Lindsay 2006). They were designed to enhance hierarchical menu navigation for mobile and screen-limited devices, in which they can be created automatically by converting menu item text (e.g., *Export File*) to synthetic speech via text-to-speech software. Keeping pitch invariant, this synthetic speech is then time-compressed, rendering it incomprehensible. Whilst each spearcon in an application is unique, phonetic similarity is maintained. For example, the initial phonemes are invariant in *Open File*, *Open Location…*, and *Open Address Book* as are the endings of *Open File*, *Export File*. The relative lengths of the text strings are maintained by their spearcons and this assists the listener to and learn and identify the mappings.

2.3.1.5 Discrete data representations compared

The postulation that, because of the connection to fundamental percepts, auditory icons should be easier to learn than earcons, was questioned by Lucas (1994) who found it took significantly less time to learn spoken messages than it did to learn either earcons or auditory icons, and unlike earcons and auditory icons, these spoken messages were consistently interpreted error-free. Further, he found no significant differences in the amount of time needed to learn the meanings associated with earcons or auditory icons. This result is at odds with a simple interpretation of ecological theory (Gibson 1966; 1979) and Ballas (1994) suggests that the activity of listening contains intermediate mental processes that take account of a listener's expectations and experience and the context in which the auditing occurs. One could speculate that, because the listener is also immersed in a real auditory scene as well as the computer's virtual one, a differentiation between virtual and real elements requires greater cognitive processing than if the two scenes were not superimposed. Bussemakers and de Haan (2000) undertook a comparative study of earcons and auditory icons in a multimedia environment and found that having sound with a visual task does not always lead to faster reaction times. Although reaction times are slower in the experiment with earcons, it seems that users are able to extract information from these sounds and use it. Furthermore, users find real-life sounds annoying when they hear them frequently.

Earcons are constructed from lexical elements that are ordered categorical symbols. So, while, like musical themes, their meaning is multiplicative under symmetry group transformation (transposition, retrogradation, inversion etc), the information needs to go through a decoding phase. As there is no standard syntax or lexicon, their meanings have to be learnt (Blattner, Sumikawa and Greenberg 1989), requiring a high initial cognitive load. Moreover, without absolute standardisation across all software and hardware, disarray rather than greater clarity seems the more likely outcome.

Palladino and Walker (2007) conducted a study comparing menu navigation performance with earcons, auditory icons, and spearcons. Their results indicate that faster and more accurate menu navigation was achieved with spearcons than with speech-only, hierarchical earcons and auditory icons (the slowest). They suggest that one reason for the speed and accuracy of spearcons is that, because they retain the relative lengths of their sources, these different lengths provide a "guide to the ear" while scanning down through a menu, just as the ragged right edge of items in a visual menu aids in visual searching. Because spearcons can be created and stored as part of a software upgrade or initialisation process, access times would be no longer than earcons and auditory icons. Furthermore, language or dialect-specific issues can be managed by the operating system's standard internationalisation procedures.

Since the mapping between spearcons and their menu item is non-arbitrary, there is less learning required than would be the case for a purely arbitrary mapping. Unlike earcon menus, spearcons menus can be re-arranged, sorted, and have items inserted or deleted, without changing the mapping of the various sounds to menu items. Spearcons may not be as effective at communicating their menu location as hierarchical earcons. However, spearcons would still provide more direct mappings between sound and menu item than earcons, and cover more content domains, more flexibly, than auditory icons.

Whilst auditory icons and earcons clearly have lexical properties (pitch, duration etc) they are used as auditory cues or messages; as signifiers of stable gestalts, i.e. as clear denoters of the presence of known, separable or discrete information states. The design of sets of auditory icons, earcons, spearcons etc is concerned with finding ways to effectively indicate these states under auditory segregation pressure. That is, the ability with which they remain immutable in human audition, in the presence of each other and within the perceptual environment in which they exist. These audio cues (or *messages*) function semantically in that they convey information about the state of the system of which they are a part and so can be used to reduce visual workload or function as a way of monitoring the system when visual or other means are not available or appropriate.

A weakness of the primarily ontological classification of discrete data representations as earcons, icons, etc, is that an emphasis on obtusely veridical denotations rather than often subtle, meaningful qualities, can have the effect of discouraging the user's active listening and so significantly discourage their engagement with the auditory display altogether. Such subtlety is more likely to be achieved by the cohesive use of careful mixtures of real-world audio–graphic and syntactic constructions whose symbolic power comes not from their concreteness or abstractness but from their intuitively understood affect. The design challenge is how to do this without recourse to distractional splendiferous phantasmagorias and in ways that support the user's need to segregate their computed and physical environments with a minimum cognitive load.

2.3.2 Continuous data representations

Continuous data representations treat data as analogically continuous. They rely on two preconditions: an equally–spaced metric in at least one dimension and sufficient data to afford a high enough sampling rate for aural interpolation between data points. Continuous data representations are most commonly used for exploring data in order to learn more about the system that produced it. Their applications range from monitoring the real-time operation of machines, capital–market trading, geographic and demographic features, weather and the environment, and so on, so as to discover new regularities and to assisting those with visual impairment to gain access to information normally presented graphically. There are three types of continuous data representation that, for consistency, are labelled *Parametric Mapping Sonification*, *Direct Data Audification* and *Homomorphic Modulation Sonification*.

2.3.2.1 Parametric mapping sonification

Parameter mapping is the most widely used sonification technique for representing high-dimensional data as sound. Parameter mapping sonifications are sometimes referred to as sonic scatter plots (Flowers, Buhman and Turnage 1997; Flowers 2005) or n^{th}–order parameter mappings (Scaletti 1994). Typically, data dimensions are mapped to sound parameters: either to physical (frequency, amplitude), psychophysical (pitch, loudness) or perceptually coherent complexes (timbre, rhythm). Parameter mapping sonifications can have both analogical and symbolic components. Analogic variations in the sound can result when mapping from a large data domain into a small perceptual range or when data is specifically mapped to acoustic modifiers such as frequency or amplitude modulators. Parametric mapping sonification is sometimes referred to as *multivariate data mapping*, in which multiple variables are mapped to a single sound. Scaletti describes one way of implementing it by "mapping of each component of a multidimensional data point to a coefficient of a polynomial and then using that polynomial as the transfer function for a sinusoidal input" (1994). Within an overall analogic mapping, symbolic representations such as auditory beacons (Kramer 1994) can be used to highlight features such as new maxima and minima, or absolute reference points in a sonification such as ticks to indicate the regular passing of time.

Parametric mapping sonification has a number of positive aspects, which Scaletti (1994) outlines in some detail. Many data dimensions can be listened to simultaneously. It is very flexible and the mappings can be easily changed, allowing different aural perspectives of the same data. In addition, acoustic production can be assigned to sophisticated tools originally developed for computer music synthesis. These are readily available and permit many quite sophisticated parameter mappings to be synthesised in real-time.

The main limitation of the technique is the lack linear independence or orthogonality[7] in the psychophysical parameter space: loudness can affect pitch perception, for example. Though conceptually simple, in practice, parameter mapping requires a working knowledge of how the parameters interact with each other perceptually. Linear changes in one domain produce non-linear auditory effects, and the range of the variation can differ considerably with different parameters and synthesis techniques. These perceptual interactions, caused by coupled perceptual parameters, can obscure data relations and confuse the listener. Flowers, an experienced multivariate data sonifier, observed that while “the claim that submitting the entire contents of ‘dense and complex’ datasets to sonification will lead to the ‘emergence’ of critical relationships continues to be made, I have yet to see it ‘work’” (Flowers 2005). However, although a truly balanced multivariate auditory display may not be possible in practice (Kramer 1994), given powerful enough tools, it may be possible to heuristically test mappings to within acceptable limits for any given application. Frysinger (2005) provide a useful overview of the history of the technique, and Flowers (2005) highlights some of its pitfalls and possible future directions.

2.3.2.2 Direct Data Audification

Direct data audification is a technique for translating data directly into sound. Kramer (1994: 186) used the unqualified *audification*, which he describes as “a direct translation of a data waveform to the audible domain for the purposes of monitoring and comprehension.” Direct data audification may be applicable as a sonification technique for datasets that have an equally–spaced metric in at least one dimension. It is most easily applied to those that exhibit oscillatory time-series characteristics, though this is not a requirement.

---

[7] A simple description of orthogonality is that of vectors that are perpendicular, such as the X and Y axes of two-dimensional geometry. The etymological origins are from the Greek ὀρθός (orthos), meaning "straight", and γωνία (gonia), meaning "angle" (OLED). The term is used here in its more general vector space definition: Two vectors x and y in an inner product space V are orthogonal if their inner product is zero. Formally, a linear transformation $T: V \rightarrow V$ is called an orthogonal linear transformation if it preserves the inner product. That is, for all pairs x and y in the inner product space V, $\langle Tx, Ty\rangle = \langle x, y\rangle$. Our concern here is to develop sonification spaces with an equally–spaced metric, that is, that preserve the psychophysical inner–product between a data vector and a linear transformation of it in that space. For a more formal definition of orthogonality, see Behnke, Bachmann, Fladt and Süss (1983: 273). There *are* subtle differences between orthogonality, dependence and correlation (Rogers, Nicewander and Toothmaker 1984) that need not particularly concern us here.

Because of the integrative capabilities of the ear, audification is useful as a technique for very large numerical datasets whose datum can be logically arranged as a time sequence of audio samples. These samples can be either stored in an audio file for delayed audition or streamed directly through the computer's audio hardware in real-time. On playback, any number of standard audio-signal processing techniques such as filtering, frequency shifting, sample interpolation, and time–and–amplitude compression can be applied, perhaps under user control, to enhance information detection. The inclusion of these techniques indicates that *direct* is used as a general descriptive, rather than taxonomic, classifier. So, while direct data audification allows for signal-processing techniques, the defining characteristic is that there are no sound-generating or acoustical models used.

Direct data audification has been shown to be effective in situations where the data is voluminous, such as that produced by monitoring physical systems such as seismology. Speeth (1961) for example, audified seismic data for the 90% successful differentiation of events caused by bomb blasts from those caused by earthquakes. Speeth's experimental results were remarkable because the task is apparently very difficult to achieve using visual plots of the data (Frysinger 2005). By time-compressing the signals to bring them into audio range, analysts could review twenty-fours hours worth of data in a few minutes. Hayward (1994), in describing the use of audification techniques on seismic data, found the technique very useful, but stressed that proper evaluation and comparisons with visual methods are needed. In summarising a body of work on earthquake data, Dombois (2002) remarks that

> eyes and ears give access to different aspects of the phenomenon of earthquakes. Free oscillations are relatively easy to recognize as the ear is all too familiar with many kinds of resonance. On the other hand synthetic seismograms, which are calculated for fitting as good as possible the curve of a measured seismogram, show low significance in the auditory display. This is less astonishing if one remembers that the power of calculation routines has been judged only by the visual correspondence of measured and synthetic seismogram.

In monitoring the operation of a complex machine, Pauletto and Hunt (2005) determined that key set of attributes (noise, repetitive elements, regular oscillations, discontinuities, and signal power) in helicopter flight data were equally discernable via an audification or a visual spectrogram. Krishnan et al. (2001) undertook a comparative study of direct data audification and other sonification techniques to represent data related to the rubbing of knee-joint surfaces and did not find that audification was the best technique for displaying the difference between normal and abnormal signals.

In summary, direct data audification with variable sample-rate playback can be useful for data 'dredging' large datasets at high speed, for bringing sub-audio information into an audible range, and for realtime monitoring by allowing buffered time-lapse playback of the most recent data. Because of the ways these types of audification appeal directly to a listener's low level pre-cognitively auditory stimulus-processing faculties, such as those described by the Gestalt psychologists and J. J. Gibson (1966, 1979, §Appendix 2), this technique is useful for monitoring global features of large time-series and in situations requiring extended passive auditing.

2.3.2.3 Homomorphic Modulation Sonification

A homomorphic mapping is one in which the changes in a dimension of the auditory space tracks changes in a variable in the dataset, with only as few mediating translations as are necessary for comprehension (Kramer 2004b: 26). This section describes a narrow interpretation of the term; here named Homomorpic Modulation Sonification. There is a subtle but important distinction between the mapping described here and the parametric mapping approach, described above, in which each datum is played as, or contributes to a separate tone with its own amplitude envelope. In the separate-tones case, the audio-amplitude profile of the resulting audible stream fluctuates from–and–to zero while with modulation, a single continuous pulsed waveform results. In the case of frequency modulation, there is the opportunity for the amplitude formant to be held relatively constant. In research reported in Chapter 5, this appears to assist the listener's auditory system to respond to it as a single modulating tone rather than a sequence of auditory objects individuated by rapid onset transients. This difference is illustrated in Figure 2.1 and may result in lower perceptual loading, especially for extended listening periods. Patterson's warning design guidelines study of navigation alerts mentioned earlier (§2.3.1.1), support this suggestion in recommending slower onsets/offsets times to avoid startling the auditor, to impress itself upon their consciousness without it dominating cognitive function.

Pre-attentive perceptual faculties are also applicable when the data is used to frequency– or amplitude–modulate a simple carrier signal. This is a relatively unexplored territory but initial results (Worrall 2004; de Campo 2007) are encouraging: in situations where audification may be an appropriate technique, but the data needs to be 'massaged' beforehand, or when there is not enough data to sustain it. For example, it is possible to apply signal processing techniques such as granulated time-stretching (vocoding), shifting the pitch while keeping time invariant, using the data to amplitude– or frequency–modulate a carrier signal. De Campo also suggests modulating frequencies of an array of sines for detection of polarity and time alignments in multiple channels of EEG data.

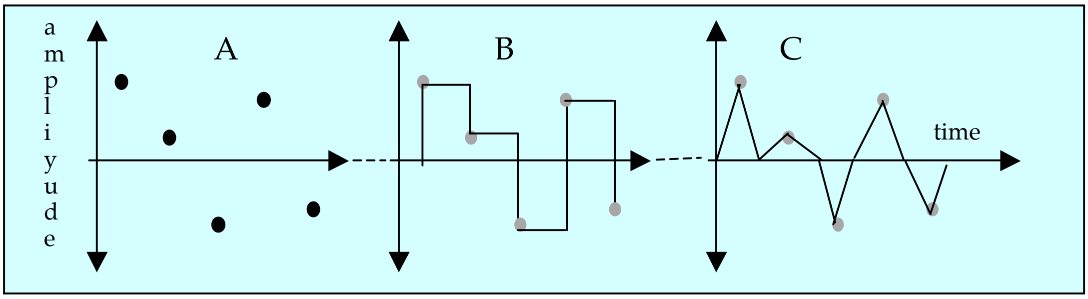


Figure 2.1.The different amplitude profiles of (A) the samples (B) being realised with amplitude modulation, and (C) individually enveloped events.

This possible difference in perceptual loading is emphasised by the noticeable difference between the resultant effect of passing parameter mapped and homomorphic modulation sonification of signals through post-synthesis signal 'enhancing' modification such as reverberation. In the case of parametric mapping, the by–products appear to enhance the original, making it more embodied, while for the homomorphic mappings, they appear as distractional, noisy artifacts, as is discussed in the homomorphic modulation sonification study reported in Chapter 5 (§5.7).

The observation that (sonic) object continuity enhances auditory attention has received research attention in the wider context of object continuity in the attentional loading of perception (Best, et al. 2008). The acoustic startle reflex is a gross version of the effect described here. This effect can be summarised as follows:

> The minimum stimulus required to elicit a response in normal rats is about 80-90 dB of 50 msec. duration, provided that the stimulus reaches full potential within about 12 msec. initial onset. In mammals, the amplitude of the startle response is

> proportional to the duration of the stimulus… The startle reflex has been observed in all mammals tested (Ladd, Plotsky and Davis 2000).

While there does not yet seem to be any empirical studies of the relative perceptual loadings of mono–modal stimuli with different spectral characteristics, there is a growing recognition of the potential for auditory displays to *inform* rather than to just alert (Vincente 2002; Watson and Sanderson 2007) and this is an important consideration for the design of interfaces to continuous sonifications.

2.3.3 Interactive data representations

2.3.3.1 Sound graphs

The term *auditory graph* is used in a variety of ways, often simply meaning the output of a multivariate data sonification. In order to provide a restricted meaning I use the term *sound graph*, to refer to a sonic representation of a visual graph (Stockman, Hind and Frauenberger 2005; Harrar and Stockman 2007). Other names by which it is known are *tone graph*, *auditory graph*, *tree-graph* and *auditory box plot*. Its function is to provide a sonified interface to a discrete dataset so that the relationships between the datapoints can be investigated interactively and asynchronously. The addition of auditory tick-marks, axes, and labels to add context is not uncommon (Stockman, Nickerson and Hind 2005).

Although the original impetus for the sound graph was to provide visually impaired people with access to line graphs and spreadsheets, it clearly has wider application, including as a design space for parameter mapping sonifications. A slightly different interpretation is provided by Vickers (2005) whose *CAITLIN*, based on design principles similar to auditory icons, is used to aurally identify and locate bugs in computer programs.

2.3.3.2 Model–based sonifications

Model–based sonification is a relatively new technique for sonification, in which parameters of a virtual physical model function in same way as the coordinates of a visual display. There are two basic kinds of models available: those based on the known physical properties of acoustic resonators and those based on the structure of the data to be sonified. Digitally modelling the resonating components of a musical instrument (called *physical modelling*) is a relatively common practice in computer music. For such dynamic models, the temporal behaviour of the components of the model are determined ahead of time by detailing their resonant properties and the way they are connected together to form a single resonator. The instrument is then excited with virtual bows, drumsticks, scrapers etc and virtual contact-microphones placed at strategic places on the 'body' of the instrument to capture its resonance. (Pearson 1996).

In *Model-Based Sonification* (Hermann and Ritter 1999; Hermann 2002), a variable of the dataset to be sonified is assigned to some structural properties of a component (elasticity, hardness etc) of the model. A user interacting with this model via 'messages'–virtual beaters, scrapers etc–causes it to resonate. The resulting sound is thus determined by the way the data integrates through the model. By virtually beating, plucking, blowing and scraping the model, the characteristics of the dataset are available to the listener in the same way that the material and structural characteristics of a physical object is available to a listener who beats, plucks, blows or scrapes it.

Model-based sound synthesis has proved effective in producing complex, 'natural' sounding acoustic events (see Pearson, op.cit. for an extensive bibliography). While the building of such models requires considerable skill, this would not be a barrier for their use in data sonification if templates were readily available. And, although they can be computationally expensive, such models tend to lend themselves to parallel computing. The application of these models to sonification is relatively new and the results of empirical testing against other sonification methods will, in time, determine the degree to which they become generally accepted techniques. Before they become more generally used, it will be necessary to make intuitive implementation tools more widely available.

One reason for exploring such techniques is to try to improve the dimensional orthogogonality of a sonification by integrating a dataset in, or through a quasi-physical medium[8]. Attempts to produce acoustically–dimensioned linear psychophysical spaces, such as with equally-distanced timbres (Barrass 1997: 88-216), so as to differentiate multiple orthogonal dimensions, on which parameter-mapping techniques rely, have not been yet

8 This matter is discussed more fully in Chapter 3.

been very successful. The representational dimensions of physical systems, on the other hand, *appear* to be more promising as they are of a higher order than acoustic parameter-maps, involve the redundant use of acoustic parameters, and are structurally closer to those in ecological use in every day listening.

2.4 The need for better tools

The *Sonification Report* considered the field of sonification research to consist of three principal components: (1) Research in perception and cognition, (2) Development of research and application tools, and (3) Design and applications using the sonification techniques. To use these components in consort requires the integration of "concepts from human perception, acoustics, design, the arts, and engineering" (Kramer et al. 1999). Hermann amplifies this by considering knowledge of data availability and meaning, as well as domain–specific dependencies, to be sufficient for simple-purpose sonifications such as for auditory alarms and the enhancement of graphic user interfaces. However in sonifications of higher-dimensional data, he considers expertise in other techniques is also necessary: statistics and data–mining, HCI, computer science, acoustics, sound engineering, physiology and neurobiology, psychology, psychoacoustics, musicology, and cognition (Hermann 2002: 24-25).

The need for better software tools, including some general proposals for adapting sound synthesis software to the needs of sonification research, has been a constant theme of auditory display researchers and was highlighted in the *Sonification Report*. Several attempts to provide generic tools are either in hibernation or extinct but there is evidence that the issues continue to be addressed (de Campo, Frauenberger and Höldrich 2004). As exploratory research continues, and software development iterates, it becomes clearer that the tools needed are not simple adaptations of existing sound synthesis software, fine though much of it is, but more robust models which can better integrate current expertise and the software expression of it from the variety of fields, that need to meet in order to advance the research into how data might be better displayed to our auditory systems. The problems raised in the *Report* prompted the work undertaken for this thesis, as first reported discussed in Worrall et al. (2007) and now more fully developed in Chapter 4.

2.5 Music and sonification: some relationships

In everyday life, sounds are produced when physical objects touch or impact upon each other. It takes no special training to be able to infer certain material qualities from the sound of impacting objects: their density, hardness and perhaps shape for example, as well as the manner of impact: collision and recoil, scraping, plucking, blowing etc. All this information is available and perceived by listeners almost instantaneously and effortlessly millions of times every day, even when they are not looking at the objects, and often even when asleep. Not having earlids, the listener's hearing constantly monitors the surrounding world, and in doing so, directs their visual and kinesthetic attention. The observation that our hearing leads our vision in making sense of the world is amply demonstrated by the importance of Foley (sound effects) in film; the imitation of the sound of a horse's hooves, over stones, through mud etc, by knocking coconut halves together, is much more convincing than an unenhanced recording of the sound of the hooves themselves. An important aspect of Foley is that its effectiveness is in remaining hidden, that is, its functions not to distinguish itself, but to enhance the realism of the audio-visual scene as a whole (Alten 2002: 364–368; Chion 1994).

2.5.1 Acoustic generation

The ability for digital computers to generate, transform and present sound to users in a timely manner is fundamental to sonification research. The vast majority of the tools and techniques used for the computer synthesis of sound have been developed by composers and engineers engaged in the task of making new music, leading John Chowning to remark, "With the use of computers and digital devices, the processes of music composition and its production have become intertwined with the scientific and technical resources of society to greater extent than ever before" (Chowning in Roads 1996: ix). Because, as he goes on to say, "a loudspeaker controlled by a computer is the most general synthesis medium in existence", these researchers were quick to begin exploring adaptations of, and eventually alternatives to, the static models of musical instrument tones as described in the literature of the time (Olson 1967: 201-241) and experience which analog synthesis had already revealed to be deficient of the lively qualities of the sounds of acoustic instruments.

The original periodic synthesis techniques include additive, subtractive and modulation models (Mathews 1967), which were augmented by dynamic (non-periodic) techniques such as microphone-sampled waveforms, stochastic function and granular synthesis-generated timbres (Xenakis 1971: 242-254; Truax 1988; Roads 2001). As computing hardware became faster and object-oriented software tools became more prevalent, software models of physical resonance systems were developed, ranging from difference equations, mass-spring, modal, non-linear excitation, waveguide, and formant (after speech) synthesis to the highly individual Karplus-Strong techniques for plucked-string and drum-like sounds. Roads provides a brief introduction to these techniques (1996: 261-315), most of which are still under active development. Their usefulness to sonification is that they provide appropriate coupling mechanisms between data structures and the synthesis models being used to express them. As Hermann (2002) concludes, apart from the often–considerable time involved in setting up such systems, their major disadvantages are the time and expertise needed to set up the models and the computing resources necessary to implement them. As experience grows and computation speeds increase, neither of these impediments is likely to prove permanent. Computing science and software engineering continues to play important roles in the development of the theoretical models and efficient algorithmic processes within which these sound synthesis techniques can be implemented.

2.5.2 Sonification research

The systematic study of using computers to *display* information to the human auditory system is a relatively young discipline. The first International Conference on Auditory Display (ICAD) was held in 1992 and Kramer's extensive introduction to the *Proceedings* (1994) provides an overview of interesting precedents, the then current state of the field as well as some possible futures. Before the computing resources with which to process large amounts of data were readily available, sonification was restricted to relatively simple tasks, such as increasing the frequency of a metal detector's audio oscillator in proportion its proximity to metals and the timely production of attention-capturing beeps and blurps. As the size and availability of datasets have grown and the complexity of communication tasks increased, so also have the expectations of sonification, requiring sonifiers to draw on skills and research knowledge from a variety of disciplines.

In such interdisciplinary enquiries, researchers from various fields come together to share and integrate their discipline-specific knowledge and in doing so reinvigorate themselves with new questions and new methods. For example, when introducing the concept of intelligent help using knowledge-based systems, Pilkington (1992: vii) notes that different disciplines often adopt different forms of evidence. While psychologists prefer experiments, computer scientists prefer programs, composers prefer sound activities, and linguists prefer rules, for example, this can lead to communication failures when they try framing a theory that will satisfy each discipline's criteria of proof. Moore (1991: 23) observes that in interdisciplinary research, the challenge is to do justice to several points of view simultaneously because, "An awareness of [a variety of] points of view is important not so much because ignorance of any one of them makes it impossible to do anything but because what may be done will eventually be limited by that lack of awareness." As experience continually reaffirms, knowledge is not value free, so when these differing points of view come from disciplines with divergent aims, a lack of such awareness may not be obvious. The opportunities to more closely examine disciplinary assumptions that collaborative work provides to researchers can lead through apparent impasses and strengthen the foundations of domain knowledge.

In the current context, it may be worthwhile to speculate on the kinds of knowledge and experience a cultural activity such as music composition can bring to the data sonification process, and inversely, of what use are the results of experimental psychology to music composition. As numerous historical examples attest–from the ancient Greeks, who believed in a direct relation between the laws of nature and the harmony of musical sounds, through the numerical symbolism of the Middle Ages[9] and beyond–musicians

---

9 One example is Dufay's 1436 motet *Nuper rosarum flores*, which is based on the proportions of the Florence Cathedral for whose consecration it was composed.

were sonifying numerical data long before computers and multivariate datasets occupied the Western mind. While their approaches were perhaps more heuristic and teleological, surviving artifacts from past eras are more likely than not to have been either workable solutions to practical problems, or sufficiently novel to have induced a new way of thinking music. However, except of the very recent past, no sound recordings are extant, and the function of the musical score as something more than an aide-mémoire is only a relatively recent one.

So, although musicologists have described many of the un-notated performance practice and 'style' characteristics of the music from contemporaneous non-musical sources, such descriptions are rarely couched in the phenomenal language of modern science. From the contemporary design perspective, the appropriation of scripted musical gestures of the historical past, together with some a-historical harmony book's 'rules of composition' which omit reference to the pitch or temperament system employed, or the timbral distinctiveness of the instruments of the period, fails to grasp the cultural imperative: music is not science, and its aims are not always for clarity of line, or purpose. The sounds of music are social sounds and as society changes so the ways its components interact are determined as much by cultural forces as by psychoacoustics (Attali 1985: 46-86).

The parametric analysis of tone according to physical characteristics (frequency, loudness, etc) and the availability of computational tools for additive synthesis from these parameters, belies the psychoacoustic evidence that, for all but the simplest sonifications, the results of design by simply assigning data dimensions to such physical parameters quickly leads to unsegregated, difficult-to-interpret clumping (Flowers 2005). Whether stream segregation is maintained during a temporal simultaneity of two or more separate spectra as components of textures or 'chords', or whether integration, occlusion or emergence results (Bregman 1994: 456-528) is dependent upon a combination of characteristics, including the system of tuning in use, the spectral density profiles (timbres) involved, the duration of the simultaneity, the previous and following directional profiles of the spectral contributors, their relative amplitudes, and so on. Furthermore, hearing is very sensitive to spectral congruence, as illustrated by the account auditing example at the beginning of this chapter. Sequences of spectral events that are highly congruent, such as occurs when using General MIDI instruments, or auditory icons for example, quickly pall on the ear. The extra cognitive load needed to work within such soundscapes is not insignificant, and possibly contributes to the annoyance commonly reported by listeners when using such sonifications for extended periods of time. There are ways out of this impasse, as illustrated by the introduction to sound synthesis techniques in common use in computer music, such as physical modelling (including voice), granulation and stochastics, as well as careful attention to second-order features such as reverberation and spatialisation; these latter being environmental ‘embodying ’ of sound sources. The importance of embodiment in information sonification, is discussed more fully in Part B of Chapter 3.

Flowers (2005) comments that most of the problems that arise in data sonification stem from lack of adequate understanding about key properties of auditory perception and attention, and from inappropriate generalisations of existing data visualisation practices. Such generalisations arise more as a result of the overwhelming dominance of vision over audition in the perceptual psychology literature than experimental evidence (Bregman 1994: 1-3). They can be overcome with more sustained applied research in auditory perception, including the rôle played by silence, and a deeper understanding of the relationship between speech and non-speech auditory processing (Slevc, Rosenberg and Patel, 2008). For sonification design to advance, *sonification designing* needs to be practiced and critiqued as designs to be interpreted, not just listened to as *musique e d'ameublement*[10]. To that end, task and data analysis of information requirements (Barrass 1997: 35-45) together with generally available datasets and interesting mapping templates in software environments that combine sonic complexity with experimentation flexibility will encourage the much-needed community-wide sharing of examples towards a catalogue of best practice.

2.5.3 Data music

The advent of the Internet and the percolation of computing devices into ordinary, everyday objects and activities, means that most of us exist in an

10 Literally “furniture music”, as so called by the composer Eric Satie.

increasingly datarised cultural environment. One characteristic of music is that it is continually being recontextualised, both technically and culturally as it is influenced by, and influences, the fluctuating *zeitgeist*. So, from a cultural, experimental composition perspective, data sonification is of significant interest.

One way of classifying data music (data sonification music) is according to where it can be placed on a perceptual continuum, with works using representational data mapping on one end and those using free data transformation on the other. Closer to the 'free data' end would be data music that use arbitrarily formatted digital documents as control data for some sound synthesis routines as arbitrarily determined by the sonifier (Whitelaw 2004). In such circumstances, it is likely that the (possibly imagined) content of the documents would play a role in culturally contextualising the resulting soundscape, in-keeping with postmodernist or other mannerist aesthetics.

At the representational end is data music that employs the techniques used in pragmatic information interpretation for decision-making, system regulation, vision substitution etc.; techniques that are derived from a developing knowledge of psychoacoustics, and the cognitive sciences more generally. For composers who are interested in engaging with listeners as interpreters rather than the struggling receivers of obscure messages, this research offers a stronger technical basis for their practice, not necessarily for overtly programmatic narratives but to better control the overall dramaturgy of a composition (Landy 2007: 36-38). For example, although the techniques for making adherent tone complexes have formed the basis of studies in orchestration for at least two centuries, in order to discover how to disassemble the ensemble, to compose coherent sounds that maintain their perceptual segmentation, what is needed is 'unorchestration' studies that include psychoacoustic techniques for producing disaggregated complex sounds in situational environments (Bregman 1994: 458) and a developing understanding of auditory cognition.

Computer-based composition tools have their origins in the desire to explore and express algorithmically-generated structures at both micro- (sound synthesis) and macro-(gestural) compositional levels. As a cultural activity, making new music is not just about exploring new ways to put new noises together but about providing others with the opportunities to engage, both actively and passively, in that exploration. Once digital musical instruments became available, MIDI was quickly adopted as a computer protocol suitable for capturing and transmitting data from human performance gesture and for coordination with other technologies, such as lighting controllers. However, not all interesting sources of data have MIDI interfaces or are related to human corporeality and proprioception. So, at a time in human history when there is an imperative for us to take responsibility for our relationships with both natural and man-made environments, the extension of sound synthesis software to aurally explore such datascapes opens the possibility of compositional access to, and thus cultural dialogue about, a broader spectrum of phenomena: interspecies, planetary and interstellar.

Notwithstanding the particular application, whether for cultural spectacles or for more pragmatic reasons, there is a general need for a new generation of software; tools that integrate flexible sound-synthesis engines with those for data acquisition, analysis and manipulation in ways which afford both experiments in cognition and lucid, interpretive *soniculations*. Such software will need to afford the exploration of the cognitive and psychological aspects of the perception of mental objects formed through the sonification of abstract multidmensional datasets that have no analogue in the material world.

The formation and support of such embodied ‘in-formation’ is a difficult task and remains a long-term goal. It will be aided by active cross-fertilisation between psychophysics, the relatively new field of cognitive science, and the older disciplines of the philosophy and psychology of perception. The next chapter and its appendices outline the epistemological and ontological dimensions of that study; the cultural ‘background radiation’ against which new, more effective, sonification software can be engineered.

THIS PAGE HAS INTENTIONALLY BEEN LEFT BLANK

Chapter 3

INFORMATION AND PERCEPTION

> Every mental phenomenon is characterized by what the Scholastics of the Middle Ages called the intentional (or mental) inexistence of an object, and what we might call, though not wholly unambiguously, reference to a content, direction toward an object (which is not to be understood here as meaning a thing), or immanent objectivity (Brentano 1874: 88).

3.1 Introduction

The task of sonification software is to provide a means by which listeners can acquire new ideas about the nature of the source of derived data. In so doing listeners can increase their knowledge of that source and thus improve the efficiency and/or accuracy of their decision-making based on that knowledge. In broad terms, sonification[1] can be considered to have four major ingredients: data, sound, listener and environment, as illustrated in Figure 3.1. In the current context, the term *listener* is used to mean a human being[2] without gross auditory system abnormalities, who, in various ways, engages in listening to sound[3] that is generated or manipulated by sonification software within an environment. *Environment* is used in a very broad sense here to mean a complex of intersecting fields that surround or encompass a listener physically, socially and psychologically.

The purpose of this chapter is to develop an understanding of what information is as a concept, as an ingredient in the acquisition of knowledge, and as something that can be communicated between its source and its (human) receiver in aural form and retained. Clearly, a complete philosophical and psychological overview is outside the scope of this thesis, however it is considered important, in the context of developing software that enables such communication, to have an understanding of some of the principles involved. Firstly, exactly what is meant by the term *information?* and secondly what are the characteristics of human sensation, perception and

---

1 *Sonification* is used synonymously with *data sonification*, as discussed in Chapter 2

2 Much of what is said could equally apply to other organisms with functioning auditory systems, or perhaps, even similarly endowed machines.

3 Here, sound is understood simply as contractions and rarefactions of air-pressure that are detected by a listener's auditory system. The ontology of sounds is addressed in Chapter 4.

cognition that affect this transmission process? By restricting the inquiry to the phenomenal aspects (what appears to us as humans), a discussion of the mechanisms of *how* it appears is largely deferred until later in the chapter.[4] By way of simple example, understanding *what* melody is and that it appears to human beings to be invariant under certain types of transformations such as pitch transposition and temporal compression, is a different enquiry to *how* it occurs, both in terms of the physical production and transmission of energy in the environment, and the physiological mechanisms by which that energy is sensed by human beings.

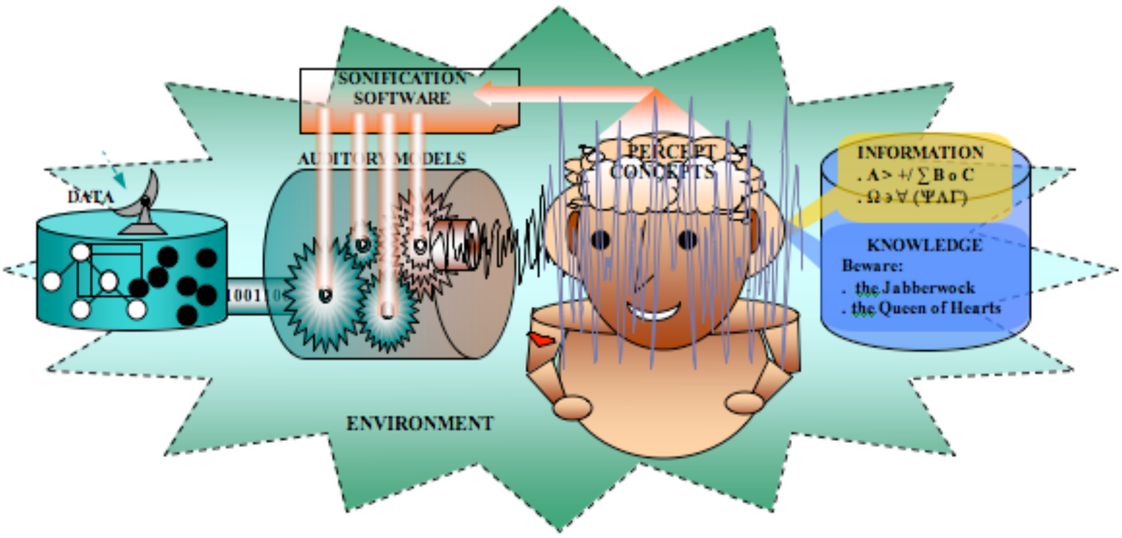


Figure 3.1. Four major ingredients of sonification: data, software, listener and environment.

The structure of this chapter is as follows. A distinction is made between information and knowledge and a contemporary understanding of information is presented. Then a quasi-historical epistemology of human perception and the types of information these epistemologies engender is followed by a discussion of the phenomenal nature of sounds and their ability to convey information of various sorts. A discussion of the means by which sounds is used to convey musical information is deferred to Chapter 4. This leads to some ideas about how sound can be more effectively used in the sonification of immanent phenomenal objects, such as those derived from multidimensional datasets, that don't have simple analogues in the material world.

4 This is not to suggest that *what* and *how* are unrelated. In fact, they exemplify two principle aspects of knowledge: objects and of relations.

3.1 Information and knowledge

There is a distinction to be made between the terms *information* and *knowledge*, despite them being frequently used in common parlance to mean the same thing. Here, knowledge is considered to be *a theoretical or practical understanding of a subject* (AOED). Most often, knowledge is gained through the integration of new information to that already retained. In broad terms, *information* can mean the content of knowledge; new information can be added to a corpus, body of knowledge[5], but not the reverse. The process of 'adding to' is termed *learning* which is considered distinct from the *structure* of the corpus, as well as the *mechanisms* of knowing it, as enunciated by researchers in the cognitive science program at Yale (Galambos, Abelson and Black 1986). These terms become somewhat problematic if they are treated as definitions rather than descriptions and their meanings have changed over time. Such changes continue to occur as our understanding of the underlying perceptual and cognitive mechanisms, together with their environmental influences, evolves. Whilst it is important to understand the different processes by which knowledge is acquired, whether from information or not, it is not central to the theme of this chapter, so a succinct overview is provided in Appendix 1.

As a process of thought, metaphysics is less directed to providing conclusions than it of exploring a mental landscape, of organizing beliefs and experiences into holistic descriptions. Of the two principal divisions of metaphysics, epistemology and ontology, the focus of this chapter is epistemological. The objective in examining the epistemology of information and perception is to gain an understanding of the nature of certain types of mental states that are induced to reside in the minds of listeners on hearing a sonification. As will be argued, because of the directed or intentional nature of perception, such information is bound between the sounds of a sonification and the intentions of listeners.

---

[5] The historical use of the expression *corpus* 'a body of knowledge' is of interest in the light of the discussion of embodiment later in the chapter.

3.2 Information: some descriptions

Considering the current scientific, economic and social importance that is given to the concept of information, it is remarkable how unclear its meaning is in some contexts. Ronald Stamper, the founder of systems analysis studies at the London School of Economics, writes about his common experience of computers being used with great technical skill but with poor results. He argues that the overwhelming reason is that there is a lot of technical skill with information technology but little knowledge about the information it carries. Whilst there is a general agreement about the nature of data, information, he remarks,

> is a vague and elusive concept. … My greatest hope is that, one day, members of the information systems profession will stop talking about information as something 'obtained by processing data to produce meaning' (Stamper 1996: 349).

3.2.1 Information as quantity of improbabilities

Information Theory is an applied mathematical theory invented by Claude Shannon to assist in finding measures of the limits to compressing data and reliably communicating it within an existing telecommunications system (Shannon and Weaver 1949). In the mathematical theory, information is understood as an objective quantifiable entity, the information content of a message being a quantitative measure of its improbability. Information Theory

> ...provides a measure for how much information is to be associated with a given state of affairs and, in turn, a measure for how much of this information is transmitted to, and thus available at, other points. The theory is purely quantitative. It deals with amounts of information - not, except indirectly and by implication, with the information that comes in those amounts (Dretske 1981 cited in Hoffmeyer and Emmeche 1991).

In such quantification, the value of information in a statement (a message) reflects the statistical structure of the statement. The question of how information was created or what should be meant by the significance of information was not addressed through the theory. In human communication, most statements are only understandable at a semantic level. Information theory requires a finite set of possibilities, or it cannot assign precise probabilities to the possibility of any particular message. Thus

Khinchin (1957: 90), more properly calls Information Theory a theory of discrete information.

Sound spectra can considered as frequency encodings of information, with white noise, which contains no redundancy, at one extreme and a sine tone, which contains maximum redundancy, at the other (Moles 1966). One of the fundamental tasks that auditory systems are known to perform well is frequency spectra differentiation (Shepard 1999), and this capability can be used in data audification "to provide a measure of how much information is to be associated with a given state of affairs" by affording comparisons between the microsound qualities of spectra-encoded data and various recognisable encodings such as white, pink, blue, and Brownian noises. Experiments 1 and 2 in Chapter 6 articulate an experimental technique for the application of sonification to aurally identify the distinction between informationless sound (a uniform distribution) and one that exhibits correlational information.

In general, the concept of information as a quantitative measure of noise or redundancy is of little use in human-to-human sensible communication, since the statistical analysis of the probabilities to be ascribed to any definite statement is not only unfeasible, but also theoretically impossible. While totally unforeseen events are demonstrably a part of life, their eventual appearance makes it impossible to ascribe distinct probabilities to any event.

3.2.2 Information: general, scientific and pragmatic

The term *information* is commonly used synonymously for an instruction, an answer, a news message or announcement and, in general, people do feel that they are often *informed* through observation and conversation. The quantitative improbability definition of Information Theory does not account for this sense of information.

In empirical science the concept *information* is as a statement (Roederer 2005). It is known as the semantic aspect of information, because it answers a pre-formulated question, such as "what is the fundamental frequency of this wave?" or it indicates the specific outcome of known alternatives, such as "Did the viola section enter one measure too early or too late?" or "Is the next card the queen of hearts?"

Only *living* systems have the capability of engaging in information-driven interactions with each other and the environment. This aspect of information is called *pragmatic* (Küppers 1990) and is often used by such systems to counterbalance the otherwise normal course of physical events. Some important characteristics of this information are

- Information is the shape or pattern of something, not the energy or forces used to give it that pattern. These forces and energy may be necessary to effect the pattern but they are separate from, and subservient to the purpose for which the information was formed.
- Information always has a *purpose*: to cause specific *change* somewhere or at sometime that would not otherwise occur, except, perhaps, by chance.
- It is what information *does* that is important, not how the form in which it resides is textured, or sounded.
- The *form* in which information resides *expresses it*, but this expression is not the information. This idea is expressed differently later on in this chapter; for example in Husserl's 'bracketing' of the phenomenal world from the 'world–as–it–is' (§3.3.3.4) and in Korzybski's "the map is not the territory"(§3.3.7). Music, for example is not a score or a CD, or the compactions and rarefactions of air pressure in a sound wave, or the neural activity of the brain. It is the intended *effect* that ultimately identifies information.
- Altering the form in which the information resides may result in a different expression or may destroy the information. For example, rearranging the notes of a melody destroys the melody and the continuity of a glissando on a violin is not preserved when expressed by an oboe.
- There is no such thing as information in isolation. Information always requires
    - An origin, source or sender where the original pattern is located or generated,
    - A recipient, where the intended change is supposed to occur,
    - A transmission from one to the other. For the purpose of a specific change to occur, a specific information–processing mechanism must exist and be activated.

- Information can be stored and, providing it has not been corrupted, retrieved, either in the form of the original pattern, or of some transformation of it.

3.3 Forms of information and perception

3.3.1 Platonic Ideals

The elusiveness of the term *information* is embedded in its historical epistemologies (Hoffmeyer and Emmeche 1991). In the Middle Ages it was used in connection with a person being informed or educated[6]–to become aware of the form (of something), give shape to, fashion (ACOD). The concept of *form* extends back through Aristotle and Plato to at least Pythagoras who is responsible for adding it as a correlative concept to the Milesian's conception of 'matter'. (Burnet 1904/1981: 35). This shift of information in the mental sphere moving to the physical sphere of substance and action, is paralleled in the shift of Form as a Platonic Ideal to something physical in which

> [f]orms were induced on, not derived from substance. They reflected human or Godly will. Thus, to bring something into form presupposed a person in whom the idea of the form must first have occurred. And this occurrence, the idea, was the root of information (Hoffmeyer and Emmeche 1991).

Burnet traces this thinking though Socrates' Doctrine of Forms, and Plato's Ideals which assert that individual objects are but imperfect examples of universals, divinely given and residing in man's soul (mind):

> There is a sharp distinction between the objects of thought and the objects of sense. Only the former can be said to be; the latter are only becoming. …We know what we mean by equal, but we have never seen equal sticks or stones. The sensible things we call equal are all 'striving' or 'tending' to be such as the equal, but they fall far short of it. Still, they are tending towards it, and that is why they are said to be becoming (Burnet 1904/1981: 126-127).

In contrast to Plato's position, which held that particular things are deduced from these *a priori* Ideals[7] by contemplation, Aristotle held that particular things contained universal essences, and such Forms (ideas) were induced

---

6 From L. *informationem* (nom. *informatio*) "outline, concept, idea," noun of action from *informare* (OED); L *informere* – in (into) + forma (form)

7 Greek ιδεα. Plato seems to use the term interchangeably with *eidos*, which serves to designate any of those primary realities that have come to be known as the Forms. The term takes on a significant philosophical meaning from his writings onwards that the term seems to develop an elaborate life of its own (Taylor 1911). See also Note 17.

from particular Category instantiations. He compares the act of perception to the pressing of a signet ring into wax:

> By a 'sense' is meant what has the power of receiving into itself the sensible forms of things without the matter. This must be conceived of as taking place in the way in which a piece of wax takes on the impress of a signet-ring without the iron or gold; we say that what produces the impression is a signet of bronze or gold, but its particular metallic constitution makes no difference: in a similar way the sense is affected by what is coloured or flavoured or sounding, but it is indifferent what in each case the substance is; what alone matters is what quality it has, i.e. in what ratio its constituents are combined (Aristotle 350BCE: BkII, Ch12).

Upon the reappearance of Aristotle's works in the west, this position was adopted by medieval Scholastics such as Thomas Aquinas, for whom the representation of an external object in the mind and the object it self was the same object in two different forms of existence. This is because, put simply, they held the same form, or *conformed*. Medievalist Henrik Lagerlund explains:

> This 'conformality' account of mental representation is for Aquinas embedded in a much larger, causal theory of the reception of these forms into the mind or intellectual soul, according to which forms are transmitted through the intervening medium between subject and object (the doctrine of the *species in medio* or 'species in the medium') and received in the external sense-organs and sense-faculty, which leads to the production of phantasms or sensible species and ultimately to the creation by the active intellect of a mental representation or intelligible species in the passive intellect (Lagerlund 2008).

3.3.2 Material Ideals

3.3.2.1 Early rational approaches to perception

In addition to the mental representations adopted from Aristotle, based on conformity and resemblance, the late Scholastics such as Ockham developed other notions based on causality and signification. Following the Scholastics, seventeenth century contemporaries Descartes, Spinoza and Leibniz continued to look for knowledge using rational thought including the continued development of logic, mathematics and Euclidian geometry.

René Descartes' starting point was his own experience (*cogito ergo sum*) and thought that the mind reached out to external objects themselves, with the sense organs and nerves serving as literal mediating links between the objects and the brain events that afforded the perceptual awareness of them. He is clear, however, that sensory awareness does not reach out to the physical things themselves; that in veridical sensation, the mind's ideas are not the immediate objects of awareness:

> …whence I should forthwith be disposed to conclude that the wax is known by the act of sight, and not by the intuition of the mind alone, were it not for the analogous instance of human beings passing on in the street below, as observed from a window. In this case I do not fail to say that I see the men themselves, just as I say that I see the wax; and yet what do I see from the window beyond hats and cloaks that might cover artificial machines, whose motions might be determined by springs? But I judge that there are human beings from these appearances, and thus I comprehend, by the faculty of judgment alone which is in the mind, what I believed I saw with my eyes (Descartes 1641: II, 13).

3.3.2.2 Material or Perceptual Idealism

At the same time and in contrast to a rational approach, try-it-and-see empirical[8] heuristics developed out of the methodologies of occidental physicians and alchemists, as exemplified by William Harvey and Francis Bacon. Their increasing success led to attempts to apply such methods to understanding how we obtain knowledge itself. The Scottish physician John Locke is credited as the founder of modern Empiricism, as he took the crucial step of denying all *a priori* knowledge (Gregory 1981: 338-339). Locke asked, if our understanding of the physical world was not descendant from Ideas but induced in the natural world, how did we acquire such understanding? He thought that a child's mind was like a blank tablet (*tabula rasa*) and all ideas came from experience, and then later, from reflection:

> Let us then suppose the mind to be, as we say, white paper, void of all characters, without any ideas: How comes it to be furnished? Whence comes it by that vast store which the busy and boundless fancy of man has painted on it with an almost endless variety? Whence has it all the *materials* of reason and knowledge? To this I answer, in one word, from *experience*. In that all our knowledge is founded; and from that it ultimately derives itself. Our observation employed either, about external sensible objects, or about the internal operations of our minds perceived and reflected on by ourselves, is that which supplies our understandings with all the *materials* of thinking. These two are the fountains of knowledge, from whence all the ideas we have, or can naturally have, do spring. (Locke 1690: Book II, Ch.1)

Locke's concept of perception was simple: objects have characteristics, and perception reflects the world much as a mirror reflects objects. The Irish bishop George Berkeley went on to argue that if an empirical analysis of human knowledge was followed rigorously, it had to be admitted that all

---

[8] Empirical. First use recorded in 1569, from L. *empiricus*, from Gk. *empeirikos* "experienced," from *empeiria* 'experience', from *empeiros* ''skilled', from *en-* 'in' + *peira* 'trial, experiment'. Originally the name of a school of ancient physicians who based their practice on experience rather than theory [OLED].

experience is nothing more than experience, that the qualities that the human mind registers are ultimately experienced in the mind itself, and there can be no conclusive inference from this as to whether or not some of those experiences represent or resemble anything outside it.

> There was an odour, that is, it was smelt; there was a sound, that is, it was heard; a colour or figure, and it was perceived by sight or touch. This is all that I can understand by these and the like expressions. For as to what is said of the absolute existence of unthinking things without any relation to their being perceived, that seems perfectly unintelligible. Their *esse* is *percipi*, nor is it possible they should have any existence out of the minds or thinking things which perceive them (Berkeley 1710: Book II, Ch.3).

However, the similarity of individual's reported experiences of the world convinced Berkeley that these experiences indicated the presence of order that was not just subjectively determined by whimsical fantasy. He postulated that the order that inheres in the world depends on God. (Tarnas 1991: 335-336).

Faced with things considered unknowable, both Locke and Berkeley relied on the same resolution as the majority of their predecessors–to a universal mind that transcends individual minds, which, as Gregory (1981: 346) puts it, *takes away the interest of this account*. David Hume agreed with Berkeley's empirical arguments but saw no reason to accept the resolution of a universal mind. He reasoned that, because the intellect cannot deduce the veridicality of received sensations or the connection between these sensations and the Truths in the intellect, the only truths of which the intellect is capable are tautological, that is are not capable of producing new information. He goes on to argue that, not only can one not logically *deduce* the nature of things using reason alone, neither can one *infer* them from experience. The only thing that connects sequences of sensations together is the principle of causality, which relies on the experience of individual concrete events in temporal succession.

> When we look about us towards external objects, and consider the operation of causes, we are never able, in a single instance, to discover any power or necessary connexion; any quality, which binds the effect to the cause, and renders the one an infallible consequence of the other. We only find, that the one does actually, in fact, follow the other (Hume 1777: Book VII, Ch. I, §50).

Hume maintained that whilst one might recognize the regularity of perceived events, the acceptance of the necessity for events to follow each other is not based on logical certainty, but on habit.

> When we say, therefore, that one object is connected with another, we mean only that they have acquired a connexion in our thought, and give rise to this inference, by which they become proofs of each other's existence: A conclusion which is somewhat extraordinary, but which seems founded on sufficient evidence (Hume 1777: Book VII, Ch. II, §59).

The mind draws an explanation of experience from itself and it cannot know what *causes* a sensation because it doesn't experience 'cause' as a sensation. Further, he goes on to argue, the concreteness of these individual events cannot be logically asserted, so causality is made meaningless. If all human knowledge is based on empiricism, and induction cannot be logically verified, there can be no certain knowledge. So, Hume had begun by trying to eliminate the necessity for metaphysics in deductive rationalism to investigate man's reasoning by using the empirical experimental techniques employed so successfully by Galileo and Newton, and ended up like Berkeley, skeptically questioning whether objective certainty could ever be empirically achieved because inductive inference couldn't be logically justified. (Tarnus 1991: 340).

3.3.3 Intention: Transcendental ideals and phenomena

3.3.3.1 Kant's Transcendental Idealism[9]

Although Immanuel Kant admired Hume's reasoning that, in his own terminology, causal judgments are synthetic, involving an act of the mind that connects the cause and the effect, he was also convinced that the (empirical) experimental methods of Newton had really revealed generalised knowledge, so he set about the task of trying to unify the rational and empirical methods. In other words, to show how (scientific) knowledge is possible, how both reason and experience contribute to that knowledge, and in doing so, refute the skepticism of Hume's claim that experience and reason are extremely limited in the kinds of knowledge they can provide.

In *Critique of pure reason* (hereafter CPR), Kant proposed that the 'world' that science explained was a world already ordered by the mind's own cognitive apparatus. He begins by agreeing with Locke and Hume that there are two kinds of mental propositions: those in the intellect based on sensations and those in the intellect alone, based on relations between

---

9 Transcendental Idealism is Kant's term for his doctrine that knowledge is a synthetic, relational product of the logical self (Runes 1942). Sometimes called Critical Idealism.

concepts. For him, it was the (internal, conceptual) relations between intuited sensations that constitute forms of appearance:

> The capacity (receptivity) for receiving representations through the mode in which we are affected by objects, is entitled *sensibility*. Objects are *given* to us by means of sensibility, and it alone yields us *intuitions*; they are *thought* through the understanding, and from the understanding arise *concepts*. …
>
> The effect of an object upon the faculty of representation, so far as we are affected by it, is *sensation*. That intuition which is in relation to the object through sensation, is entitled *empirical*. The undetermined object of an empirical intuition is entitled *appearance*.
>
> That in the appearance which corresponds to sensation I term its *matter*; but that which so determines the manifold of appearance that it allows of being ordered in certain relations, I term the *form* of appearance. (Kant 1787/1929: 65).

Kant suggests that there are two 'pure' forms of sensible intuition that serve as principle of *a priori* knowledge: space and time:

> That in which alone the sensations can be posited and ordered in a certain form, cannot itself be sensation; and therefore, while the matter of all appearance is given to us *a posteriori* only, its form must lie ready for the sensations *a priori* in the mind, and so must allow of being considered apart from all sensation.
>
> I term all representations *pure* (in the transcendental sense) in which there is nothing that belongs to sensation. The pure form of sensible intuitions in general, in which all the manifold of intuition is intuited in certain relations, must be found in the mind *a priori*. This pure form of sensibility may also itself be called *pure intuition*. [For example,] if I take away from the representation of a body that which the understanding thinks in regard to it, substance, force, divisibility, etc., and likewise what belongs to sensation, impenetrability, hardness, colour, etc., something still remains over from this empirical intuition, namely, extension and figure. These belong to pure intuition, which, even without any actual object of the senses or of sensation, exists in the mind *a priori* as a mere form of sensibility. (Kant 1787/1929: 66).

Kant considers space and time to be the two forms in which sensibility occur because they are obtained not from sense data but by the actions of the mind in coordinating the sensations it receives. In other words, we intuit the (real-world) objects of our senses by the detection of the manifold presentation of sensations (from hearing, touch etc) of those objects by the two aspects of our minds that are not (empirical) sensations, namely space and time. Whatever is perceived is perceived as having spatial relations and/or temporal relations.

In CPR, Kant sought to determine, despite Hume's conclusion otherwise, if it was possible to expand knowledge by proving the validity of propositions that are true without reference to experience. In the process, he accomplished what he called a 'Copernican revolution' in philosophy, now known as 'transcendental idealism': whereas philosophers had previously considered the mind to be a passive agent of knowledge of an objective world, he showed that the actual world cannot be known, but that the human understanding of reality is shaped by both our means of perceiving it (sensible intuitions) and pure (transcendental[10]) concepts which are available to us through our awareness of space and time. These transcendental concepts are derived from the translation of sensible Judgments of Quality, Quantity, Relation and Modality into transcendental Categories[11] (Kant 1787/1929: 104-119).

Having described the process of form perception, Kant asserted that all necessity is grounded in a transcendental condition and therefore, so must consciousness, which he calls transcendental apperception[12], *the synthesis of the manifold of all our intuitions* - the process whereby perceived qualities of an object are related to past experience, the inner self. Without transcendental apperception, the consciousness that is the foundation of the synthetic unity of experience, it would be impossible to think. Kant made a distinction between this transcendental apperception and empirical apperception which he described as the consciousness of the changing states of the inner self, or *[c]onsciousness of self according to the determinations of our state in inner perception is merely empirical, and always changing* and it is as a result of this mental plasticity, this openness of the inner self to change, that knowledge, *the final goal of the understanding in combining intuitions and concepts*, is possible (Kant 1787/1929: 136).

Kant answers the question of how synthetic *a priori* propositions of mathematics are possible, with his Transcendental Aesthetic and the doctrine

---

[10] In Kant's philosophy, the adjective *transcendental* is applied to the condition of experience or anything related to it. Transcendental knowledge is possible, though transcendent knowledge is not (Runes 1942). Transcendental cognition is thus *a priori* knowledge. Kant description of the expression 'transcendental knowledge' as *all knowledge which is occupied not so much with objects as with the mode of our knowledge of objects in so far as this mode of knowledge is to be possible "a priori"* (Kant 1787/1929: 59), is the meaning used throughout the current work. Not to be confused with the use of the term by the New Englanders Emerson, Thoreau, Fuller and Ives etc.

[11] These categories are derived from Aristotle's Categories: Substance, Quantity, Quality, Relation, Place, Time, Position, State, Action, and Attention. *[O]ur primary purpose being originally identical with his, notwithstanding the great difference in the execution.* (Kant 1787/1929 113).

[12] The term *apperception* was introduced by Leibniz to denote the introspective or reflective apprehension by the mind of its own inner states, in contrast to *perception* as the inner state of representation of outer things. In psychology, the term came to be used for the process by which an individual's new experience is assimilated into and transformed by the residuum of their past experiences to form a new whole (Runes 1942).

of the transcendental ideality of space and time. He answers the question of how synthetic *a priori* propositions of natural (empirical) science are possible in his Transcendental Analytic, where he demonstrates the essential role the categories play in establishing the possibility of knowledge from experience. In attempting to refute Hume's idealism, Kant concluded that we cannot be certain of the knowledge of the objects of our sense experiences 'as they really are' (*Ding an sich, noumena*[13]); that *a priori* knowledge of them can only be transcendental.

3.3.3.2 Kant's refutation of material idealism

Kant identifies two type of material idealism: Dogmatic and Problematic. The Dogmatic idealism of Berkeley asserts that things in space are merely imaginary entities. Whilst Kant acknowledged that this position is unavoidable if space is interpreted as a property of things in themselves, he showed through his Transcendental Aesthetic, that space is an *a priori* pure (non-empirical) intuition in us prior to any perception of objects and thus does not represent a property of objects. (Kant 1787/1929: 70-71).

In holding the empirical assertion that *I am*, the Problematic idealism of Descartes is pleading our inability to prove, through immediate experience, any experience other than our own.

> The required proof must, therefore, show that we have *experience,* and not merely imagination of outer things; and this it would seem cannot be achieved save by proof that even our inner experience … is possible only on the assumption of outer experience (Kant 1787/1929: 244).

Idealism assumes that the only immediate experience is inner experience from which we can only *infer* outer things, and then only uncertainly. Turning the idealists assertion on its head, Kant argues that, given that I am conscious of my own existence in time, and an awareness of time requires something permanent in my perception against which I can observe the progress of time, this permanence cannot be *in* me, that is, cannot be a property of me, since it is only through it that I can be conscious of my existence in time. So, it follows that

> perception of this permanent is possible only through a *thing* outside of me and not through the mere *representation* of a thing outside of me; and consequently

[13] Many, most famously Schopenhauer, in his *Criticism of the Kantian Philosophy* (1844), consider Kant to have inappropriately appropriated the Greek word *noumena*, 'that which is thought', and used it to mean 'things-in-themselves'.

> the determination of my existence in time is possible only through the existence of actual things which I perceive outside of me. … [and *ipso facto*] the consciousness of my existence is at the same time an immediate consciousness of the existence of other things outside me (Kant 1787/1929: 245).

However, although *I am* immediately includes the existence of outer things, it does not follow that every intuition of outer things implies their existence, because a mental representation (thought) of them might have, for example, been imagined, dreamt or hallucinated. Kant thinks such (imagined etc) representations are the reproduction of previous outer perceptions (Kant 1787/1929: 247).

In showing the way reason and empirical enquiry fitted together, Kant demonstrated the role of the human mind in *constructing* reality and knowledge and his insight affected all subsequent epistemological enquiry. In the course of addressing Hume's skepticism, that the inherent nature of things (the so-called *thing–in–itself*) is always beyond the reach of our empirical capacity to know, Kant had divided the world between the *phenomenal* world of empirically unknowable things (the world measured by Newtonian physics, for example) and the ever-unreachable *noumenal* world of *things-in-themselves* which we know exists if we exist. A phenomenon, then, is an object of empirical knowledge that is conditioned by space, time and the categories.

3.3.3.3 Brentano's mental phenomena and intentional inexistence

The general view of post-deists was that consciousness was composed of simple or basic elements–experiential events–that combine to form complex experiences. Both Locke and Hume considered that such simple events were parts of experiences and must, given the seeming unrestricted possibilities for their recombination, have a certain independence vis-à-vis other parts with which they might be realized.

> When the Understanding is…stored with…simple Ideas, it has the Power to repeat, compare, and unite them even to an almost infinite Variety, and so can make at pleasure new complex Ideas (Locke 1690/1975: Bk.II, Ch.2, §2).

> [A]ll simple ideas may be separated by the imagination, and may be united again in what form it pleases (Hume 1739/1978: Bk.I, Part I, §IV).

The neo-scholastic philosopher and psychologist Franz Brentano (1838-1917) was widely influential as a teacher and mentor and his work is often considered precursory to Phenomenology[14], a philosophy principally developed from his descriptive psychology by his student Edmund Husserl who used the term to mean

> the reflective study of the essence of consciousness as experienced from the first-person point of view (Smith 2007, cited in Wikipedia).

It became clear, largely through the work of Brentano and his students (principally Stumpf and Husserl), that such a strong form of independence among experiential parts is unnecessary; in fact, various kinds of dependency relations are possible. Brentano published the first volume of his large-scale work on the foundations of psychology, *Psychology from an Empirical Standpoint,* in 1874. He emphasized that all our scientific knowledge should be based on direct experience. However, for him, doing empirical psychology meant describing what one directly experiences in inner perception from a first-person point of view, in contradistinction to contemporary empirical science, which attempts to take a third-person approach. Brentano's aim was to provide a scientific basis for psychology which, in his *Psychology*, he defined as *the science of mental phenomena* (1874: 18), and which he distinguishes from physical phenomena because they are the exclusive objects of inner perception, they always appear as unities, and they are always intentionally directed towards objects (Huemer 2008). He later renamed his approach as *descriptive,* to distinguish this first-person analyses of subjective processes from the third-person *genetic* approach in which causal or genetic laws are developed to explain what phenomenology merely describes.

Brentano was essentially an idealist, in that he maintained that external, sensory perception could not tell us anything about the *de facto* existence of the perceived world, though we can be absolutely sure of our internal perception. When I hear a police siren, I cannot be completely sure that there is a police siren in the real world, but I am absolutely certain that I do have an internal perception, that I hear. External sensory perception can

[14] Not to be conflated with Phenomenalism, the doctrine that all human knowledge is confined to the appearances presented to the senses. See §3.3.4.4. Husserl's pupil Martin Heidegger believed that Husserl's approach overlooked basic structural features of both the subject and object of experience. He developed a more ontological approach to phenomenology that influenced the development of existentialism. Hegel also used the term to describe a dialectical phenomenology that begins with an exploration of that which presents itself to us in conscious experience (phenomena) as a means to finally grasp the ontological and metaphysical Spirit that is behind them. As already indicted, the ontological implications are beyond the scope of the current discussion and have been all but suppressed.

only yield *hypotheses* about the perceived world, neither the truthfulness of such perception nor the existence of the apparent external origin of such perception.

In considering the separability of experiential parts of consciousness, Brentano distinguished between *merely distinctional* and *actually separable* parts, the latter of which are either *one-sided* or *mutually* (two sided) separable. (Brentano 1891/1995: 15). An example of one-sided separable parts is in evidence in the way one can hear (notice) a sound without listening to the sound itself, but can't listen to a sound without noticing it. Multi-sensorial experiences have frequently mutually–separable parts. For example, the visual and aural experiential parts of a violin performance can be experienced separately, as can the aroma and sight of a banana.

Tracing the idea back through the Scholastics, Descartes and the Greeks, Brentano introduced the notion of the directed intention of mental objects into contemporary philosophy:

> Every mental phenomenon is characterized by what the Scholastics of the Middle Ages called the intentional (or mental) inexistence of an object, and what we might call, though not wholly unambiguously, reference to a content, direction toward an object (which is not to be understood here as meaning a thing), or immanent objectivity (Brentano 1874: 88).

Rather than developing a full and systematic description of intentionality, Brentano's goal was to outline the criteria for distinguishing mental and physical phenomena and he used the terms *mental or intentional inexistence* to refer to what today is know as a characteristic of consciousness: the mind's capacity to *refe*r or be *directed* to objects that existing solely in the mind. Although Brentano's formulation of intentional objects does not address their ontological status, this was addressed, at least to some extent, by his students, principally Alexius Meinong and Edmund Husserl. Meinong's concern was with the intentional *relation* between the mental act and an object. He maintained that such a relation existed even when the object external to the mental act towards which it is directed doesn't exist, such as Pinocchio, Orpheus, Unicorns and the Fountain of Youth. Earlier, Hume had considered the concept of non-existent objects contradictory, Kant and Frege considered it logically ill-formed and later, Russell adopts the idea (Reicher 2006).

These examples seem more like Platonic Ideals than non-existent objects. In data sonification, a dataset can be considered as a particular ('accidental') object, so it seems reasonable to assume that an exploration of the rôle of non-existent objects in data sonification is only of relevance if can be reasonably concluded that sounds, or the other objects of data sonification software, cannot be ontologically accounted for in other ways.

According to Brentano's theory, mental acts cannot have duration. This brings up the question of how we can perceive temporally extended objects like melodies. Brentano accounted for these cases by arguing that an object towards which we are directed does not immediately vanish from consciousness once the mental act is over. It rather remains present in altered form, modified from *present* to *past*. Every mental phenomenon triggers an 'original association' or *proteraesthesis*, as he calls it later, a kind of memory which is not a full-fledged act of remembering, but rather a part of the act that keeps lively what was experienced a moment ago. When I listen to a melody, for example, I first hear the first tone. In the next moment I hear the second tone, but am still directed towards the first one, which is modified as past, though. Then I hear the third tone, now the second tone is modified as past, the first is pushed back even further into the past. In this way Brentano can explain how we can perceive temporally extended objects and events.

3.3.3.4 Husserl's transcendental phenomenology

The notion of intentionality also played a central role also in Husserlian phenomenology. Applying his method of the phenomenological reduction, however, Husserl addresses the problem of directedness by introducing the notion of *noema*, which plays a role similar to Frege's notion of *sense*.

Edmund Husserl (1859-1938) was the first to apply the term *Phänomenologie* (phenomenology) to a whole philosophy and his usage of the term has largely determined the sense of it in the twentieth century[15]. As the term underwent development since his early use of it, the present comments are derived from Husserl's own mature introduction (1927)[16] in which he outlines the two principal uses of the term: Firstly, in application to "a new kind of descriptive method which made a breakthrough in philosophy at the turn of the twentieth century, and secondly, a new psychological discipline parallel to the first in method and content: the a priori pure or

15 In contradistinction to Hegel's 1807 use of the word in his *Phänomenologie des Geistes* (Phenomenology of the spirit), in which is expressed a radically different concept.

16 A summary of the historical development of Husserl's connotation of the term is available in Runes (1942).

"phenomenological" psychology, which raises the reformational claim to being the basic methodological foundation on which alone a scientifically rigorous empirical psychology can be established."

Husserl's aim was that the empirical approach, which he calls *an objective science of nature* and which, significantly, he considered as a branch or anthropology or zoology in that it was decidedly *egoical* (anthropocentric) and physicalist, would lead to a better understanding of philosophical phenomenology, which he calls *pure psychology*, after Brentano's *descriptive psychology*. While the term has fallen into disuse in favour plain of *philosophy*, it does exemplify the importance that early scholars involved in the burgeoning science of psychology placed on dynamic interplay between empirical investigation and its philosophical underpinnings (Husserl 1927/1971: §5). It is the contention here, as exemplified by the very existence of this chapter in the current work, that this relationship remains important, especially when creating, as we do with data sonification software, phenomena (objects, events) that need not conform to any resonating material Kantian thing–as–it–is.

Building on Brentano's outline, the notion of intentionality played a central role in Husserl's development of phenomenology. He understood intentionality as a fundamental attribute of subjective processes, maintaining that phenomenology must describe these processes, not only with respect to their immanent cognitive components, but also with respect to their intended (external) objects.

> In unreflective holding of some object or other in consciousness, we are turned or directed to-wards it: our *intentio* goes out towards it. The phenomenological reversal of our gaze shows that this *being directed* [*Gerichtet-sein*] is really an immanent essential feature of the respective experiences involved; they are "intentional" experiences (Husserl 1927/1971: §2).

Husserl describes how "a multiple and yet synthetically unified intentionality" arises from the phenomenological reflection on the synthesis of appearances of an object as orientation to in changes. In considering his example of a die, he mentions orientations of left-and-right, near-and-far, front-and-back. These orientations form an intentional structure comprised of the perceptions of the *actually-seen*, *undetermined* (such as the back side) and *unseeable* which has to be inferred because they are obscured. This 'directness of appearance' formed the basis of Gestalt psychology, yet Husserl was thinking of more than the direct appearances:

> The intentional structure of any process of perception has its fixed essential type [*seine feste Wesenstypik*], which must necessarily be realized in all its extraordinary complexity just in order for a physical body simply to be perceived as such. If this same thing is intuited in other modes-for example, in the modes of recollection, fantasy or pictorial representation- to some extent the whole intentional content of the perception comes back, but all aspects peculiarly transformed to correspond to that mode (Husserl (op cit.).

He was also not restricting himself to visual phenomena:

> This applies similarly for every other category of psychic process … these constitute themselves, with fixed essential forms corresponding to each process, in a flowing intentionality (op cit.).

but was aiming to

> investigate systematically the elementary intentionalities, and from out of these [unfold] the typical forms of intentional processes, their possible variants, their syntheses to new forms, their structural composition, and from this advance towards a descriptive knowledge of the totality of mental process, towards a comprehensive type of a life of the psyche (op cit.).

It follows, then, that a phenomenal sound object is a *multiple and yet synthetically unified intentionality* whose appearance is revealed through time, and consists of *actually-being-heard*, *undertermined* (previously heard) or *unhearable*–a missing fundamental, for example.

Husserl's phenomenology was a starting point and major influence on Pierre Schaeffer's treatise on musical objects (1966) as objects in their "flux of modes of appearing and the manner of their 'synthesis'". Husserl's concern was to provide a theory of the purely phenomenal, psychical, multiple 'appearances' of objects; each appearance being a unit or component of meaning accruing to the phenomenal object as each of these appearances occurs. In order to do this he applied an attitude of 'self-restraint' (*epoche*) in which he 'bracketed-off' consciousness from the 'world-as-it-is' in order to develop what he later called an *eidetic* science of essential forms[17] that is, one that involves no assertion of actual material existence; pure, in the sense that

[17] Husserl calls these forms *Eide*, singular *edios*. Greek origin: "By *eidos* I mean the essence of each thing and its primary substance" (Aristotle, *Metaphysics*. The verb is *eido* 'to see' appears in the Latin verb *video*. The term is related to Sanskrit's 'veda', a cognitive activity like 'knowing', and the Old English *wit*, 'to know'.

pure logic is pure[18]. In constructing such phenomenological psychology it could be

> exclusively directed toward the invariant essential forms. For instance, the phenomenology of perception of bodies will not be (simply) a report on the factually occurring perceptions or those to be expected; rather it will be the presentation of invariant structural systems without which perception of a body and a synthetically concordant multiplicity of perceptions of one and the same body as such would be unthinkable. (op cit. §4).

While Kant's proof of the existence of things outside of him even though their nature was uncertain (§3.3.3.2); of the necessity for 'things as they are' as a chink in the Cartesian mind/body separation, had an impact on Husserl, his phenomenally pure psychology is still firmly bound to the mind. In his four criteria for the phenomenally pure psychology, there is no recognition of the body at all:

1. The description of the peculiarities universally belonging to the essence of intentional mental process, which includes the most general law of synthesis: every connection of consciousness with consciousness gives rise to a consciousness.
2. The exploration of single forms of intentional mental process which in essential necessity generally must or can present themselves in the mind; in unity with this, also the exploration of the syntheses they are members of for a typology of their essences: both those that are discrete and those continuous with others, both the finitely closed and those continuing into open infinity.
3. The showing and eidetic description [Wesensdeskription] of the total structure [Gesamtgestalt] of mental life as such; in other words, a description of the essential character [Wesensart] of a universal "stream of consciousness."
4. The term "I" designates a new direction for investigation (still in abstraction from the social sense of this word) in reference to the essence-forms of "habituality"; in other words, the "I" as subject of lasting beliefs or thought-tendencies-- "persuasions" --(convictions about being, value-convictions, volitional decisions, and so on), as the personal subject of habits, of trained knowing, of certain character qualities (Husserl 1960).

To some, he struggled, despite his protestations to the contrary, to separate his *eide* ('essential forms')[17] from Platonic Ideals. (Smith 2007, Martin 2007)

3.3.3.5 Gestalt psychology

Gestalt psychology arose from experimental findings of perceptual *gestalts*, just mentioned above, principally in the empirical experimentation laboratories established by another of Brentano's pupils, Carl Stumpf (1838-1946). This occurred at a time when what we now think of as experimental psychology was separating itself from philosophy into its own discipline. Max Wertheimer, Kurt Koffka, and Wolfgang Kohler were instrumental in the founding of Gestalt psychology (Hergenhahn, 1992). They considered

---

[18] Husserl described such eidetic phenomenon as *noetic*, that is, *noemata* (singular *noema*) conceived in the stream of consciousness (*noesis*) entirely by reason (*nous*).

perceptual gestalts as phenomena arising naturally and directly from the physical nature of sensation.

Although initially Gestalt theory began as a theory of perception, Wertheimer clearly recognized the fundamental epistemological issue, and Kohler and Koffka presented Gestalt theory explicitly on epistemological dualist grounds. Curiously, as Harvard cognitive-neuroscientist Steven Lehar (2000) indicates, the Gestaltists did not reference Kant as the originator of the idea of gestalts, possibly because of their confusion over his Idealist position. Lehar also suggests that one of the most controversial and pivotal aspects of Gestalt theory is Wertheimer's principle of isomorphism, which was elaborated by Kohler in 1924 as the hypothesis that every perceptual experience is "not only blindly coupled to its corresponding physiological processes, but is akin to it in essential structural properties" (Kohler, quoted in Lehar 2000) which, he suggests "is a direct consequence of the indirect realist foundations of Gestalt theory, whereby phenomenal experience is a direct manifestation of neurophysiological processes in our physical brain, and therefore it cannot help but be similar in structure, since they are identical in ontology."

Like Husserl before him, Ernst Mach (1838–1916)[19] was interested in the phenomenon of melody. Mach understood sensations to be not simply raw experiences but the interaction of experience with a pre-formed cognitive structure. For instance, when we hear a known melody, we recognize it no matter what its transposition or even mode. It can be hummed, whistled, or plucked from a harp. Furthermore, even if one or more notes are incorrect, we still recognize it. In *Analysis of Sensations* (1886) Mach asks, "What constitutes a melody?" It seems empirically incorrect, he argued, to say, as we must in recognition of the above, that a melody exists in our ability to recognize it and not in sounds; having been formed as an idealisation by experience of one or more examples of it. This idealization captures not the actual sounds, but the relationships of the sounds to one another. For Mach, this process is at the basis of all perception. Experience requires an a *priori*, but that *a priori* is itself formed by experience.

[19] Mach lends his name to the speed of sound and was considered by Einstein to be his forerunner on the theory of relativity (Einstein 1954: 26).

John Dewey[20] also thought that immediately felt or sensed experiences constitute by far the larger part of total human experience, but which are on a different level from the knowledge experience. Sensory qualities, he contended, “are not objects of cognition in themselves, … [but] acquire cognitive function when they are employed in specific situations as signs of something beyond themselves“ (1938: 147). The French phenomenological philosopher Maurice Merleau-Ponty (1908–1961)[21] described the structure of perception as sensation and conscious awareness, between which phenomenology uncovered a ‘figure-ground’ invariant. As he put it: “To be conscious=to have a figure on a ground–one cannot go back any further” (1964: 191).

In the second quartile of the nineteenth century, the mathematical notion of the group was being discussed, though its full importance as an organising and clarifying principle was yet to be realised. According to the neo-Kantian philosopher Ernst Cassirer, the first attempt to apply speculations concerning the *concept of group* to psychological problems of perception was made by the physician and physicist Hermann von Helmholtz in 1868 (Cassirer 1944). Amongst his many enterprises, Helmholtz was interested in tracing Kant's philosophical theories in fields such as physiology and the newly emerging empirical psychology. He recognised that at the root of perception is the concept of the *constancy* of perceptual objects under changing sensory conditions; of relations that remain unchanged or invariant under transformation. He endorsed Kant's thesis of space as a "transcendental form of intuition" but for him this was the beginning of, not a solution to, the problem of finding the most general form of invariance, that is, in which systems of points in a multi-dimensional manifold may be displaced relative to one-another without changing their forms.

Henri Poincaré recognised the concept of the group as a fundamental *a priori* one derived from an intuition that is imposed on us "not as a form of our sensibility, but as a form of our understanding" (Poincaré 1913: 79), and as

---

20 Humanist philosopher, educator and founder, with Peirce and the psychologist William James, of the Pragmatism school of philosophy.

21 A phenomenological philosopher strongly influenced by Husserl and Heidegger and closely associated with Sartre and de Beauvoir. He was the only major phenomenologist of the first half of the Twentieth Century to engage extensively with the sciences, and because of it, his writings have become influential with the recent project of naturalizing phenomenology as discussed in §3.3.6.

such precedes and underlies all experience. This leads him to conclude that the axioms of geometry and empirical statements derived from observation and measurement cannot be compared because there is an irreducible difference between them; they belong to entirely different objects. It is a characteristic of perception that it can never abandon the here–and–now (*hic et nunc*), since its task is to apprehend it as precisely and completely as possible. Helmholtz is similarly clear that laws cannot be responsible for the causes of natural phenomena; that explanations are the urge of our intellect (our intent, or what he calls "judgment") to bring our perceptions under its control. Further, that though the perceptual world does possess a structure, objective stimuli are not simply copied in perception, but are transformed in a certain direction.

Cassier understood, following Hering's inquiries into the sense of light, that perceptual content is characterised by the reduction of dissimilarity in objective stimulus rather than construction of similarity. Thus perception integrates the impressions of stimulii rather than being bound to their flux. His question (1994: 12) is whether it is a mere accident that the concepts of invariance and transformation belonging to group theory, recognised as a being fundamental to mathematics, appears in the exposition of psychological facts, even if the connection from such extrapolation is a "mediate" one. He concludes that relationship between the formation of invariants in perception and in geometry is an analogous one; that the differences may be characterized by an expression which Plato used to define the opposition of perception to thought in which all perception is confined to the "more or less"; that the realisation of perceptual constancy is never ideally complete, but always remains within certain limits and beyond these limits there is no further “transformation."

3.3.4 The immediate perception of objects through sensation

Most of the epistemological discussions that followed the idealists accepted a subjectivist or internalist explanation of perceptual experience; namely, that whilst the veridicity of perceived objects cannot be ascertained on the basis of sense data alone, the experience of *internal* representations of objects was such as to justify inferring the existence of the corresponding *external* objects. What follows provides and overview of the principal non-naïve[22], non-sceptical[23] explanations for how immediate perception of physical objects and the physical world generally can be warranted on the basis of sensory experience: *perceptual subjectivism*, consisting of *phenomenalism*, *representionalism* (otherwise known as *indirect realism*) and *direct realism*, and less subjective *external* or *process reliabilism. Perceptual subjectivism* has a longer history and is widely accepted as the most justifiable (BonJour 2007), while *direct realism* grew out of Gestalt psychology, and the development of *reliabilism* began in the twentieth century and continues today.

Historically, a distinction has been made between sensations and perceptions: sensations are basic experiences elicited by simple stimuli, while perceptions are experiences elicited by complex, often meaningful stimuli. Being more complex, perceptions are often considered to be the result of the integration of simpler sensations. They may also involve other processes such as memory, thus ensuring the possibility of them being influenced by a knowledge of past experiences, and independent of the methods by which that knowledge is acquired, as discussed in Appendix 1.

The relationship between personal knowledge and perception has been extensively researched. For example, a faint tone is easier to hear if a listener knows what pitch to expect (Green 1961 cited in Sekuler and Blake 1985: 423) and comprehension is enhanced for native speakers of a language when they understand that someone is speaking with a strong foreign accent. Other examples include the increasing ease, on perseverance, in reading someone's scrawly handwriting and the ease, having acquired the skill, of reading music or riding a bicycle. On the other hand, existing knowledge can impair or mislead accurate perception through mistaken or inferential expectations and ambiguities.

The term *immediate perception* is use to denote the content of those propositions that arise in the mind directly from external sensation, as distinct

22 The naïve position, know as *naïve realism* is that physical objects are directly experienced; rejected by a large proportion of the philosophers as discussed in the previous section.

23 The *skeptical* position is that sensible experience provides no evidence of external substances. Arising in the fourteenth century, it was used by those for whom the only certitudes are those of immediate experience and those of principles known *ex terminis* (by definition) together with conclusions immediately dependent on them. Most sceptics usually accepted a degree of probabilism, namely, that probability is the only guide to belief and action. Despite this lack of direct influence, the sceptical arguments of fourteenth century thinkers bear marked resemblances to those employed by Berkeley and Hume discussed earlier (Runes 1942).

from those that arise as a result of memoric reflection, for example; both types of which are (Kantian) phenomena. Also, the expression *public* is taken to mean external to the perceiver and *physical* or *material object* as a signifier to mean some *thing* with a non-virtual public existence in matter or energy, such as a table, a cloud or a violin tone.

3.3.4.1 The Sense-Datum theory of immediate perception

In the literature, the term *sense-datum* or sometimes *sensum*[24] is used to denote an immediate un-analyzable private object of sensation; a non-physical entity that actually possess the various sensory qualities that a person experiences. Sense-datum theory argues that the direct or immediate object of an experience is an entity produced at the end of a causal process and is thus distinct from any physical object, if any, that initiated the process. Examples of the veridicality of this distinction include seeing the light from a star that no longer exists, viewing a straight stick that is immersed in water and so looks bent, feeling an itch in a previously amputated limb and hearing the voice of a deceased friend. Epistemologist Laurence BonJour (2008) suggests such immediate experiences can be classified according to the following perceptual qualities:

- **Relativity.** What is perceived has different qualities under different perceptual conditions, even though the relevant physical object does not change;
- **Illusion**. Qualities are experienced that the relevant object clearly does not possess, and
- **Hallucination**. Qualities are experienced in a situation in which there is no physical object of the relevant sort present in the sensory field.

While it argues for the existence of sense-data, sense-datum theory doesn't account for the existent nature of it (its ontology), or the *relation* between it and the experiencing mind. The sense-datum is an object immediately present in experience and has the phenomenal qualities that it appears to have. The natural thing to say is that sense-datum somehow influences the state of mind of an individual in a way that reflects the sense-datum's specific character. It is this resulting state of mind that the adverbial theory describes.

[24] Plural forms are *sense-data* and *sensa.*

3.3.4.2 The Adverbial theory of immediate perception

In contrast to the sense-datum theory, the adverbial theory:

> … has no need for such objects and the problems that they bring with them (such as whether they are physical or mental or somehow neither). Instead, it is suggested, merely the occurrence of a mental act or mental state with its own intrinsic character is enough to account for the character of immediate experience … I sense in a certain manner or *am appeared to* in a certain way, and it is that specific manner of sensing or way of being appeared to that accounts for the specific content of my immediate experience (BonJour 2007). (my italics).

The essential feature of adverbial theory is that there need not be an object or entity of any sort in the material world. They are objects of awareness, modes, or states of the mind and do not exist independent of it. To say that an idea is *an object of my awareness* is just a grammatically convenient way of saying that the idea is *that of which I am aware*.

In comparing the sense-datum and adverbial account of the contents of experience, BonJour thinks that the adverbial account is most likely to be more correct, because

> if sense-data somehow affect the mind in a way that reflects their character, then the resulting adverbially characterizable states of mind are really all that matter, making the sense-data themselves superfluous; and if they do not affect the mind in such a way, then their apprehension by the mind is difficult or impossible to make sense of. … any characterization of sensory experience that can be given in sense-datum terms can equally well be adopted by an adverbial theorist, simply by construing a comprehensive sense-datum description of one's sensory experience as characterizing the specific manner in which one is adverbially "appeared to" (BonJour and Sosa, 2003: 78-79, n3).

So the sense-datum and adverbial theories are just different ways of expressing the same idea, *sense-datum* being the nominal or objective way of expressing the subjective content of sensing, or *being appeared to*. The adverbial form is almost always more unwieldy in English, so for the sake of simplicity I will use the simpler *sense-datum* to imply both, unless a distinction is called for.

3.3.4.3 Representationalism (Indirect Realism)

Indirect, or representative realism is the hypothesis that there is a justification for believing that our immediately experienced sense-data, when taken with further beliefs that we arrive at on the basis of these sense-data, constitute a representation or depiction of realm of material objects independent of our sensing of them.

One striking contrast between the representative realist's explanatory hypothesis and the others considered here, is that under the representative realist view there is a clear intuitive sense in which the qualities of the objects that explain our immediate experience are reflected in the character of that experience itself in such a way that these, albeit indirect, experiences can be said to be of the qualities of the objects.

3.3.4.4 Phenomenalism[25]

Phenomenalism is a theory of a perceptual subjectivity that maintains that the characteristics and relations of sense-data is *all* that constitutes the content of propositions about immediate (unreflective) perceptual experiences of public material objects. The theory, more correctly labeled *ontological phenomenalism*, grew as a radical form of empiricism with roots in Berkeley and Hume's subjective idealism, as discussed earlier, and developed by Ernest Mach in the nineteenth century, to be later refined by Bertrand Russell and the logical positivists (Tarnas 1991: 383), and not confused with Kant's *epistemological phenomenalism* which doesn't deny the existence of objects not experientially knowable (*noumena*). Ontological phenomenalism maintains that to believe that public material objects exist is to believe that various sorts of sense-data have been, and/or would be experienced under certain specifiable physical conditions, such as those that would intuitively permit the public objects to be perceived; the (relatively) *permanent possibilities of sensation* as John Stuart Mill expressed it (Mill 1865: 225-232). Phenomenalists offer no reasons for *why* such sense-data is *permanently possible*, and on the assumption that it would be immediately available to the perceiver if they happened to be there then, they do admit sense-data that is confined to a specific time and place.

Because the claim of phenomenalism is that the content of propositions about perceptual experiences of public material objects is given *entirely* in terms of sense-data, the sense-data and of those environmental factors that are aspects of the order of the immediate experience must be premised by other sense-data. For example, being able to assert that there is a violin in the room from actual or obtainable sense-data from a violin in the room is premised on sense-data of the room, as well as the violin because the room does not exist to the perceiver external of sense data. The need for sense-data to be

[25] Phemonenalism is to be differentiated from Phenomenology, a philosophical movement initiated by Edward Husserl, as discussed earlier.

specifiable only in terms of other sense-data, as the phenomenalist position seems to imply, leads to an infinite regress, so it fails as a theory of knowledge. Further, an offer of a specified route to the location of the violin (sense-data), would require a guarantee that such a sensory route exists to that location, difficult enough in present ('now') time, but unfathomable about objects and events in the (distant) past. BonJour reports Roderick Chisholm's generalized argument that

> there is in fact *no* conditional proposition in sense-datum terms, however long and complicated the set of conditions in the "if" part may be, that is *ever* even *part* of the content of a material-object proposition. This is shown, he claims, by the fact that for any such sense-datum proposition, it is *always* possible to describe conditions of observation (including conditions having to do with the state of the observer) under which the sense-datum proposition would be false, but the material-object proposition might still be true (BonJour 2007).

A deeper problem is that our sense-data are obviously not random but it is not clear *how* they are ordered without any reference to public material objects. Lastly, it seems that, as the above discussion implies, phenomenalists, by only being able to infer knowledge of public objects by their own immediate experience of them, put themselves in the untenable solipsist condition that they cannot know of the existence of other minds or mental states. In addition, Kant's refutation of material idealism (§3.3.3.2), showed this position to be false.

3.3.4.5 Direct Realism and ecological psychology

The application of physicalism to psychology was the logical basis for the method known as of behaviorism and was used to support a theory of perception known as Direct Realism in which representations or ideas are not thought of as being themselves the immediate objects of awareness, but instead as directly constituting the act or state of awareness itself. Hence, at least in the case of veridical perception, the immediate object of awareness is the external thing itself and not a representation of that thing.

> Cognitive states [i.e., representations], are not cognitive relations with objects, nor are they themselves peculiar objects supposed to mediate the occurrence of cognitive relations. They are simply the perceiver's awareness of possible objects.
>
> The immediate object of awareness is always the ordinary object and not some special object, and that therefore, for example, 'Intuitions . . . are the immediate awarenesses of . . . ordinary objects', rather than themselves objects of awareness (Aquila 1983: xi).

The most important immediately preceding realist philosopher was the Scot Thomas Reid (1710-1792) who wrote with extreme common sense in trying to refute his countryman David Hume's scepticism. (Gregory 1981: 349-351). The psychologist James Gibson (1966; 1979) developed Direct Realism into a theory of perception founded on the understanding that the senses had developed in an environment that thus latently afforded actions, whether or not these affordances were recognised. In a thorough analysis of the metaphysical roots of Gibson' psychology, Lombardo notes that

> …over a 50-year period he came to challenge both mind-matter dualism in his ecological theory and the epistemology of indirect perception in his direct realist philosophy of perception. Gibson's theory of ecological reciprocity avoids both the absolute philosophical dichotomies inherent in Platonic thinking, and the one-sided treatments of reality in monistic philosophies. … In examining the growth of Gibson's ecological psychology, one can find numerous roots. Aristotle, as a single theorist, probably anticipates Gibson more than any one, and in modern times, functional psychology, process ontology, Gestalt wholism, and the evolutionary-ecological view of nature have all influenced Gibson's thinking. (1979: 4).

Gibson is one of the few people to explicitly defend a näive realist view of perception, however he was forced to make seemingly implausible assumptions about the perceptual process in its defence, including a denial of the general materialist view that the sensory organs transmit sensory information into the brain, where neurophysiological processes compute a perceptual representation of the external world. Instead, Gibson suggested that perception occurs somehow out in the world itself, rather than in the physical brain. Exactly how this occurs, or what this actually means however, he could never explain very satisfactorily.

While direct realism may be a useful model for some perceptual mechanisms, it does not seem to be a plausible model of the complete experience of material objects, and so cannot be the basis of a justification for believing that our immediately experienced sense-data constitute a representation or depiction of an independent realm of material objects.

3.3.5 Information as relations through signs

Fallibilism is the term the philosopher and scientist and founder of pragmatic semiotics Charles Peirce (1839-1914) used to describe the unreliability of such empirical methods to provide meanings truthfully stronger than probabilistic propositions outside of the mind:

> …the doctrine that our knowledge is never absolute but always swims, as it were, in a continuum of uncertainty and of indeterminacy. Now the doctrine of continuity is that *all things* so swim in continua (Peirce 1897).

One consequence of the exploration of idealism was the understanding that, although sensations are experienced immediately, sense perceptions are not ideas and they do not give an instant knowledge of things; they are simply non-cognitive natural events that are neither true nor false in–and–of–themselves. Sense perception only becomes informatively meaningful when it stands for, becomes a sign of, something more than or other than itself in some respect or other for somebody, as when the perception of a high pitched repeated tone comes to signify *bird* to a human in one context and *alarm-clock* in another. Such hypothetical inferences were labeled *abductions* by Peirce: "[T]he content of consciousness, the whole phenomenal manifestation of mind, is a sign resulting from inference" (1868)[26]. Peirce developed a *pragramatical*[27] approach to the study of semiotic frameworks–the relationships between signs and their impacts on those who use them.

Following Kant's *Categories,* Peirce developed a system of three existential Ceno-Pythagorean[28] categories:

- Firstness. Reference to a ground (a ground is a pure abstraction of a quality). Essentially monadic; Informally: Quality of feeling.
- Secondness. Reference to a correlate (by its relate). Essentially dyadic; Informally: Dependence.
- Thirdness. Reference to an interpretant. (An interpretant is the product of an interpretive process, or the content of an interpretation.) Essentially triadic; Informally: Representation.

According to Peirce, a sign is something that stands for something else in some manner or other for somebody. Thus, the sign relation is triadic involving

- A *causal relation* between a sign user and something (an Object) that stands for something else (a sign, or in Peirce's terminology, a "Representamen");

---

26 For a fuller explanation of abductive inference, see Appendix 1.
27 A term Peirce invented to distinguish it from the more widely used *pragmatic*.
28 The prefix *ceno-* is from the Greek word *kainos*, which means "new" or "recent" and Peirce calls them 'Pythagorean" because they have to do with number.

- A *semantical relation* between the sign and the something else it stands for; and
- A *pragmatical relation* between the sign and the thing it stands for, and the user (an Interpretant)–that is, the sign in the mind that is the result of an encounter with a sign.
- Peirce distinguished three different ways in which things might stand for other things in some respect or other;
- *Iconic relations*, in which things resemble other things; auditory icons and spearcons, for example;
- *Indexical relations*, of causes and effects; earcons, for example, and
- *Symbolic relations*, in which things are associated in essentially arbitrary ways; sounds and the letters of the alphabet, for example.

While iconic and indexical relations exist in nature whether or not anyone notices them, they can only function as signs when the relationships by which they can be associated is noticed by someone as standing for some associated properties of other things. For example, words such as "wind" and "pipe" do not resemble the forms of moving air or a cylindrical tube for which they stand, so the words are not icons; tree rings are an effect of the aging of tree but can only act as an indication of (an indexical sign) of this aging once the relationship has been observed.

Peirce's semiotics is dependent on his deep understanding of phenomenal experience. It is useful to recognise that he thought that all processes of consciousness, including perceptual consciousness, involve or are sign processes, that is, are relational. According to Innis, Peirce's central idea is that "the content of consciousness, the entire phenomenal manifestation of mind, is a sign resulting from inference…" (Pierce 1868: 53). Peirce emphasises that he thought "Thirdness, [that is law-governedness] pours in upon us through every avenue of sense…*There can be no perceptual object without a unifying factor that distinguishes it from the 'play of impressions'*" (Innis 1994: 13) [my italics].

This is an important understanding with respect to the perception of immanent abstract objects, such as the mental impressions resulting from the sonification of multivariate datasets that have no direct perceptual correlates in the physical world. If it is necessary for such objects to have such a unifying factor, it bears some consideration how sonification can offer any possibilities in that regard.

3.3.6 Information in networks and connections

Cognitive science arose as an interdisciplinary research endeavour in the early 1960s around the idea that the brain could be identified with hardware, and on which cognition was computed. Because of its physical approach, direct realism has been very influential in the development of the application of neural networks and other connectionist models, in pursuit of environment-sensing robots capable of automatic navigation, for example. This has proved remarkably successful up to a point. In its broadest interpretation, cognitivism considers that the mind is rational, autonomous and independent of the body; ideas strongly valued in Western culture, as evidenced by the long-running fascination with talking dolls, robots and other automata (Wood 2002). However Cognitivism's identification of the individual as central to thought is currently being challenged by the growing body of evidence from artificial intelligence research that embodiment is important to cognition, perhaps more important than vision (Varela, Thompson and Rosch 1991, Hutchins 1995); that different sense modalities provide access to different types of information and even artificially intelligent agents require ontologically mediated schemata, as seminal artificial intelligence researcher Derek Partridge clearly infers. (1991: 171-227).

The connectionist models favoured by the cognitivists was in parallel with the composition modelling experiments of those computing composers who saw the connectionist paradigm as offering "a new and unified viewpoint from which to investigate the subtleties of musical experience. Connectionist systems employ 'brain-style' computation, capitalizing on the emergent power of a large collection of individually simple interconnected processors operating and cooperating in a parallel distributed fashion" (Todd and Loy 1991: ix).

Though it is still a very active field, *cognitivism*, no longer enjoys the limelight it once did. Computer scientist and editor of the *Journal of Consciousness Studies*, Joseph Goguen provides a succinct, if somewhat unbalanced, critique of cognitivism and its relationship to the emerging field of consciousness studies (2002). As the approach continues to fail to deliver results beyond the elementary, there is a growing awareness that it will not turn out to be the hoped–for panacea. Goguen thinks the next step, developing a computational model of conscious awareness, will not be achievable without a more sophisticated model of mind. David Chalmers, currently one of the leading philosophers of consciousness agrees:

> The easy problems are those of finding neural mechanisms and explaining cognitive functions: the ability to discriminate and categorize environmental stimuli, the capacity to verbally report mental states, the difference between waking and sleeping. The hard problem is that of experience: why does all this processing give rise to an experienced inner life at all? While progress is being made on the easy problems, the hard problem remains perplexing (1992).

3.3.7 An attempt at integration

Of the various attempts to integrate the three main threads: Idealism, Realism and Information Theory, probably the most widely known is the work of the British anthropologist and semiotician Gregory Bateson (1904-1980), who coined the definition of information as "the difference which makes a difference." Bateson adopted Alfred Korzybski's concept that "the map is not the territory":

> But what is the territory? Operationally, somebody went out with a retina or a measuring stick and made representations which were then put on paper. What is on the paper map is a representation of what was in the retinal representation of the man who made the map; and as you push the question back, what you find is an infinite regress, an infinite series of maps. [T]he process of representation will filter [the territory] out so that the mental world is only maps of maps, ad infinitum. All 'phenomena' are literally 'appearances' (Bateson 1972).[29]

Bateson considered that the great dualist dichotomy of epistemology has shifted under the impact of cybernetics and information theory and that with the discovery of cybernetics, systems and information theories was a formal base "enabling us to think about mind and enabling us to think about all these problems in a way which was totally heterodox from about 1850 through to World War II" (op.cit.). In discussing the origins of "the map is different from the territory", he emphasised that the idea came out of ancient Greece:

---

29 The original source of the quotation is from Bateson's Nineteenth Annual Korzybski Memorial Lecture entitled *Form, Substance and Difference*, delivered January 9, 1970, under the auspices of the Institute of General Semantics. and printed in the *General Semantics Bulletin*, No. 37, 1970. It was republished in Bateson (1972)

> It all starts, I suppose, with the Pythagoreans versus their predecessors, and the argument took the shape of "Do you ask what it's made of—earth, fire, water, etc.?" Or do you ask, "What is its *pattern*?" Pythagoras stood for inquiry into pattern rather than inquiry into *substance*. That controversy has gone through the ages, and the Pythagorean half of it has, until recently, been on the whole the submerged half. The Gnostics follow the Pythagoreans, and the alchemists follow the Gnostics, and so on. The argument reached a sort of climax at the end of the eighteenth century when a Pythagorean evolutionary theory was built and then discarded—a theory which involved Mind. (op.cit).

He does not mean imply by this that he is supports traditional Cartesian dualism but a new approach in which the

> individual mind is immanent but not only in the body. It is immanent also in pathways and messages outside the body; and there is a larger Mind of which the individual mind is only a subsystem. This larger Mind is comparable to God and is perhaps what some people mean by "God, but it is still immanent in the total interconnected social system and planetary ecology (op. cit.).

Bateson's overt aim is one of raising his listener's awareness of ecological issues, so perhaps his primary concern is best described as 'functional epistemology'. Suggesting that the resolution lies in the God Mind is a familiar way of collapsing the unresolved mystery back into itself and further emphasises that epistemology is a process of thought, less directed to providing conclusions than it of exploring a mental landscape, of organizing beliefs and experiences into holistic descriptions.

3.4 Summary

The relationship between our sensing of a variegated world and the way we interpret it has been a major theme in Western philosophy since its recorded beginnings. For Plato, all experiences of the world were understood by the extent to which they conformed to pre-existent Ideas that were 'received' at birth and ever-present in the mind of the perceiver. Aristotle's dissatisfaction with this 'unworldly' view led him to speculate about, and importantly experiment with, the structure and function of the world as he experienced it. On regaining Grecian scholarship from the Near East following the European Dark Ages, the Scholastics, began to question the received wisdom of the church, which many identified as a form of misplaced Platonism. For the sake of brevity, the insights of this period were not discussed in this chapter, important though some of them may turn out to be for an understanding the relationships between emotion and awareness that are currently engaging neuroscientists (Damasio 1995; 2003).

The uncertainty of obtaining reliable knowledge of the world through direct sensory experience that the idealists had exposed, undermined the Cartesian confidence in human reasoning about the perceived world as the foundation for truth and sent philosophers in various directions, including a search for new ways to underpin traditional rhetorically-based logic with mathematical foundations. At the same time as Hume demonstrated the unreliability of the senses and thus the unknowableness of the world, empirical methods were being used to increase knowledge *of* that world, and obtain power over its natural forces. Kant's resolution of this apparent conflict, by demonstrating the special relationship between space, time and perception, which he called *Transcendental Idealism*, was a lens through which later philosophical investigations were obtained.

According to Kant's understanding, what, exactly, is meant by *information* is embedded in relationships between the sensation, perception and apperception of phenomena; what he called *appearances*:[30] things as they are for humans, as opposed to things as they are 'in–or–of–themselves'. From this perspective, information can be simply characterized as phenomena, or thoughts about phenomena in the mind of some person. Brentano and Husserl developed their pure (contemplative) psychology of perception, known as phenomenology, by bracketing off the 'world–as–we–know–it', Kant's 'things–as–they–really–are' (*Ding an sich*, *noumena*), to try to determine whether or not the properties of phenomena were capable of being formally (that is logically) organized; firstly in the mind of the perceiver and secondly as sharable with others—a characteristic Husserl called *intersubjectivity.* His aim was to develop an *eidetic* science of essential, invariant phenomenal forms that involves no assertion of actual material existence, but he struggled to keep them conceptually separate from Plato's Ideas. Peirce thought that there can be no perceptual objects without a unifying factor that distinguishes them from the 'play of impressions'".

The success of methodologically rigorous empirical approaches in the natural sciences wedged the study of the psyche away from its purely first-person introspective philosophical roots. The most significant early discovery that resulted was that of perceptual gestalts, which their discoverers considered as phenomena arising naturally and directly from the physical

---

30 *Erscheinungen* in German.

nature of sensation. Though the Gestaltists had found an important invariant, they were not able to extend the idea past the noumenal world. However, it is a characteristic of phenomenal forms (such as melodies) that their properties can remain unchanged when the objective stimuli upon which they rest undergo certain modifications (such as transposition). This phenomenon of identity is related to a much more general problem found in abstract mathematics; of invariances with respect to variations of the primitive elements out of which a form is constructed. The particular kind of identity that is attributable to apparently heterogeneous figures, because they can be transformed into one another using operations that define a group, exists in the domain of perception and permits us to grasp perceptual "structures". The mathematical concept of transformability corresponds to the concept of transposability in perception. So by accepting "form" as a primitive concept, Gestalt psychology made an attempt to free psychological theory from contingency on the mere mosaic of perceptions.

Without considering cultural differences[31], not all group-theoretic transformations of perceptual objects are equally cognised, nor are the same transformations as easily perceivable in different sense modalities. For example, symmetry group transformations of pitch and temporal structures, such as transposition, inversion and retrogradation, occur frequently in music though they seem not to be all equally evident to the casual observer; under non-extreme pitch transposition and tempo acceleration a melodic structure remains strongly invariant; pitch contour inversion and rhythmic retrogradation are common occurrences but not as strongly invariant, while rhythmic inversion seems not to be perceptually invariant or even generally defined.

Gibson reacted to the findings of perceptual gestalts by denying the existence or importance of the phenomenal. Instead, he based his direct realism theory of vision on an evolutionary approach whereby organisms and their environments develop reciprocities over genetic and diurnal time; their

[31] Which needs to be done cautiously. The structural characteristics of music from a wide variety of cultures seem generally comprehensible, suggesting that cultural differences are more likely to be of degree rather than kind. Assumptions need to be treated sceptically, however. Werker and Vouloumanos (2001) document many studies that indicate speech plasticity in infants is subject to cultural influences and in a classic study, Segall Campbell and Herskovits (1966) showed that the Müller-Lyer illusion is culturally not neurophysiologically determined.

senses develop in an environment that thus latently afforded actions, whether or not these affordances were recognised. While direct realism may be a useful model for some perceptual mechanisms, it does not seem to be a plausible justification for believing that immediately experienced "sense-data" constitute a representation or depiction of an independent realm of real material objects.

The connectionist/cognitivist approach has been to build sensing, then responding, then perceiving, then environmentally aware machines, in the hope that, given enough neural connections, zombies would appear that could function in the world around them with intent.[32] These models have failed to bridge the divide between the easy and the hard problems of consciousness (§3.3.6) and so researchers are beginning to explore other models of mind and the rôle emotions plays in intent.

If it is the *intended* effect that ultimately identifies information, as Küppers suggests (§3.2.2), and that intent is *by* the listener *of* a sonification, understanding what the principles are by which sonifications of abstract multivariate datasets can be made to reliably afford such intent, seems to require more, or something other, than what psychology, psychophysiology and the cognitive sciences have so far revealed about the processes involved. Yet music somehow continues to communicate non-verbally and so the next chapter explores some non-rational, even non-mental, aspects of music with a view to considering their possible usefulness for the intentional user's perception of immanent abstract phenomenal objects.

---

32 According to Chalmers, "A zombie is physically identical to a normal human being, but completely lacks conscious experience. Zombies look and behave like the conscious beings that we know and love, but all is dark inside. There is nothing it is like to be a zombie." http://consc.net/zombies.html.

Chapter 4

AN INTERMEZZO: SOUNDS AND SENSE

4.1 Introduction

At a time when computers can process massive amounts of data more impartially that humans, and cognitive science offers the elixir of computed intelligence, it may appear somewhat ironic that, not only do researchers continue to rely on wetware[1] for investigating the 'soundness' of computed results but there is a growing appreciation that audition is not a second cousin to vision, only to be brought into play when vision is unavailable or already overstrained (Bregman 1994: 1).

Data sonification is arguably the most abstract form of auditory display and historically, music is probably the most complex form of abstract aural communication made by humans. However, until very recently in human history, when recording technology afforded the production of disembodied sounds, all forms of music involved the ongoing production of corporeally-controlled interactions of physical, material objects to produce coherent sonic streams.

This short chapter, by way of an intermezzo to the consideration of sonification software, is an attempt to address in a more discursive way the question, if sonification software is to meet and anticipate the needs of sonifiers in the future, what sorts of problems will it be required to address?

4.2 Three ways of thinking about sound

The three principal ways of thinking about music can be found in three philosophers in ancient Greece in the fifth century BCE: Pythagoras, Aristoxinos and Aristotle.

---

1 Described by Merriam-Webster Online as a human being considered especially with respect to human logical and computational capabilities. The Wikipedia entry describes it as a term used to describe the embodiment of the concepts of the physical construct known as the central nervous system (CNS) and the mental construct known as the human mind. It is a two-part abstraction drawn from the computer-related idea of hardware or software. Both Accessed 23 August 2008.

4.2.1 Numerical rationality

Pythagoras' approach to music is the most well–known of the three. It was a numerically rational analysis of the way string lengths and sounds relate to each other physically, that is *acoustically*, and importantly, he classified the sensation of harmoniousness according to these ratios.

4.2.2 Empirical experience

Aristoxenos' approach was more concerned with the structure of the listening *experience*, which he explained in terms of the various modes[2]. A pupil of his musician father and the later Pythagoreans who were keen on distilling their inherited scientific knowledge from its more mystical entrapments, Aristoxinos' own writings on music are somewhat empirical, perhaps influenced by Aristotle, with whom he also studied. He maintained, in contradistinction to the Pythagoreans, that the notes of the scale could be tuned by the ear rather than ratio measurement, and formulated a theory that the soul is related to the body as harmony to the parts of a musical instrument[3]. Here, for example is his description of vocal pitch inflection such as glissandi:

> The continuous voice does not become stationary at the "boundaries" or at any definite place, and so the extremities of its progress are not apparent, but the fact that there are differences of pitch is apparent…; for in these cases we cannot tell at what pitch the voice begins, nor at what pitch it leaves off, but the fact that it becomes low from high and high from low is apparent to the ear. In its progress by intervals the opposite is the case. For here, when the pitch shifts, the voice, by change of position, stations itself on one pitch, then on another, and, as it frequently repeats this alternating process, it appears to the senses to become stationary, as happens in singing when we produce a variation of the mode by changing the pitch of the voice. And so, since it moves by intervals, the points at which it begins and where it leaves off are obviously apparent in the boundaries of the notes, but the intermediate points escape notice… (Vitruvius 1914: ch 4).

4.2.3 Expressive power

Aristotle was interested in the ability of sound (music and poetry), to *express* states of mind and evoke these states in the soul (mind) of the listener.

> …for when we hear [music] our very soul is altered; and he who is affected either with joy or grief by the imitation of any objects, is in very nearly the same situation as if he was affected by the objects themselves; ... now it happens in the

2 Xenakis (1971: 183-189) has a more detailed explanation of Aristoxinos' modal thinking.
3 Vitruvius. *De architectura* (Book V Chapter IV) contains a paraphrase of an extant fragment of a treatise on metre in writings on music in attributable to Aristoxenos.

> other senses there is no imitation of manners; ... for these are merely representations of things, and the perceptions which they excite are in a manner common to all. Besides, statues and paintings are not properly imitations of manners, but rather signs and marks which show the body is affected by some passion. ... But in poetry and music there are imitations of manners; ... those who hear them are differently affected, and are not in the same disposition of mind when one is performed as when another is; the one, for instance, occasions grief and contracts the soul, ... others soften the mind, and as it were dissolve the heart: others fix it in a firm and settled state, ...fills the soul with enthusiasm... The same holds true with respect to rhythm; some fix the disposition, others occasion a change in it; some act more violently, others more liberally (Politics, VIII:V).

4.3 Disembodiment

Descartes' disembodied mind is a potent symbol of the armchair philosopher passively observing the world. "Sensing bodies" were considered as convenient mechanical devices to support thinking minds. While he mentions the three Greek approaches in his *Musicae Compendium* (1618), most of Descartes' effort was applied to the scientific study of the perception of objects as if by a disembodied mind (*res cogitans*). This resulted in his downgrading of the corporeal origins of musical sounds in favour of their more objective characteristics. In his *Meditations Métaphysiques* (1641), he explains that since objects have extension and can be put into motion they can be measured, however he is not sure whether qualities of objects like sounds and colours could be known:

> And in regard to the ideas of corporeal objects, I do not recognise in them anything so great or so excellent that they might not have possibly proceeded from myself; for if I consider them more closely, … I find that there is very little in them which I perceive clearly and distinctly. Magnitude or extension in length, breadth, or depth, I do so perceive; also figure which results from a termination of this extension, the situation which bodies of different figure preserve in relation to one another, and movement or change of situation; to which we may also add substance, duration and number. As to other things such as light, colours, sounds, scents, tastes, heat, cold and the other tactile qualities, they are thought by me with so much obscurity and confusion that I do not even know if they are true or false, i.e.whether the ideas which I form of these qualities are actually the ideas of real objects or not [or whether they only represent chimeras which cannot exist in fact] (Descartes, 1641 : VI).

This argument, that the unreliability of the senses made them not very suitable for building up reliable knowledge, was the subject of the Material Idealists, as discussed in §3.3.2. So, as empirical approaches to knowledge of the external world were gaining respectability, sounds were considered, like colours, as secondary properties of material objects (after Locke). Emphasising the development of objective physical knowledge of sound over its psychological effects turned many musical issues such as tuning and temperament into targets for solutions based on logic and reasoning, the natural domain of the Cartesian mind. This had the consequence that investigation into the emotional and gestural influences, so apparent to Aristoxenos and Aristotle was neglected; it also suppressed their conscious import on musical discourse, restricting research in sound perception to 'brain–in–head' effects.

This restriction continued through the nineteenth century in experimental psychology laboratories of Wundt and Stumpf, as already discussed in §3.3.3.5. The underlying assumption in Helmholtz' psychoacoustical studies (1863) is that the sensation of *music* is based on the physiological sensing and information extraction mechanisms in the human ear, and the discoveries of representational Gestalts in perception seemed to emphasise that sensibility was achieved more through direct sensation–brain––connectivity[4] than through the experience and attention of embodied individuals.

4.3.1 Abstraction: symbolic & sub-symbolic modelling

The rise in the importance of the written score in defining musical 'works' is well documented (Goehr 1994; Attali 1985). Building on this notated representational perspective of music, the computer, assisted by MIDI-enabled keyboards for easy data input, afforded a more data–processing approach to such extra-spectral aspects of music such as melody and rhythm (Cope 1991).

In–keeping with the attempt of a burgeoning cognitive science to use computers to develop an 'artificial intelligence' which mimics human intelligence, it continued to emphasise the *res cogitans* (thinking thing) approach to musical cognition under the assumption that this was where the holy grail for musical sensibility lay (Balaban, Ebciöglu and Laske 1992).

The limited success of this approach was causal in developing sub-symbolic modeling techniques for connecting the symbolic approach to the spectral properties of sound (Kohonen 1995), aided by the stochastic techniques of the connectionists in accounting for the material constraints of musical instruments (Todd and Loy 1991). While closer to a more ecological

[4] That is, psychoneural connectivity.

approach than the previous techniques discussed, it still focused exclusively on a disembodied *res cogitans* approach to the perception and sensibility of musical sound.

4.4 The en*light*ened way

Allied to the reduction of music to noises–in–the–head was the growing dominance of visual terminology to describe aural phenomena. In his introduction to the seminal *Handbook for Acoustic Ecology*, Barry Truax provides a contemporary perspective on this occidental bias. Of the dominance of visual terminology in describing aural sensations, he writes:

> No field of study based on sensory experience seems to be overburdened by terminology to the same extent as that dealing with sound and hearing. … Terms such as perspective, foreground, background, colour, spectrum, shadow, focus, image, reflection, transparent, translucent … have found public familiarity in a way that is hard to imagine their sonic counterparts ever matching. … [T]his paradox reveals the tendency of our culture to trade its ears for its eyes (Truax 1978: v).

This privileging of the visual over the aural is implicitly supported by the sense-datum theory of perception discussed §3.3.4.1, because it accounts for immediate perceptual experience as *acts* of awareness of *objects* that such acts apprehend; When one *observes an object*, one "*watches over* a *thing aimed at*[5]. Sometimes this privileging functions to mask the unwarrantedly prejudicial, as for example in this first paragraph of *The Stanford Encyclopedia of Philosophy*'s entry on sound:

> Possibly, the philosophical privilege of the visible just reflects the cognitive privilege of the visible—as vision is considered to account for most of useful sensory information gathering. (Casati and Doric 2008).

4.4.1 Objectification

In reviewing the extensive body of literature used in chapter 3, the extent to which arguments about sensation and perception are constructed around, or exemplified by, the visual perception of spatially physical three-dimensional objects is overwhelming. The implication is that the arguments made about vision ostensibly apply equally to the other senses, and it is left up to the

---

5 From L. *ob*-"over"+ *servare* "to watch, keep safe" and *ob*-"against"+ *jacere* "to throw," as in a jet. [OLED]

reader to induce them. Unfortunately, the problems that arise from such generalisations are embedded in language and culture.

Some researchers consider the use of the term *object* to signify a sound, as conceptually inadequate. In his seminal study of auditory scene analysis, for example, Bregman restricts the term to the visual domain and prefers the expression “auditory stream” as “our perceptual grouping of the parts of the neural spectrogram that go together … the perceptual unit that represents a single happening”. In adopting *stream* he remarks that

> the word “sound” refers indifferently to the physical sound in the world and our mental representation of it. It is useful to reserve the word “stream” for a perceptual representational and the phrase “acoustic event” or the word “sound” for the physical cause. …
>
> The stream plays the same role in auditory mental experience as the object does in visual (Bregman 1994: 9-11).

While Bregman makes the uncontroversial phenomenal physical and representational distinction, his reference to the sound itself as a physical causal event is ambiguous, both in terms of the location and material composition of this “sound in the world“, and whether the causal predicate is the physical “sound in the world“ or the perceptual event “stream”. Further, the tautological description of the origin of the sound as the sound source, does not resolve the veiled ontology of the sound itself. Bregman’s work is concerned with the primitive perceptual organization of sounds, and it remains to be determined whether such perceptual organisation is affected by the omission of the consideration of the ontological nascence of either physical sounds or the auditory streams themselves.

4.5 The body returns

Husserl's pupil Martin Heidegger (1889-1976) was critical of the subject/object split that pervades the Western tradition as is indicated by the root structure of Husserl and Brentano's concept of intentionality, i.e., that all consciousness is consciousness of something, nor are there objects without some consciousness beholding or being involved with them. He encompassed terms such as ‘subject’, ‘object’, ‘consciousness’ and ‘world’ into the concept of "being-in-the-world" (*In-der-Welt-sein* or *Dasein*) and crucially, distinguished between the "present-at-hand" and the "ready-to-hand". Present-at-hand knowledge roughly corresponds to posivitist knowledge; that required for understanding and navigating the environment–measurement, size, weight, shape, cause & effect etc. Ready-to-hand is characterized as:

> . . the kind of dealing which is closest to us, not a bare perceptual cognition, but rather that kind of concern which manipulates things and puts them to use; and this has its own kind of 'knowledge'. (Heidegger 1927/1962: 95).

In other words participatory, first-hand experience: familiarity, tacit know-how, skill, expertise, affordance, adaptability etc. Heidegger argues that our scientific theorizing of the world is secondary and derivative and he exposes an ontology that is far broader than the dualistic Cartesian framework. He stresses the primacy of the readiness-to-hand, with its own kind of knowing or relating to the world in terms of what matters to us.

It follows, from Heidegger's perspective, that human action is embodied, that human knowing is *enactive*, and participatory. Hubert Dreyfus (1972) used these ideas to criticise artificial intelligence, as discussed in §3.3.6, and to develop a learning model for physical skills.

The Hungarian scientist and philosopher, Michael Polanyi (1891-1976) proposed a type of participative realism in which personal knowledge plays a vital and inescapable role in all scientific research, indeed, in all human knowing.

> Let us therefore do something quite radical . . let us incorporate into our conception of scientific knowledge the part which we ourselves necessarily contribute in shaping such knowledge (Polanyi 1975: 28-9).

By stressing the tacit nature of participatory knowing, Polanyi is claiming that "we know more than we can tell." In this way he is emphasising knowledge that is implicit to a tasks (e.g. know-how, skill), to situations (e.g. navigation) and perspectives (e.g. beliefs), etc. He uses the term 'tacit knowledge' to refer to those things we can do without being able to explain how, with the way the 'body' enables one to find one's way through the world in the absence of explicit rules or calculative procedures. The "indwelling" nature of tacit knowledge is important in the development of the skill of reflexivity, such as needed in the sifting through and interpretation of qualitative data.

There seems little point in speculating whether or not listening will ever regain the relative importance to humans it enjoyed prior to the European En*light*enment, but there are signs of a growing recognition that the resolution of the mind/body dilemma will not be solved by dispensing with the body. In many ways, the tradition of emphasising disembodied cognition over alternative approaches has never really been totally applicable to musical sensibility. The idea that musical involvement is based on the embodiment of movement and the bodily sensing of music has a long history, of which the traditional connection between dance and music is but a gross example.

Truslit studied the body movements of musical performers and suggested they were articulations of inner movements in the music itself. (Repp 1993). Central in Truslit's approach to musical movement are the notions of dynamics (intensity) and agogics (duration). If the music has the dynamo-agogic development corresponding to a natural movement, it will evoke the impression of this movement. He makes a distinction between rhythmic movement and the inner movement of the music. In contrast to rhythmic movement, which is related to individual parts of the body, the inner movement forms the melody via the vestibular labyrinth of the inner ear and is related to the human body as a whole. Both Nettheim (1996) and Clynes (1977) also make a connection between music and gravitational movement, based on the idea of a dynamic rhythmic flow beyond the musical surface.

4.5.1 Embodied cognition

Interestingly, the early criticism of the disembodied approach to the development of artificial intelligence came from inside cognitive science, in particular from philosophers (Varela, Thompson and Rosch 1991; Charmers 1995) who stress the phenomenological and embodied aspects of cognition from two perspectives. The first was that it neglects the active relationship between the perceiver and their environment and the second, that it does not take into account the activity of body components, such as eye movement in visual perception and head movement in auditory perception. (Gibson 1966; 1979; Lombardo 1987).

It is also interesting to remember that Husserl was not convinced that perception could eliminate the physical:

> The idea of a phenomenological psychology encompasses the whole range of tasks arising out of the experience of self and the experience of the other founded on it. But it is not yet clear whether phenomenological experience, followed through in exclusiveness and consistency, really provides us with a kind of closed-off field of being, out of which a science can grow which is exclusively

> focused on it and completely free of every thing psychophysical. … Ultimately the great difficulty rests on the way that already the self-experience of the psychologist is every where intertwined with external experience, with that of extra-psychical real things. (Husserl 1927/1971 §3)

A new movement-encompassing action-based approach to the relationship between sound and sensibility began in the 1980s (Cumming 2000). Other than empirical induction, the methodologies include approaches such as abductive inference (§A1.2.6.3) and are contributing to new perspectives on how to approach the relationship between and sensibilities (Varela, Thompson and Rosch 1991; Maturana and Varela 1987). In some ways this can be seen as a return to the Aristotelian integration of sound and sensibility through *mimesis* and related to the Kantian problems of openness and *endness* in the containment of beauty in formal structures and the empathic relationship within them through movement and action (Kant 1790).

4.5.2 Where is the body in sonification?

Irrespective of the sensation of (private, subjective) experiences that sounds produce in listeners, the sounds themselves clearly originate external to them and can be explored objectively to arrive at descriptions of the physical laws they obey. The sensation of sounds is clearly mediated through hearing physiology and this too can be explored empirically to arrive at an understanding of psychophysical principles involved. However, listeners are clearly more than Cartesian brains–in–vats and the perception of the immanent perceptual objects of music clearly involves more than the medium of transmission, that is, sound. For example, in an experimental study of gestures, subjects of various ages were able, with a high degree of accuracy, on only hearing different individual human's walking and running on various kinds of surfaces, to determine their sex (Bresin and Dahl 2003).

The broad ontological issues are dualistic ones: relationships between body and mind, self and other, objects and processes. They inject themselves into this present study as questions about the nature of the existence of perceptual objects and how they relate to the way listeners with bodies perceive patterns in sensations and integrate them, in conscious and unconscious ways, into their existing knowledge schemata[6].

---

[6] The term *schemata* is used here in the same sense as it was used by Piaget (1926).

Aristotle didn't include proprioception and vestibulation in his list of senses, which now should probably be augmented to at least seven. Peirce thought that there can be no perceptual object without a unifying factor that distinguishes it from the "play of impressions".

In a general way, how knowledge is acquired and validated is intertwined with the ontology of the knowledge seeker, the object of their search and the environmental contexts in which the activity occurs. The relatively recent empowering of the cognitive sciences with digital computing machines has considerably enlivened the mind-body debate, particularly with respect to the nature of consciousness, or at least some form of self-awareness (Partridge 1991).

While physical models produce convincing syntheses of individual instrumental sounds there seems to be no such model of the listener. D'haes (2004) suggests a gesture-based account of physical modelling is the way to proceed. In any event, one of the goals of music-related gesture research is to understand the biomechanical and psychomotor laws that characterise human movement in the context of music production and perception (Camurri and Volpe 2004, Camurri et al. 2005). If such an approach is fruitful, as we might hope, there is still some work to be done in generalising the human action link in the following chain to non-human action.

-> human action ->
    -> sound ->
        -> human empathetic movement ->
            -> human sensibility/understanding ->
                -> human action

Perhaps this will become the most demanding task of data sonification software in the future. Pragmatically, the next chapter attempts to address some of the requirements on software before such an approach can even begin.

Chapter 5

THE *SONIPY* SOFTWARE FRAMEWORK FOR DATA SONIFICATION

The need for better software tools was highlighted in 1979 in a comprehensive status review of the field, published as the *Sonification Report* (Kramer et al. 1999). It included some general proposals for adapting sound synthesis software to the needs of sonification research. Now, over a decade later, it is evident that the demands on software by sonification research are greater than those afforded by music composition and sound synthesis software. This chapter compares some major contributions made towards achieving the *Report*'s proposals with current sonification demands and outlines *SoniPy*, a broader and more robust framework model that can integrate the expertise and prior development of software components using a public-domain community-development approach.

5.1 Introduction

To date, software for data sonification has been developed either as standalone applications engineered from first principles, sometimes incorporating third-party low-level audio routines, or as more expansive sonification 'environments' that attempt to encapsulate some general principles and procedures that can be adapted for specific sonification projects as the need arises.

The standalone applications tend to be designed for individual experiments entailing clearly defined tasks such as accurate monitoring (Chafe and Leistikow 2001), or graphic user-interaction (Walker and Cothran 2003), whereas environments tend to be more expansive projects, often with less deterministic outcomes. They afford greater flexibility than is possible within standalone applications. Some recent environments still in development (Pauletto and Hunt 2004; de Campo, Frauenberger and Höldrich 2004) seem to have been designed by first choosing a music composition environment and working backwards, perhaps trusting that the data-processing needs at the 'input end' can be adequately handled by the language tools available from within the particular composition system chosen. Considering that many software tools for music composition have a long gestation period (in the case of *Csound*[1], about forty years, for example) and are still being actively developed, as can be witnessed, for example, by the daily activity on the developer mailing lists for *Csound* and *SuperCollider*[2], this approach is natural and is the approach assumed in the *Sonification Report*.

In addition to the two primarily scripted environments just mentioned, *MAX/MSP*[3] and its younger sibling *PD*[4] are another type, which emphasises graphical user interfaces (GUIs) over program code. Whilst the scripting-versus-GUI debate is still active, it is clear from the large user-base and active development of new patch objects for these platforms that the GUI approach is appealing to some users and perhaps offers a gentler initial learning curve for many exploratory sonification researchers who are visually inclined. In any event, a considerable investment of time is necessary to become proficient in any of these environments and having made the investment, a certain amount of environment 'stickiness' is apparent and understandable.

5.1.1 The First Major Bottleneck: Data

In data sonification, whilst the input data can be thought of as eventually controlling the sound rendering, the transformations it has to undergo in the interim can be considerable. Such data processing can reasonably include multidimensional scaling, filtering and statistical analysis which itself may itself become the subject of sonification. Also, each input dataset can have potentially unique structural characteristics. Some, such as EEG data, may be multiple channels of AC voltages with a variety of DC-biases and noisiness as determined by the particular data collection set-up on a particular patient. Others, such as security data flowing from a market trading-engine, will be massively paralleled, metadata embedded and multiplexed into a single 'feed'. Difficulties in using such data are compounded when the it needs to be buffered and streamed in non-real-time as is the need for multiple overlays of time sequences of different temporal compressions.

---

1 http://www.counds.com/
2 http://www.audiosynth.com/
3 http://www.cycling74.com/
4 http://crca.ucsd.edu/~msp/software.html/

High-level tools for processing such data complexities are rarely, if ever, found in computer music environments, and even less likely if the input data is spatial rather than temporal. When such an a environment is the principle sonification tool, a common response to complex data processing requirements is for someone, where possible, to 'bite the bullet' and write data-processing routines in the language of the composition environment itself. This is currently the approach used by *SonEnvir* (de Campo Frauenberger and Höldrich 2004) and *OctaveSC*[5] (Hermann et al. 2006), which both use *SuperCollider* and also the *PD*-based *Interactive Sonification Toolkit* (Pauletto and Hunt 2004)[6]. The following is a recent exchange on the *Supercollider* users list which epitomises the situation exactly:

```
Question: Hi, I'm looking for linear algebra and affine
transformation routines for 2D and 3D vector spaces; is there
any quark or extension that implement stuff like that?
Answer: The MathLib quark[7] maybe has some useful stuff... or it
would be the place to put these.
```

…or it would be the place to put these implies if someone else using *SCLang* hasn't erstwhile written them and made them publicly available. The user has to make a decision as to whether or not they have the required expertise and time to dedicate to implementing them. While *SClang* is a very elegant and powerful composition language that *can* support the development of data-processing solutions, being unique, it lacks the transportability that more general and widely available tools afford. One consequence of this is that, in projects without a dedicated programmer, practical assistance for what are essentially data processing problems is more difficult to obtain. Data is thus often pre-processed using external tools such as spreadsheets and then read from files by the music composition environment; a procedure that, whilst it may be appropriate in 'limited data' experiments, is at best susceptible to data corruption and of no use if the data is coming from a real-time feed or from a dynamic model (Bovermann, Hermann, and Ritter 2006).

This situation may be characterised as 'data SONIFICATION' i.e. the primary focus is on sound rendering whilst input data is constrained so it can be dealt with adequately by the rendering software. The alternative, an emphasis on data-processing tools at the expense of sound rendering

---

5 http://www.sonification.de/projects/sc3/index.shtml/

6 PD is Miller Puckette's public release of his Max/MSP.

7 A Quark is a package of SC classes, helpfiles, C++ sourcecode for unit generators and other code.

flexibility ('DATA sonification') is no more attractive because the sound palette tends to be small and the range of controls limited, the outcome of which is too-often difficult to listen to over extended periods of time. Some examples of this latter approach include extensions for the *Matlab* numerical environment (Miele 2003), *AVS Visualization Toolkit* (Kaplan, B. 1993), *Excel* spreadsheets (Stockman, Hind and Frauenberger 2005), and the *R* statistical analysis package (Heymann and Hansen 2002). They provide data handling and processing capabilities but very basic sample-based sound capabilities modelled on the MIDI protocol. What are needed are tools that afford a balanced, equally-flexible approach.

Excellent data-processing tools exist in the public domain (see §5.3.2.2 below) and are an integral part of much scientific research. Furthermore, they are continually being extended and modernised by teams of developers spread across the world. Yet because of the decision to use a music composition environment for sound rendering, these tools remain inaccessible to most sonification environments. Whether this situation has come about because the data for music composition by computer is mostly internally generated rather than externally acquired is open to debate. However, until sophisticated tools for handling externally-acquired often massively multiplexed datasets can be brought to bear on the acquisition, analysis, storage and re-presentation requirements of the sonification process, even before any mapping or sound rendering is undertaken, there is limited chance that such software will enable the *choosing* [of] *mappings between variables and sounds interactively, navigating through the data set, synchronizing the sonic output with other display media, and performing standard psychophysical experiments* that the *Sonification Report* envisaged.

5.1.2 Motivations

Many of the issues raised in the previous section are generic to software in general. Anecdotally, 'late delivery' and 'over-budget' are as common as they are notorious characteristics of the commercial software industry as all but the largest applications find it difficult to maintain hardware and operating-system currency; two of the essential activities necessary to meeting their obligations to paying customers, if they hope to survive. Some do, of course, but increasingly developers and users are turning to open-source, community support, to sustain ongoing viability in the medium-to-longer-term.

The need for a serious solution to the issues outlined above was first undertaken in the context of this thesis as a result of the difficulties encountered in trying to sonify a single large multidimensional dataset for the ICAD-04 *Listening to the Mind Listening* project. The sonifiers involved used different hardware platforms, under different versions of operating systems, and with each having a preference for, and expertise in a different collection of sound synthesis/music composition programs (Barrass 2004; Barrass, Whitelaw and Potard. 2005; Dean, Worrall and White 2004). While all involved technically literate, it was soon realised that if the difficulties experienced were any indication, it must be very difficult for almost everyone to achieve consistent, repeatable results under the same conditions with anything but the simplest datasets. This led us to specifying the requirements for an experimental software sonification framework. Some requirements are clearly identified in the *Sonification Report*, others of our own concoction. We call the framework *SoniPy*, in-keeping with the naming convention used for frameworks that extend the *Python* programming language.

5.2 *SoniPy*: concepts and requirements

SoniPy is designed to be a heterogeneous software framework for data sonification research and auditory display. It integrates various existing independent components such as those for data acquisition, storage and analysis, cognitive and perceptual mappings as well as sound synthesis and control, as illustrated in Figure 5.1, by encapsulating them, or control of them, as *Python* modules. The choice of *Python* was not arbitrary; it possesses all the features of a modern modular programming language that we consider essential for an experimental development environment. *Python* is

> a general-purpose programming language … which may also serve as a glue language connecting many separate software components in a simple and flexible manner, or as a steering language where high-level Python control modules guide low-level operations implemented by subroutine libraries effected in other languages. (Watter, van Rossen and Ahlstrom 1996).

Other descriptors include: simple, but not at the expense of expressive power, extensible, embeddable, interpreted, object-oriented, dynamically typed, upwardly compatible, portable and widely and freely available[8].

5.2.1 Design Requirements

As the *Sonification Report*'s *Sample Research Proposal* (op.cit. #3) acknowledges, the development of a comprehensive 'sonification shell' is not easy. The depth and breadth of knowledge, and skills required to effect such a project are easily underestimated. Whilst it has been a decade since the *Report* was published, progress has been quite slow. This is not to criticise those that have fallen by the wayside, nor those still in development, but to acknowledge both the difficulties involved in such a project and the need for new requirements if such projects are to have a better chance of survival. We briefly address the requirements indicated in the *Sonification Report* and add some of our own.

- *Integrability*. As discussed earlier with regard to data, due consideration needs to be taken of the requirements of the various components of the sonification and experimentation process. As is the case with most interdisciplinary ventures, each contributing discipline brings its collection of tools, techniques and standards to the venture and they need to be synergistically integrated. A software environment needs to be chosen that supports this goal. It is for this reason we have chosen *Python*, which can be used to 'wrap' independent pieces of conformable software in such a way as to permit data to flow between them. We follow *Python* convention and call them Modules.
- *Flexibility*. Rather than try to be the 'killer application,' *SoniPy* aims to wrap (inherit, or be extended by) the best collection of Modules available to it. These Modules need to have no computational interdependencies, though conceptually they may be similar, thus ensuring that no one of them is indispensable. Each of these Modules has evolved independently, probably over a considerable period of time. Independent of *SoniPy*, they have their own ongoing development teams that extend and improve then

8 http://www.python.org/

as well as adapting them to ever-changing hardware and software platforms.

- *Extensibility*. In the situation where no Module exists for a particular task, a new Module can be designed in the knowledge that it will fit seamlessly within the existing Modular framework. This implies all Modules need to be thread compliant.
- *Accessibility.* It is desirable that as many Modules as possible be known in their own right. This reduces learning overhead for all users and enables work that may have already been undertaken with those tools to be accommodated within the *SoniPy* framework.
- *Portability. SoniPy* needs to be able to be instantiated on all major platforms. Furthermore, it is desirable, in certain applications, for Modules to be instantiated on different machines, in different locations and networked together: that is, to be heterogeneous.
- *Availability.* To protect both authors and users, *SoniPy* needs to be freely available with a minimum of restrictions. There are numerous licensing flavours for public-domain software whose source-code is made generally available, as outlined by the GNU organisation[9]. Because *SoniPy* employs heterogeneous components, the license of each component carries through into *SoniPy* in a way that is standard practice in the software industry. The *SoniPy*-specific components will be issued under the General Public License Version 2,[10] thus encouraging the sharing and free exchange of these tools in the community at the same time as enabling restrictions to be applied for individual projects as confidentiality agreements demand. *SoniPy*'s sources and documentation can be freely downloaded from its Sourceforge Internet repository[11].
- *Durability. SoniPy* needs to *survive*. While survival can never be guaranteed, in complex projects such as this, maximum risk-mitigation is essential. *SoniPy* is unlikely to survive if it remains the effort of a very small group. Essential to this is the community involvement and support in ongoing improvement and development: the very conditions under which *SoniPy*'s independent modules have been, and continue to be,

---

9 http://www.gnu.org/philosophy/categories.html
10 http://www.gnu.org/copyleft/gpl.html
11 http://sonipy.sourceforge.net/

developed. This approach is not without its own issues, as those who are involved in public-domain projects can attest (Luke et al. 2004).

5.2.2 Integration Through Wrapping

Although *Python* comes with an extensive standard library and there is a good resource of external *Python* libraries, we are not limited to using *Python* libraries. A powerful feature of *Python* is its set of well-defined interfaces to other languages. Libraries written in most languages can be integrated through *Python* by 'extending' it [24]. The basic principle of *SoniPy* is to use *Python* to 'wrap' independent software that can be compiled with python-bindings in such a way that data can flow between them. Quite a few tools exist for the (semi-) automatic generation of Python bindings, such as the *Simplified Wrapper and Interface Generator* (SWIG)[12].

Some applications provide *Python Application Program Interface* (API) libraries; other applications need to have a *Python* API written in order to use it (Lutz 1996: 505). Although some others embed *Python*, either by bundling it as an interpreter or by invoking the Python interpreter installed on the user's system as a basic API (ibid.: 505). We mention embedding in this context because, whilst it may be useful in its own right, it does not provide the interface flexibility needed by *SoniPy*. *SoniPy* requires an application to provide *Python* bindings so that *Python* can be *extended* by the application.

5.3 The design of SoniPy

The *SoniPy* design specifies five module sets communicating over two different networks: the *SonipyDataNetwork* (SDN) and the *SonipyControlNetwork* (SCN). Modules are grouped according to their role in the data sonification process: Data Processing (DP), Conceptual Modelling (CM), Psychoacoustic Modelling (PM), Sound Rendering (SR) and Monitoring & Feedback (MF). Depending on the dictates of a particular project, modules in a set may be instantiated on different machines. A particular module set may be empty, i.e. contain no modules, or a particular module may belong to more than one module set.

---

12 http://www.swig.org/

5.3.1 Inter-Module Communication: The Networks

SoniPy's modular design makes it well suited for the instantiation of all selected modules on a single processor or, in order to take advantage of the computing power that multiple CPUs and machines can afford, the distribution of modules over multiple CPUs, a LAN or the Internet. We are currently extending *Python* versions greater than 2.4 under Apple's latest *OSX* operating system, but workable alternatives will flow as the development team expands.

Python's platform independence should assist the instantiation of parallel implementation of *SoniPy*, distributed over a heterogeneous network. Some testing of different approaches to distributing the computing has been undertaken, in order to maximally benefit from the trade-off between performance (including real-time latency, data throughput and CPU overhead issues), ease-of-use, maintainability, reliability (over a network), scalability and heterogeneity; that is, the ability for non-*Python* third-party applications or devices to communicate with *SoniPy* modules (Coulouris, Dollimore and Kindberg 2005). Communication technologies being tested range from class inheritance, sockets, *OSC* and *MIDI* through to network audio mixing, using *Netjack* for example[13].

Other potential uses of a distributed approach include mobile phone sound-rendering and the processing of data remotely under local control, perhaps with the result being sent to another site for mapping, and psychoacoustic adjustment before being rendered to sound. The testing in ongoing and the interim results are not reliable enough to warrant reporting at the time of writing.

Referring to the diagrammatic representation of the way the module sets interrelate (Figure 5.1), it can be observed that *SoniPy*'s modules operate through two networks: *Data* and *Control*. The *SonipyDataNetwork* (SDN) is topologically configured as a bus whilst the *SonipyControlNetwork* (SCN) is a star configuration. This is analogous to the signal and control busses of an automated audio mixing desk. Control routing uses the same network technology as the data, though the destinations may be different. For example data from a DP module may be sent to a CM module on the SDN bus, under

---

[13] http://netjack.sourceforge.net/

the control of an MF module communicating on the *SonipyControlNetwork*, without the data itself needing to go through an MF module. *Sonipy* controls need to be XML compliant and each module set may itself be the hub of a network of processors of topology unknown to the SCN router.

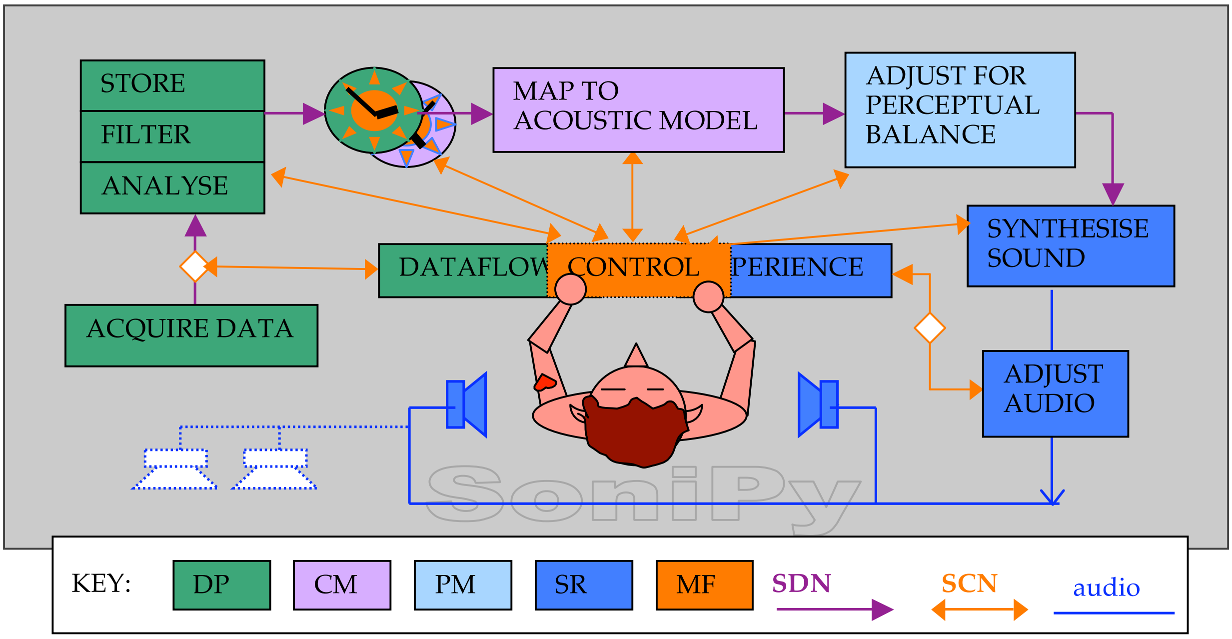


Figure 5.1. Flow diagram of *SoniPy*'s five module sets and two networks

Module Sets	Data via SoniPy Data Network (SDN)	Controls via SoniPy Control Network (SCN)
DP	Input: Raw data to be sonified Processes: Analysis, filtration, translation, storage Output: Modified Data to be sonified, Metadata	Process selection & OP switching options State (active, waiting, idle)
CM	Input: Data from DP, model selection algorithms Processes: Model Selection Output: Selected Model(s)	Process selection & output switching options State (active, waiting, idle)
PM	Input: Model control parameters Processes: Psychoacoustic transforms Output: Modified model control parameters	Process selection, & OP switch options Model instantiation map for render(s)) State (active, waiting, idle)
SR	Input: Synthesis models and controls Processes: Render sound Output: Sound	Audio controls: mute, unmute, gain etc Audio control changes State (active, waiting, idle)
MF	Input: Login, config. & process startup Processes: Initiate, route, record activity, monitor usage, system/networks state Output: UI, feedback, log of activity	State (active, waiting, idle, off) Resource usage, UI monitoring.

Table 5.1. Overview of some key features of *SoniPy* module sets . KEY: DP: Data Processing. CM: Conceptual Modelling, PM: Perceptual Modelling, SR: Sound Rendering, MF: Monitoring and Feedback, SDN: SoniPy Data Network, SCN: SoniPy Control Network.

5.3.2 The modules

Whenever possible, local modules are instantiated as library extensions to the *Python* programming framework. Should a module be instantiated remotely, that instantiation and the control of information to and from it remains under the control of the MF module that initiated the instantiation, where the main application loop as well as thread and GUI controls resides. Table 5.1provides an overview of some key features of the module sets.

5.3.2.1 Data processing modules

As the principal data processing activity in data sonification is taking place inside the listeners, the role of sonification is to prepare source data in a format that enables the listener to extract information, and in interactive systems, to take account of their feedback. *SoniPy*'s DP modules consist of a number of object-oriented classes which themselves inherit data classes and control classes according to the form and location of the raw data and its intended destination. Class methods include those for

- Interpolated lookup and mappings, for auditory icons and earcons, for example,
- Writing data to and extracting it from storage (memory, database and/or flat file) for pre-processing or multi-stream playback,
- Audification - writing data in formats acceptable as direct input to audio hardware,
- Simultaneous handling of multiple time-locked streams, such as from biomedical monitors,
- Deconstruction, analysis and filtering, including of complex meta-tag embedded multiplexed streams, such as a data feed from a stock-market trading engine,
- Model-based sonification involving user feedback, and
- Simulation of data feeds, including buffering with time compression and expansion.

Data and control class instances can be manipulated with *SciPy* (Scientific Python), a collection of open-source libraries for mathematics, science, and engineering[14]. The core library is *NumPy*, which provides convenient and fast N-dimensional array manipulation. Figure 5.2 is a diagrammatic representation of one way of configuring *SoniPy*'s data processing modules under *SCN* control.

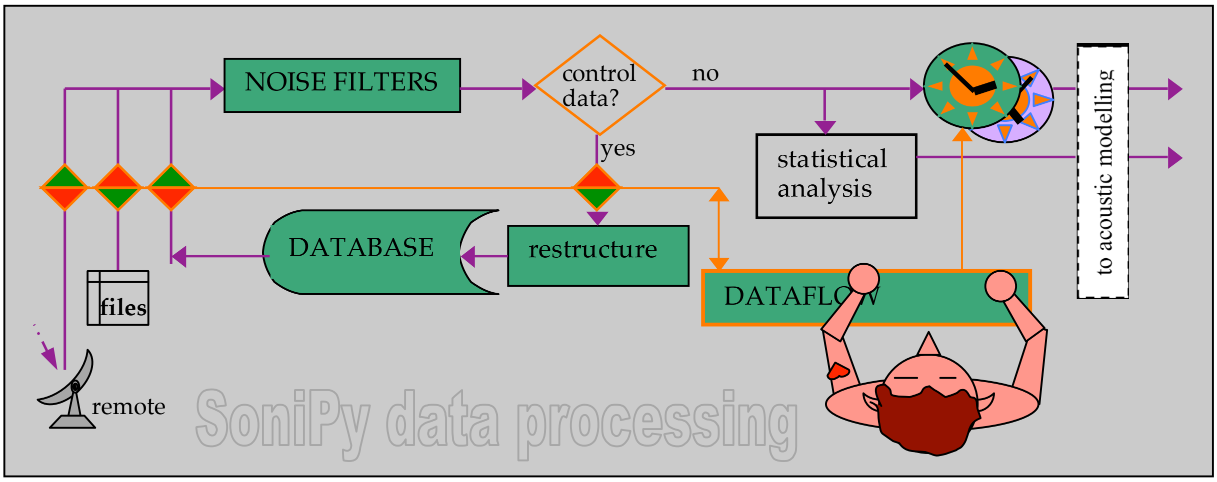


Figure 5.2. A map of the configuration of *SoniPy*'s Data Processing modules.

5.3.2.2 Scale, storage, access, and persistence

When the dataset used for sonification is finite, not too large, and is acquired in a timely manner, it can all be held in RAM and parsed using the basic data-manipulation tools of any modern programming language. Alternatively, as can be observed in the Auditory Display literature, it can be pre–processed separately using generic spreadsheet software. As the size, complexity and performance demands increase, as they frequently do multidimensional in real–time datasets, or when what is being sonified is some computed informational derivative of the raw data, the power and flexibility of an interpreter interface to a software framework reveal its superiority over purely music-oriented software[15].

In *Python*, direct access to many data management tools is available: Internal lists can be extended to typeless {key : value} dictionaries, then to flat files, perhaps with compression[16], and *NumPy* array processing, then using framework extension, to commercial relational databases like *Oracle* and the web-prevalent public domain *mySQL*. With very large datasets, such as those

---

14 http://www.scipy.org/

15 Including *SuperCollider*'s *scLang* probably the closest to Python in concept, in that it is text-based, interpretive and contains a dictionary class.

16 Python has its own simple flat-file data compression module called *pickle*.

regularly employed in astronomy, meteorology and quantitative finance, one has to decide whether to use one of a number of models, such as *hierarchical*, *network*, *relational*, *entity-relationship*, *object-relational model* and *object*[17], depending on the structure of the data and the way it is used.

The relational model is characterized by relation, attribute, tuple, relation value and relation variable. A relation is, at its heart, a truth predicate about the world, a statement of facts (attributes) that provide meaning to the predicate (Date 2004). Speaking simply, relational systems are structured around relationships between facts and data retrieval is based on predicate logic and truth statements.[18] Objects systems are structured so as to emphasise identity, state, behaviour and encapsulation. An Object has a unique identity that is distinct from its state (the value of its internal fields)–two objects with the same state are still separate and distinct, despite being bit-for-bit mirrors of one another (Date 2004).

Whether to use an equivalence or identity model to structure data persistence[19] is a well–known programming problem: the object–relational–mapping– or ORM–dilemma.[20] The solution depends on both the structure and distribution of the data and the types of enquiry needing to be made of it, and can be the subject of extensive experimentation. Relational databases tend to perform more effectively when the data can be easily structured in a few relatively large tables in with a fixed number of fields and the object model when there are a large number of semi-autonomous instances. These issues are exemplified, with sample code, in the context of data sonification experiments using high-frequency capital market data in the next chapter (§5.8) where an HDF5–compliant b–tree structure was eventually employed using *pyTables*[21].

---

17 Wikipedia has useful overviews. See http://en.wikipedia.org/wiki/Database_model and http://en.wikipedia.org/wiki/Object-relational_mapping.

18 In essence, each row within a table is a location for a fact in the world, and a structured query language (SQL) allows for operator-efficient data retrieval of those facts using predicate logic to create inferences from them.

19 In computer science, the term *persistence* refers to that characteristic of data that outlives the execution of the program that created it. Without persistence, data only exists in RAM, and will usually be 'lost' when the program is terminated.

20 Wikipedia has a useful introduction. See http://en.wikipedia.org/wiki/Object-relational_mapping

21 Hierarchical Data Format (HDF) technologies are relevant when the data challenges being faced push the limits of what can be addressed by traditional database systems, XML documents, or in-house data formats. See http://www.hdfgroup.org/why_hdf/. *PyTables* is a *Python* API for HDF5. See sonification.com.au/sonipy/dataPersistence.html on the *SoniPy* website for more information.

5.3.2.3 Conceptual modelling and data mapping

In the *SoniPy* model, information mapping is divided into separate cognitive and psychoacoustic stages. The cognitive stage involves the design of 'sound schemas' with semiotics, metaphors and metonyms relating to the task, and aesthetic and compositional aspects relating to genre, culture and palette. Decisions have to be made about functionality, aesthetics, context, learnability, expressiveness, and device characteristics. These decisions are typically drawn from existing knowledge and theories from relevant sciences, arts and design. The consequences of these compounding decisions are difficult to predict empirically: one of the reasons why sonification is currently more of an heuristic art than a science. Nevertheless, as different conceptual models are developed, some based in cognition, others culturally determined, they can be integrated into *SoniPy* using the wrapping techniques outlined.

One example might be the construction of an interface to *TaDa*'s methods; a design approach to sonification that provides a systematic user-centred process to address the multitude of decisions required to design a sonification (Barrass 1996a). *TaDa* starts from a description of a use case scenario, and an analysis of the user's task, and the characteristics of the data. This analysis informs the specification of the information requirements of the sonification. *SoniPy*'s support for the *TaDa* method would be through a python-based GUI that captures a user scenario and provides standard *TaDa* fields for analysis. This GUI, connected using the SDN to a *mySQL* database that contains an increasing number (currently about 200) of stories about everyday listening experiences, analysed using the *TaDa* data-type fields. This database, called *Earbenders*, is a case-based tool for looking up 'sound schemas' at the cognitive design stage (Barrass 1996b). In future, a *Python* interface to *TaDa*'s SonificationDesignPatterns wiki could also be developed as an alternative pattern-language approach for cognitive level design (Barrass 2003).

5.3.2.4 Psychoacoustic modelling

The psychoacoustic modelling stage involves the systematic mapping of information relations in the data to perceptual relations in the sound schema (Barrass 1996c). The concept is that *SoniPy* would provide support for this by allowing interactive reconfiguration of the mapping from information relations to auditory relations. Changes in this mapping cause the automatic remapping of source information through psychoacoustic algorithms (implemented in the fast array processing tools, *NumPy* and *SciPy*) to produce new sounds and/or rendering controls. For example a change from categorical to ordered information could automatically produce a remapping from a categorical sound (e.g. instrument, object, stream) to an ordered property of a sounding object (e.g. length, excitation, distance).

5.3.2.5 Acoustic modelling

SoniPy provides a user access to many more options than if a music composition or sound synthesis environment was chosen before beginning the development of other aspects of the data sonification framework. For low-level audio work, a *Portaudio* module can be used for audification and as the basis for the development of other such modules should the need arise (Burk and Bencina 2001). *Portaudio* is used via the python wrapper *PyAudio* for the audio streaming in the capital market net returns experiments in the next chapter. *SndObj* ('sound object') is a middle-level toolkit also immediately available in the same manner (Lazzarini 2000). *SndObj*, like many real-time audio applications today, uses the *Portaudio* interface to the audio hardware. *The STK toolkit* also appears to be wrappable (Scavone and Cook 2005), as does the higher-level *RTCmix C++library*,[22] although no attempt has been made by this author to do so. *Csound* has an embedded *Python* API for writing extended Opcodes for a couple of years, and *Csound5*, the latest evolution of this classic computer sound synthesis language, has a python extension wrapper interface. At the time of writing it is undergoing extensive multiple-platform testing.

Whilst some high-level applications such as *Max/MSP* and *PD*, are unlikely to become extension toolkits to *SoniPy*, it is still possible to use them by instantiating them independently and communicating with them using computer music protocol specifications *OSC* (Wright, Freed, and Momeni 2003) and *MIDI*[23]. *SuperCollider3* is a special case because of its inherent modularity: the sound-rendering component, *Scsynth*, can be instantiated as a separate program to the language, *SClang*. Communication tests between this 'external' *SCsynth* over *OSC* using the SoniPy framework as an alternative to

22 http://www.rtcmix.org/
23 http://www.midi.org/

Sclang indicate that this is viable. If very low-latency is a requirement, such as may be the case for interactive sonifications, *pySCLang*, originally designed as an alternative to *Sclang* for Windows platforms, enables direct communication with an instantiation of the SC language and its internally embedded sound renderer. Although this seems like a circuitous route, it works and is another example of the robustness of the encapsulating framework approach used in SoniPy.

Non-operating system dependent text–to–speech synthesis will be available through a *PySpeak*, an OSX thread-compliant *Python* API to of *Espeak*[24] [25].

5.3.2.6 Monitoring, feedback and evaluation

Monitoring and Feedback of SDN and SCN can happen via the *Python* interpreter. Having access to an interpreter in order to build a complete sonification by iteratively building on small tests is a powerful aspect of *Python*. Heterogeneous connectivity also allows the consequences of decisions at each stage to be tested on a compound design, thus enabling better understanding and control of non-linear and emergent effects in an overall design.

For cross-platform GUI-building, *wxPython* provides access to *wxWidgets*[26] and *wxGlade*[27] can assist in more-rapid development of GUIs by automatically generating *Python* control code and separating the GUI design and event-handling code. If a relatively consistent interface across all hardware platforms is more desirable, *Tcl* GUI-building tools[28] are available through native *Python* modules.

The inclusion of evaluation modules will make it possible to design different types of empirical experiments, and conduct and analyse the results within a single framework environment, even using website-based surveys if desired. This would be a marked improvement on current public-domain experimental psychology software, as there is currently no such tool that permits separately-threaded audio streaming or soundfile playback. In

---

[24] http://espeak.sourceforge.net/

[25] The most lauded text–to–speech is probably *Festival* (http://www.cstr.ed.ac.uk/projects/festival/). Initial compatibility caused it to fail the ossification test.

[26] http://www.wxwidgets.org/

[27] http://wxglade.sourceforge.net/

[28] http://www.tcl.tk/

addition, a user-contributed library of experiments for evaluating sonification designs could assist in developing standards for measuring the comparative functionality, aesthetics, learnability, effectiveness, accuracy, expressiveness and other aspects of individual design. While the design and implementation of a complete empirical sonification evaluation environment is beyond the means of many researchers in experimental psychology, the contribution to the design and implementation of such community-supported tools may not be. As already evidenced by such projects as *SciPy*, this approach is currently in other scientific disciplines such as mathematics, physics, chemistry, astronomy, meteorology and genetics, where the design and sharing of experimental tools has long been an integral component of their research endeavours.

5.4 An illustrated metacode example

Example Code 1 is a simple 'high-level' illustration of part of the *SoniPy* framework in action. The first task is to accept a multiplexed meta-tagged data stream from the Australian Stock Exchange (ASX) trading engine. The ASX is medium-sized exchange, on which about 3,500 securities are traded. It generates about 100 MB of trading text data daily, making it impractical to hold enough data in RAM to do all the calculations necessary.

The data is processed into a *MySQL* database using an Object-Relation Mapping paradigm supported by the *sqlobject* module[29]. This abstracts the handling of the dataset, providing an interface between the tables and indices database paradigm and Python's object-orientation. Other modules (such as *mySQLdb*) are available if direct interaction with the database server in *MySQL* code is more appropriate. A list of securities that meet, or are likely to meet, the criteria necessary for a sonification event to initiated, is held in RAM and processed as a multidimensional array using the *NumPy* module. When the criteria are met this data is also used as some of the input parameters to the sound renderer. The *pyspeak* module is invoked to synthesise the name of the security being newly rendered. In this example, the sound is rendered by the *SuperCollider3*'s external synthesis engine, *scsynth,* with which the python *scsynth* module bi-directionally communicates using the *OSC* protocol. This

---

[29] http://www.sqlobject.org/

permits the use of *SC3*'s *synthdefs* (synthesis definition algorithms) that are capable of responding to the criteria, as established, or as modified in real-time. Other synthesis options, such as the lower-level *pysndobj* or *pyaudio* (the python interface to *portaudio*) are possible, as is the *libsndfile* library for audiofile playback. The code example illustrates how *SoniPy* combines *Python* code, imported 3rd party modules and user-defined scripts. It is meant here to provide a sense of the immediacy and readability of the approach. More detailed code examples can be found in the Appendix 6.

5.5 The *SoniPy* website

A continually evolving website has been established to assist in establishing *SoniPy* as an active public domain project[30]. In order to do this, the site has been structured around five principle activities:

1. A first port-of-call for sonification-related activities using *Python*.
2. A single help location from which module documentation can be accessed. This is a frames portal that functions on similar principles to that of *Sonipy* modules, in general: by wrapping the existing on-line module documentation if it is available. Otherwise, the documentation that accompanies the module's sourcefiles is built in HTML and/or PDF format and served from the SoniPy website instead.
3. Tutorials on the use of the framework to undertake sonification tasks. The experiments outlined in the next chapter will eventually be placed there, for example, and links to independent third-party material such as *figusdevpack* (Fabbri and Chiozo 2008).
4. A storage and download system. This will include a high-level mechanism for automating the meta-bundling, download and installation of modules, component sets and their dependencies, for various hardware and operating systems, using *Python* Egg technology and the *Python* package manager *Easy-Install*[31]. In addition a software license reference for modules aids the collation of modules appropriate in different commercial circumstances.

---

30 http://www.sonification.com.au/sonipy/

31 "Eggs are to *Python* what jars ar to Java." *Python* Eggs are ZIP archives with the file extension .egg. See http://peak.telecommunity.com/DevCenter/PythonEggs.

5. A community-managed forum for both users and developers to facilitate the detailed orderly discussion of aspects SoniPy. At the time of writing, the forum contains a record of the testing and evaluation of many of the modules discussed in this chapter.

```
# Metacode Example; Sonipy Main Loop

# ::::::::::: import the modules needed ::::::::::::::::::::
# for Australian Stock Exchange (ASX) Trading Engine datafeed.

import DataFeeder
import StreamMonitor
            # uses matplotlib for graphical presentation
            # of trading engine stats
import DataToDB
            # includes an import of the sqlobject Module 3,
            # which provides OO Class structure of the
            # Relation-to-Object-Mapping necessary for handling
DBs.
            # For simple DB calls, use mySQLd
      # to simply pass mySQL commands to the server.
import SecurityMaps
      # mapping OO classes for ASX securities.
      # this Module calls the numpy module for
      # fast multi-dimensional array processing.
import PsychoFilters
      # a set of filters for transforming mapped data
      # Used to adjust Security Maps for psychophysical
      # non-linearities, condition enhancement etc
import SoundRenders
      # A set of synthesis engine interfaces and
      # synthesis definitions (instruments) appropriate
      # for this sonification. Includes OSC and MIDI,
      # and the phoneme synthesiser pyspeak.

# ::::::::::: ASX sonification threads ::::::::::::::::::::
ASXFeed=DataFeeder(feedURL='http://localhost')
      # Establish connection to datafeed. The dtafeed
      # URL could be an internet,LAN address or filepath.
      # Uses Python build-in modules for low-level
      # data transfer to RAM buffer.
StreamMonitor.monitorStream (ASXFeed)
      # Monitor data stream.
DataToDB.MultiplexStreamtoDB(ASXFeed, ConnectParams)
            # Start processing ASXFeed into DB. Uses a MySQL DB
            # DB client connection as specified in ConnectParams.
            # Other servers are possible, including remote ones.
ASXalerts=SecurityMaps.Indicator(stocks, UpBollinger(20,9,2))
      # a FIFO of alerts for securities currently trading
            # outside the specified upper Bollinger band. The FIFO
            # is constantly updated whilst DataFeeder is active.
SoundRenderers.BinauralOut='TRUE'
            # Process sound output will for binaural listening.
while (StreamMonitor.StillActive):
    for security in ASXalerts:
        if security not in renderList:
            pyspeak(security)  # announce new security
            SoundRenders.scsynth(security, stock.scsynthdef)
```

Code Example 1. The Sonipy Framework in action. '#' is the Python comment character. The task modelled is to accept data streamed from a stock market-trading engine and use sonification to alert the listener to specific features of the trading activity as given .

5.6 Summary

Sonification research is an interdisciplinary activity and to date, tools for undertaking it have either been discipline-specific, modified to accommodate the interconnections, ad-hoc collections, or stand-alone programs developed for a specific sonification task. Because *SoniPy*'s open architecture design can integrate modules conforming to widely accepted inter-process computation standards (wrappable libraries), in a non-conflicting way, it will be possible for it to grow in the directions its user-community needs it to. Instead of sonification researchers trying to piece together a collection of tools they will hopefully be able to make work together, they will be able to choose from a number of possible *SoniPy* framework modules, based on a best-fit-for-the–task evaluation, and rely on the continuity of the framework to provide inter-module integration.

Because the framework is implemented in a common, 'user-friendly,' scripting language, programming assistance, when needed, will be more readily available than if a specialist application were used, and this may, in term, assist individual endeavour and promote better independent evaluation of the empirical results of other research in sonification. There is currently, for example, a dearth of good public-domain software for experimental psychology that uses sound, and the integration of such a module, or set of modules, into *SoniPy* would be a welcome addition to the field.

It has not been practicable, in the current context, to test all the alternative combinations of modules accessible and useful in undertaking data sonification. The approach taken was to select a number of modules from different domains and test, albeit in a reasonably ad-hoc way, the viability of their combined use to solve a specific problem that would normally be undertaken, often with some awkwardness, using two or more unrelated tools. Once the library encapsulation had been effected, we did not experience a single conflict during this process.

Probably the most important feature of the framework approach outlined in this chapter is that, by using tools largely built by and for large communities of uses with a vested interest in their survival independent of data sonification, sonifiers are less likely to be 'held to ransom' by a reliance on the need for the developers of single software applications to continue to respond to the ongoing hardware and operating systems development environments in which they operates, and thus risk the isolation and eventual obsolescence that inevitably follows when they do not. *Python*, *NumPy*, *Matplotlib*, *Portaudio* etc will continue to evolve and eventually be replaced by improved alternatives, but in ways that are more adaptive than catastrophic.

While the project is in its infancy, a major concern of the overall design has been the efficiency and effectiveness with which it can be implemented and maintained by a small team of coordinating developers. Whether or not this occurs is dependent on to the vagaries of public forces but there is some evidence that, since the original papers were published (Worrall et al. 2007; Worrall 2008), others have begun to build on this initial work for the *Linux* operating system (Fabbri and Chiozo 2008). It may, however, find its place in the public sphere as a part of a larger project, such as *SciPy*, or the more tightly defined *python(x,y)*[32].

There are some clear, and some not–so–clear distinctions between software *design* and software *engineering* as designer and computer musician Bill Buxton emphasises (2003). Interpretive languages tend to afford individual users the opportunity to do both, but this relies on strong design principles being well implemented, both in the underlying language (*Python* in this instance) as well as extensive framework extensions such as *SoniPy*. By establishing an open source project based on such design principles, it is hoped that the initial work presented here will be taken up by others who, in turn, will contribute an evolving framework that is useful to the wider sonification community; that data sonification software design 'escapes' from the engineers to become a more widely accepted part of what it means to 'do' sonification than is currently the case.

---

32 http://www.pythonxy.com/

THIS PAGE HAS INTENTIONALLY BEEN LEFT BLANK

Chapter 6

SONIFICATIONS WITH CAPITAL MARKET TRADING DATA

> The flexibility of money, as with so many of its qualities, is most clearly and emphatically expressed in the stock exchange, in which the money economy is crystallized as an independent structure just as political organisation is crystallized in the state. The fluctuations on exchange prices frequently indicate subjective-psychological motivations, which, in their crudeness and independent movements, are totally out of proportion in relation to objective factors. (Simmel 1900/1979: 326)

6.1 Introduction

There are many reasons, both sociological and technical, why capital markets[1] are an interesting application–domain for sonification. Sociologically, they have become a powerful, some might say almost religious contemporary force, even as their overtly emotional expressive open-outcry marketplaces have become, or are quickly becoming, virtualised gatherings of disembodied screen traders. Sonification of the activities of these markets thus functions as a form of re-embodiment. While such sociological considerations form the background and motivations for the current work, the focus here is more technical.

Despite intensive study, a comprehensive understanding of the structure of capital markets exchange trading data remains elusive. As will be discussed, there is only one known application of audification to market price data, and it reported "results proved difficult to interpret, probably because the stock market does not follow physical-acoustic resonance laws" (Frysinger 1990). This chapter reports a new investigation into the application of audification and other sonification techniques to capital markets exchange trading data. It is envisaged that the approach outlined and informal results obtained will be followed by detailed empirical studies that may show a wider application of the techniques, both with capital markets exchange

1 The nomenclatures used for various kinds of markets is somewhat convoluted and are often misused. A *stock* market is for trading equities, namely ownership interests in companies. A *financial* market is an institutional structure or mechanism for creating or exchanging financial assets, namely those that are real such as land, buildings, equipment, patents. The term *capital market* is used generally to refer to markets for stocks, bonds, derivatives and other investments and it is this term we adopt here, or the less formal *The Market*. Both are generic terms signifying all regulated exchanges and their activities.

trading data and other similarly structured time-series data; from electoencephalography and hierarchical networks, such as those arising from social affiliation (Sornette 2003: 171-227).

6.2 The sonification task

As will be detailed, the statistical analysis of trading data, and stochastic modelling using it, is an ongoing area of research in finance. Two principal concerns are ways to accurately describe the way prices are distributed statistically and whether or not, and to what extent, auto-correlation exists and can be detected, even pre-empted market prices evolve. Understanding the accuracy of the description of price distribution is important for the risk analysis of various trading instruments in the longer term, and understanding the inherent autocorrelation is important in attempts to predict future, especially catastrophic, events.

The power of visual representation to enhance and deepen the understanding of phenomena and their abstractions is undisputed. Yet, as with many time-domain processes, visual representation does not reveal the structure of the data. For example, of the two graphs shown in Figure 6.1, one is of a real market data, the other generated using Brownian motion: an independent (uncorrelated) random walk in which the size and direction of the next (price) move is independent of the previous move(s).[2]. Mandelbrot poses the following question: "Is it possible to *visually* distinguish which is which?"Arguably the most experienced researcher in the field, he argues it is not (Mandelbrot and Hudson 2004: 16-18).

This question leads to a sonification question, which the research reported in this chapter seeks to answer: Are there ways of presenting trading data that enable its structural characteristics to be perceived aurally? If so, can real trading data be distinguished from a stochastic simulation of it?

The importance of understanding the data itself before attempting to sonify it has been emphasised by many authors. Sarah Bly (1994) enunciated the imperative:

---

2 A statistical analysis of time series data is concerned with the distribution of values without taking into account their sequence in time. Decorrelation changes *the sequence* of values in a time series and so completely destroys any spectral information, while preserving its statistical properties.

> Whatever the technique, a first concern is to understand as much as possible about the data itself...With multivariate data, it is important to know not only the relationships between various variables but also which variables are predominant in carrying the information about that data (Bly 1994).

And more than a decade later, John Flowers again emphasised the need to proceed judiciously rather than hoping to 'get rich quick':

> There may be some auditory analogies to visual plot clustering (auditory scatterplots offer a primitive example) but it will take a great deal of display engineering, and judicious variable selection to make this principle apply to more complex multivariate data. Insightful data groupings are not likely to pop out by simply submitting a multivariate set to a sonification engine and "plotting everything." (Flowers 2005).

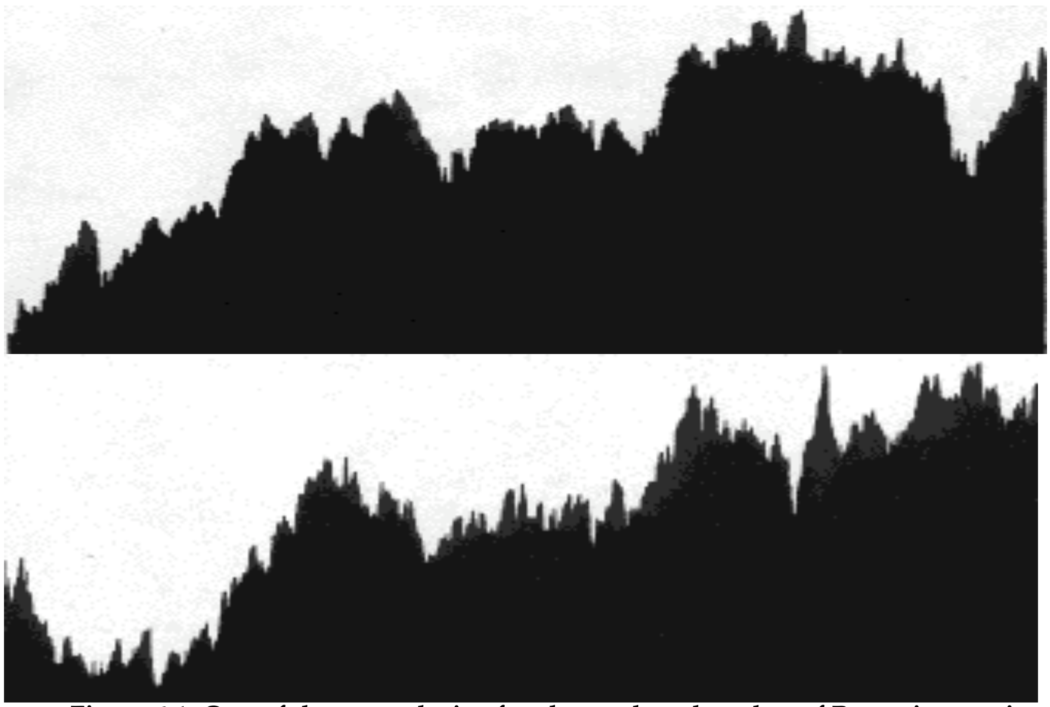


Figure 6.1. One of these graphs is of real a market, the other of Brownian motion. Are there distinguishing characteristics? (after Mandelbrot 2004: 16-18)

The rest of this chapter is organised as follows. An overview of the capital markets, the types of data generated by them and a description of how that data is used by various groups in the marketplace is followed by a general description of attempts to quantitatively simulate the structure of that data. Previous related sonification studies are reviewed, including those of stock-market data, of time series data, audification of statistical functions. Finally the chapter reports of a some experiments to investigate whether audification and some other sonification techniques can be used to distinguish market data from simple stochastic time series and decorrelated versions of the data itself.

6.3 An overview of capital markets

Capital markets are (increasingly virtual) places where companies and traders converge around an exchange to raise new investment capital and investors and speculators trade exchange-registered securities such as stocks, bonds and futures contracts. These exchanges have strict government-regulated mechanisms for such activity and the community of freely participating individuals around them communicate more-or-less informally with each other and formally through exchange-registered brokers who themselves provide information to their clients about specific trading activity as well as about other more general environmental (financial, political, meteorological etc) conditions that may affect an individual's trading decisions. Such decisions, enacted by the brokers, cause excitations of the trading system, known colloquially as a 'trading engine', which in turn produces data records of its activities. Some of that data, or summaries of it, is fed back for the information of market participants.

In turn, these marketplaces operate as systems within national and global economies; international companies may be listed on more than one exchange. Each exchange's trading system is designed to be acephalously appropriate for the types of securities that are traded on it: shares, bonds, currencies, futures and other derivatives, for example. Exchanges (Bourses in Europe) can be general, such as the Australian Securities Exchange (ASX) and the New York Stock Exchange (NYSE) or confined to specific types of securities such as the London Metals Exchange (LME) and the Chicago Board of Trade (CBOT), which specialise in commodities.

6.3.1 Systemic description of a capital market

A capital market can be modelled as a dynamic feedback system within an environment, as idealised in Figure 6.2.

The dotted loop signifies system feedback. $\vec{E}(\vec{s},t)$ represents the excitation vector of the system from the environment at time $t \varepsilon T$. $\vec{s}$ is an n-dimensional vector in space S.

The response vector $\vec{R}(\vec{s},t)$ represents the output of the system to its environment. This model, after Nigam (1983: 5), relates the excitation to the response through an equation of the form $L(\vec{a},\vec{d},\vec{s},t)[\vec{R}(\vec{s},t)] = \vec{E}(\vec{s},t)$, where $L(\ )$ is a mathematical operator, $\vec{a}$ a vector containing known system parameters, and $\vec{d}$ a vector containing the design variables. The response of the system to excitation can be symbolically expressed as $\vec{R}(\vec{d},\vec{s},t)] = L^{-1}[\vec{E}(\vec{s},t)]$, where L^{-1} represents the inverse operator.

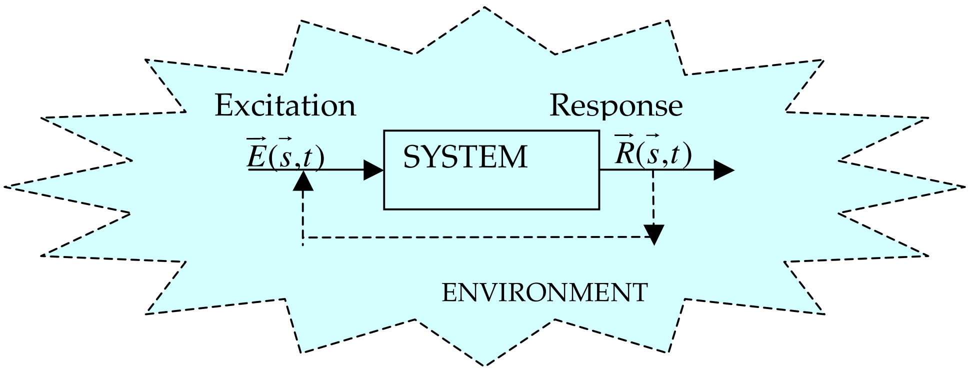


Figure 6.2. An idealised feedback environment.

Applying this generic model to a capital market, ENVIRONMENT represents an economy: its marketplace and all potential participants, SYSTEM represents the exchange's trading engine with design principles $\vec{d}$ and input/output metadata $\vec{a}$. $\vec{E}(\vec{s},t)$ represents inputs to the trading engine by all participants at time *t* and $\vec{R}(\vec{s},t)$ represents the data output from the trading engine at time *t*, which individual market participants make use of, as appropriate for the trading principles by which they operate, as outlined in §6.3.4, below.

6.3.2 Trading data

Trading engines need to be fast, efficient and accurate. They generate large quantities of data, reflecting the shifting moment-to-moment situation of the order book of each of their trading securities as potential buyers and sellers adjust their declared positions, and trades are eventually undertaken. A medium-sized exchange such as the Australian Securities Exchange (ASX) processes approximately two million trades a month: an average of 100,000 trades a day[3]. Security Trading datasets are sets of time-ordered trading events having a number of empirical dimensions, such as price and volume[4], depending on the particular type of security being traded (share, futures, options etc) and from which other data may be derived. Appendix 3 contains an overview of the different the types of data generated by a trading engine, including the data types used in the studies for in this chapter.

6.3.3 Indices: accumulations of trading data

In addition to operational statistics and the trading data itself, various weighted accumulations of it, called indices, are compiled moment–to–moment from the trading data, providing the marketplace with various indicators of the state of the market in real-time. Today, these indices are coordinated to assist global trading.[5]

6.3.4 Methods used by market participants

The actions of market participants, both of individual investors and those who manage other people's funds, is related to the methods by which they make their trading decisions:[6]

1. Fundamental analysis studies the economic forces of supply and demand that affect the performance of companies and economies in general. It seeks to examine all the relevant factors affecting the price of a security in a market so as to determine its intrinsic value both at the time of the analysis and at some generally specified time in the future. Working on the principle that the market will 'discover' true value prices, fundamentalists make buy and sell trade recommendations on the basis of whether the current market price is under value or over value as revealed by their analysis against historically-correlated company indicators such

---

[3] A breakdown of recent ASX trading volumes is available from their website, currently at http://www.asx.com.au/asx/statistics/TradingVolumes.jsp

[4] The term 'volume' is used throughout to mean 'trading volume' not as a psychoacoustic parameter.

[5] GICS (Global Industry Classification Standard) consists of 10 Sectors aggregated from 24 Industry Groups, 67 Industries, and 147 Sub-Industries currently covering over 27,000 companies globally. See http://www2.standardandpoors.com/portal /site/sp/en/au/page.family/indices_ei/2,3,2,0,0,0,0,0,0,0,0,0,0,0,0,0.html

[6] A common distinction between traders and investors is that traders are usually short-term speculators on an instrument's price movement whilst investors have longer-term goals. The distinction is not relevant here as 'taking a position' on an instrument, for whatever reason, involves stimulating the trading engine, so 'trading' is used as the generic term. Murphy (1999) is well-recognised text.

as Price-Earning (PE) ratio and Earnings Before Interest and Tax (EBIT). Fundamentalists tend to use long timeframes as a general guides in their modelling, so their interest in trading data is simply as reports of market prices, and not how the market arrived at that price[7]. Most managers of large Funds base their decisions on fundamental analysis. So, because the effects of their decisions on the market can be significant, quantitative and technical analysts maintain a residual interest in the published predictions of fundamentalists.

2. Chart reading is a collection of heuristic techniques based on feature analysis of charts by which chart-readers (chartists) identify patterns in the visual display of data and speculate on the consequent movement of the market prices based on similar pattern formations in the past[8]. Similar patterns (chiefly of price and volume formation in time) can be observed in charts of all markets across all time-frames. Chart-reading has a long history[9] and has gained increased popularity with the advent of personal computers and the availability of trading data and charting software. More advanced chart-reading techniques include Ralph Elliot's wave principle, first published by Charles Collins in 1938 (Murphy 1999: 319-342) and William Gann's squaring of price and time (Gann, 1942).
3. Technical Analysis is the analysis of market data with technical indicators, so called, so as to highlight different features of the data. Charles Dow's 1884 Theory of Averages, is credited as the beginning of technical analysis as it is known today. (Murphy 1999: 23-30). Indicators include moving averages, Bollinger bands (a measure of standard deviation), On-Balance-Volume and oscillators of various kinds. Today, the term technical analysis is used to cover both chart reading and indicator use and is taught as such in professional courses[10]
4. Quantitative Analysis begins with Bachelier's speculation that the market is efficient and thus price movement must be a random walk (Bachelier 1900/1967). The mathematics of a random walk was well known and an analysis of markets in terms of it enabled the construction of portfolios of

---

7 The 'classic' text is Graham and Dodd (1934/1962).
8 Mandelbrot describes these techniques as "folkloric" (1997: 65).
9 For example, continuous charts for Japanese rice markets go back to the 1600s.
10 Such as those run by organizations such as The Australian Technical Analysis Association (www.ataa.com.au). The 'classic' text is Edwards and Magee (1948/2001).

stocks with defined risk profiles. Benoir Mandelbrot's study of price action in cotton (1997: 430) led him to question this received wisdom, and to develop another mathematics, which he called "fractal" to accommodate his analytic findings. Fractal mathematics has become a widely applicable tool in many fields, both analytic and generative, and continues to be the basis of contemporary quantitative analysis. The first case study, detailed below, is concerned with the application of sonification to such quantitative analysis.

5. Heterodox selection methods include random choice, also known as 'dartboard trading', astrology, numerology and other such the divination practices[11].

Motivated by the need to know *why* a market will move in one direction or another, fundamentalists study what they believe are the underlying *causes* of market movement. On the other hand, technicians, believing that all they need to know is reflected in the trading data, study the *effects*. The two approaches can contradict each other because

> *[M]arket price tends to lead the known fundamentals*. Stated another way, *market price acts as a leading indicator of the fundamentals* or the conventional wisdom of the moment. While the known fundamentals have already been discounted and are already "in the market," prices are now reacting to unknown fundamentals. Some of the most dramatic bull and bear markets in history have begun with little or no perceived change in the fundamentals. By the time those changes become known, the new trend is under way (Murphy 1999:6, his italics).

From a technical analyst's perspective, given the premise that the market leads the fundamentals, the fundamentals are reflected in the market price so their independent study becomes unnecessary, at least for speculative trading. On the other hand, radical fundamentalists consider that the technician's study of charts of previous price action and their speculation about future price movement on the basis of that study is a little too close to astrology, and quantitative analysts ('quants') generally used statistical analysis and stochastics to model the risk profiles of market indices, segments and individual securities as accurately as possible so as to assist in the construction of various types of portfolios.

To underestimate the risk of a portfolio is to court calamity, while overestimating it invites lower returns than might have otherwise been

[11] Sceptics would include fund managers, journalists–in fact all those who attempt to predict future prices–as closer to divination than they might be prepared to admit.

possible. This is still and area of active research (Farmer, Patelli and Zovko 2005) and the next section provides enough of an introduction to the problem domain to make the sonification case studies that follow understandable. A more comprehensive introduction to the time-variant characteristics of real markets and their stochastic approximations can be found in Mandelbrot (1997). This brief summary of the historical development of market data analysis is derived principally from Peters (1991), (Mandelbrot and Hudson (2004) and Sorbenette (2003). Edgard Peters work is of historical interest because it was written during the foundational research in complexity economics at the Santa Fe Institute (Mandelbrot and Hudson 2004: 261) during the same period leading up to the seminal conference in auditory display held there in October 1992. Unfortunately however, the two groups did not know of each other (Kramer 2008).

6.4 Capital Market Time Series

As market-time unfolds, the price of a traded security 'wanders' from value to value. This is known mathematically as a stochastic process, or a random function. Amongst the simplest types of such stochastic processes is a random walk, called so because it may be thought of as a simplified model for an individual walking on a line who at each point of time either takes one step to the right with probability p or one step to the left with probability $(1–p)$. Appendix 4 provides a succinct summary the basic principles of probability, statistics and time–series relevant to the present capital market time series studies. Because the domain over which this function is defined is a time interval, and prices move discretely (step from one value to another), this random function is also known as a discrete time series and securities trading datasets can thus be classified as discrete multidimensional time series whose moment-to-moment behaviour is analogous to geometric Brownian motion[12].

[12] Named in honor of the botanist Robert Brown who in 1827 studied the movement of pollen in water and the movement of airborne dust. Brown did not explain the origin of the motion. A Geometric Brownian motion is a stochastic process that is used to model processes that can never take on negative values, which is a characteristic of securities prices. The best-known stochastic process is the Wiener process, which is used for modeling Brownian motion and other diffusion processes. See Appendix A5.2 for more detail. It is widely applied in financial mathematics to model the evolution of security prices, including the ubiquitous Black-Scholes equations for continuous options pricing.

In 1900, Louis Bachelier[13], using methods that had been created for analysing gambling, conjectured that price fluctuations in day–to–day exchange–traded securities were independent random variables (Bachelier 1900). He provided little empirical evidence to support the assumption, yet his work, which also included techniques for analysing the value of government bonds and displaying options-related strategies, is today considered seminal[14]. Edgar Peters mirrors most writers when he says Bachelier's "thesis was revolutionary, but largely ignored and forgotten" (Peters 1991: 15). It was Albert Einstein's independent description that brought the solution for Brownian motion to the attention of physicists. (Einstein 1905).

The unexpected crashes of stock markets in 1929, and the depression that followed, stimulated mathematicians to attempt a better understanding of market action through statistical analysis. Paul Cootner published an anthology of quantitative analysis (1962) that became the basis for what is known as the Efficient Market Hypothesis, which Eugene Fama later formalised (1965). Simply stated, the Efficient Market Hypothesis assumes that the market price of a trading security reflects all that is known about it; the difference in price from one point in time to another, simply reflecting any new information about the security as it becomes known. After digesting the new information together with an assessment of the risks involved, the collective consciousness that is the market finds an equilibrium price. This is called the random walk version of Efficient Market Hypothesis. A random walk is a sufficient but not a necessary condition for market efficiency. That is, whilst market efficiency does not necessarily imply a random walk (it may be some other process), a random walk does imply market efficiency. Such a random walk is called Brownian motion. If Bachelier's conjecture is correct, as is assumed by the Efficient Market Hypothesis, prices would exhibit no autocorrelation[15] and a statistical analysis would reveal a normally-distributed, or Gaussian, probability density function (PDF)[16].

---

13 Bachelier was a doctoral student at l'Université de Paris.of the celebrated mathematician Henri Poincaré.

14 Unlike shares, whose prices are quoted as the last traded price, irrespective of when that trade occurred, options pricing is time-dependent so their value evolves continuously, irrespective of whether there has been a trade or not. So, the price of an option is a continuous random variable.

15 Autocorerelation is used frequently in signal processing for analysing time series of values. Informally, it is a measure of the similarity of ordinate values (such as prices)

As a result of the discrepancy between this theoretical perspective and his analysis of the way certain speculative prices, such as those of cotton, moved, Benoit Mandelbrot became dissatisfied with this simplified model. Mandelbrot is one of the significant contributors in the field and his technical monograph summarises a unique and influential perspective built over many decades (Mandelbrot 1997)[17]. His investigations showed that real markets exhibit much larger variability, as well as greater leptokurtosis[18] and skewness than a normal distribution. He realised that the market process could be better described by a Lévy flight.[19]

In addition to their distribution, markets also exhibit momentary autocorrelations (Peinke et al. 2004). Modern econometrics and financial engineering place considerable import on understanding such phenomena because the increased likelihood of extreme events, both positive and negative, indicates greater market volatility than if the markets are normally distributed, which, in turn, would impact on risk assessment, options pricing and portfolio theory in general. The principal statistical techniques used in market analysis and simulation today, use Bayesian partitioning (Bolstad 2004), variable length Markov chains and Monte Carlo simulations (Bühlmann 2000, Berg 2004) and Generalized Auto-Regressive Conditional Heteroskedasticity (GARCH) modelling (Engle 1982). There seems nothing intrinsic in these techniques to limit the application of sonification to them and preliminary discussions indicated that some researchers in the field are interested in experimenting with such techniques to assist in the comprehension of the their abstraction.[20]

---

over two different abscissa intervals (such as time periods). More precisely, it is the cross-correlation of a signal with itself. Nigam (1983: 53) provides a formal definition.

16 This and other statistical terms are explained in more detail in Appendix 5.2XX.

17 Peters (1991) and Sornette (2003) are readable for the mathematically disinclined and Mandelbrot himself has also co-written a more eclectic historical account that succeeds in making his own perspectives more widely accessible (Mandelbrot and Hudson 2000).

18 Fat tails and high peaks. Also called *heavy-tailed*. See Appendix 3.

19 Named after the French mathematician Paul Pierre Lévy. If the distribution is 'heavy tailed', the distance from the origin of a random walk would tend to a *Lévy skew alpha-stable* distribution. Its four parameters control location, scale skewness and kurtosis and it is argued that Lévy flights are a more accurate, if still imperfect, model of stock-market fluctuations. Mandelbrot, a friend and champion of the much-maligned Lévy (Mandelbot 1997:106; 430), proposed it, with α=1.7, to be that which cotton prices follow.

20 Personal communication with participants in the first Econophysics Colloquium, 14-18 Research School of Physical Sciences and Engineering Australian National University November 2005. http://www.rsphysse.anu.edu.au/econophysics/

6.5 Previous related sonification studies

The first users of financial market sonification were probably the bucket shop traders who were reputed to be able to differentiate the sounds of stock codes, and prices that followed, from the sounds made by the early stock-ticker machines as they punched recent trading information, telegraphed from an exchange, into a strip of paper tape (Lefèvre 1923/1994).

6.5.1 Using finance data

Janata and Childs (2004) suggest that Richard Voss may have been the first to experiment with the sonification of historical financial data: stock prices of the IBM corporation. This is possible, as Voss and Mandelbrot were research collaborators in fractal mathematics at IBM's Thomas J. Watson Research Center and he played a early seminal role in the visualisation of fractal structures and in the analysis of the fractal dimensions of music and speech (Voss and Clarke 1975, 1978).

Finance data was used by Kramer (1994) to demonstrate multivariate sonification–mapping techniques. This work was later summarized and published with sound examples (Kramer 1994b). Trading data included the weekly closing prices from (September 1987–March 1992) of a US stock index, a commodity futures index, a government T-bond index, the US federal funds interest rates, and value of the US dollar. Mappings were to pitch, amplitude and frequency modulation (pulsing and detuning), filter coefficients (brightness) and onset time (attack). Mapping concepts include redundant mapping and datum highlighting (beaconing).

Ben-Tal et al. (2002) sonified up to a year's end-of-day data from two stocks simultaneously by mapping them to perceptually distinct vowel-like sounds of about one second duration. A single trading day was represented as a single sound burst. The closing price for the day was mapped to the centre frequency, and the volume of trade to the bandwidth. These values were scaled such that the parameters for the last day of trade in each period corresponded to a reference vowel. Closing price was mapped to the number of sound bursts and volume to duration. They informally observed that they could categorise high volume, high price trading days as loud, dense sounds, while low volume, low price days were heard as pulsed rhythmic sounds.

Brewster and Murray (2000) tested the idea that traders could use sounds instead of line-graphs to keep track of stock trends when they are away from the trading floor. Using personal digital assistants with limited screen space over a wireless network, one month of (presumably intraday) price data for a single share was mapped to pitch via MIDI note numbers. Participants, all students whose previous trading experience was unreported, were required to try to make a profit by buying and selling shares while monitoring price movement with either line or sound graphs. Profits or losses were apparently gross, as trade transaction costs appear not to have been factored into the calculations. The experimental results showed no difference in performance between the two modes, but participants reported a significant decrease in workload when they used the sonification, as it enabled them to monitor the price aurally while simultaneously using the visual display to execute trades.

Nesbitt and Barrass (2000) also undertook a multi–modal sonification and visualisation study of market depth, to test whether subjects could predict the direction of the next trade. They used real data from a single security's order book. The visualisation used a landscape metaphor in which bid and ask orders (to buy and sell), were 'banked' on either side of a river, the width of which thus represented the size of price gap between the highest bid and the lowest ask, known as the 'bid-ask spread'. A wider river implied slower flow (fewer trades) and so on. The sonification employed the metaphor of an open-outcry market. A sampled male 'buy' and a female 'sell' voice displaying a discretely partitioned dataset (price, volume, price-divergence) was mapped into a discretely partitioned three-dimensional 'importance space' (pitch, loudness, stereo-location). This experimental design illustrates how sonification can be used to assist the apprehension of data segmentation such as where the trajectory of a parameter under focus changes.

Janata and Childs (2004) developed *Marketbuzz* as an add-on to conventional trader's terminals, such as those by Bloomberg, for the sonification of real-time financial data. They used it to evaluate tasks involving the monitoring of changes in the direction of real-time price movements, with and without auditory or visual displays. A significant increase in accuracy using auditory displays was reported, especially when traders were visually distracted by a simultaneous diversionary “number-matching” task. Childs is possibly the most experienced researcher in the application of sonification to monitoring real, rather than simulated, markets as exemplified in his highlighting of significant price movements relative to opening price, as well as of the continually changing features of Stock Options (Childs 2005).

Mezrich, Frysinger, and Slivjanovski (1984) developed a dynamic representation, employing both auditory and visual components, for redundantly displaying multiple multivariate time-series. Each variable was represented by a particular timbre. The values of the variable were mapped to pitch. The analyst could focus on a subset of the data by interactively brightening or muting individual variables and could play the data both forwards and backwards. Subsets of the data could be saved and juxtaposed next to each other in order to compare areas where the data might be similar. In almost all cases, the sonified data performed as well as or better than the static displays.

Simulated price and volume for a single stock were sonified in an experiment on the orthogonality of a multivariate mapping (Neuhoff, Kramer and Wayand 2000). Price was mapped to frequency and volume to amplitude. The experimental task was to judge the price and volume from the sound. Results showed that dimensional changes interacted, creating asymmetrical perceptual distortions of the data set. The researchers admitted, however, that their results were limited because “[t]rue market conditions produce price and volume interactions that are not as regular as the stimuli presented…”. Nevertheless, they reiterate a common theme of all researchers using parameter-mapping techniques; that great care is needed when using lower level acoustic dimensions to represent multidimensional data. Simulated stock data have also been used by others to study participant perceptions of data density, the number of trend reversals and point estimation (Smith and Walker 2005, Walker and Nees 2005, Nees and Walker 2007, 2008). While these studies are of interest from generic parameter-mapping and audio-graph perspectives, it is not clear that there is anything in the findings specific to financial data, as, unless it is generated using advanced techniques such as those discussed in the following sections, fictional data is unlikely to exhibit the same structural characteristics as real financial time series data. Research focused on the comprehensibility of auditory graphs using acoustic parameter-mapping techniques, was critically reviewed by John Flowers (2005).

Two other types of sonifications of securities data demonstrate different motivations. The first is *sMax*, a toolkit for the auditory display of parallel internet-distributed stock-market data for a musical installation Ciardi 1994). *sMax* uses a set of Java and Max modules to enable the mapping and monitoring of real time stock market information into recognizable musical timbres and patterns. The second is a rendering of dynamic data specifically for peripheral auditory monitoring of anaesthetised hospital patients. The system reads and parses simulated real-time stock market data that it processes through various gates and limiters to produce a changing soundscape of complementary ecological sounds (Mauney and Walker 2004).

6.5.2 Audification

In the 1990s Steven Frysinger experimented with playing back market price data directly as a sound waveform. He reported that he found the results difficult to interpret, probably because the stock market does not follow physical-acoustic resonance laws resulting in natural or 'ecological' sounds that can be understood from everyday listening experience (Frysinger 1990, 2005). There appears to be no further reports of security data audification prior to the work reported in this chapter. Chris Hayward also suggested that another reason audification fails for arbitrary data such as stock market figures or daily temperatures is the amount of data required, even at low sampling rates, to make a sound with a duration long enough to reveal valuable information to the listener (Hayward 1994).

In summarising previous work on seismic audification, Hayward reported both Speeth's original experiment on discriminating the *seismic sounds* of earthquakes from atomic explosions and Frantti's repeat of it with a larger number of participants with less training and data from diverse locations. Frantti found a lower average accuracy and a wider variance in participant performance, which was also critically affected by the audification's time-compression ratio and the number of repeat audits. Neither study used trained seismologists, nor did participants have any interactive control (Speeth 1961, Frantti and Leverault 1965, Hayward 2002).

Concentrating on single wavelet and quantitative questions, Hayward indicated a number of solutions to the difficulties encountered as well as some strategic extensions to planetary seismology in general. Florian Dombois (2001) reported that he could hear seismological station-specific characteristics his time-compressed audifications. He informally found that, over time, overall information of a dynamic state was better comprehended with audification, whereas visualization was more effective when a detailed analysis of a single wavelet was required. He developed a unified acceleration method to make records taken under different meteorological and seismic conditions more compatible and in a later report on the state of research in auditory seismology, documented several other investigations in the field in the 1990s and much earlier (Dombois 2002a, Dombois 2002b). He reported an increase in interest among seismologists; this is also evidenced by the recent reporting in the popular media of the audification of stellar seismology (Gosh 2008).

Using data from a helicopter flight recorder, Pauletto and Hunt (2005) showed that audification can be used as an equally effective alternative to spectrograms for the discernment of complex time-series data attributes such as noise, repetitive elements, regular oscillations, discontinuities, and signal power. However, another empirical experiment found that the use of audification to represent data related to the rubbing of knee-joint surfaces was less effective at showing the difference between normal and abnormal signals than other sonification techniques (Rangayyan et al. 2001).

6.5.3 Sonification of stochastic functions

Aside from their use in algorithmic music composition, which will be discussed briefly at the end of the chapter, stochastic functions have received little attention in sonification research. Perhaps the first was a study of the use of parameter-mapping and physical model sonification in a series of experiments in monitoring the performance of Markov chain Monte-Carlo simulations for generating statistical data from higher dimensional probability density functions (Hermann, Hansen and Ritter 2001). The inclusion of some sound-generating tools in the statistical package *R* has the potential to generate wider interest. The authors reference the previously mentioned paper and also exemplify its use in tuning a parameter in the

Hybrid Monte-Carlo algorithm (Heymann and Handsen 2002). Informal auditing of a technique to sonify, using amplitude modulation, cross-correlations in irregularly spiking sequences that resemble a Poisson process led to the postulation that the use of sonification for time series analysis is superior to visualisation in cases where the intrinsic non-stationarity of an experiment cannot be ruled out (Baier et al. 2005). Using audification, a differentiation study of the perceptualisation of some statistical properties of time series data generated using a Lévy skew alpha-stable distribution whose four parameters control location, scale skewness and kurtosis[21] found evidence that kurtosis correlates with roughness or sharpness of which participants were able to distinguish differences (Frauenberger, de Campo and Eckel 2007). Time–series data were generated by shaping a uniform distribution (white noise) with a cumulative probability density function, in a manner similar to that used by Xenakis in his *ST* series of compositions (Xenakis 1971: 136-143). This research found there to be no evidence that skewness in their data was perceivable, which provides empirical support for the initial findings (Worrall 2004) of the experiments outlined below.

6.6 Experiments: Audification of security returns

This experiment sought to (a) discover a way to directly audify a capital market trading dataset that preserved its autocorrelation characteristics[22] and (b) ascertain whether such a dataset can be aurally discriminated from an audification of a statistically equivalent uncorrelated dataset and from a known, stochastically generated time series. The null hypothesis in each case was that no distinction could reliably be made between the audifications. Trading data is most frequently plotted ordinate–price against abscissa–time, as illustrated in Figure 6.3, which clearly does not oscillate around. The task is thus to generate a new dataset that does oscillate in such a way as to preserve the information content, that is, the data relations, required.

---

21 This function is used in the analysis of critical behaviour and in modelling self-similarity in financial data.

22 Autocorrelation is also called cross-correlation. A statistical analysis of time series data is concerned with the distribution of values without taking into account their sequence in time. Changing the sequence of values in a time series, known as decorrelation, completely destroys the frequency information while keeping the statistical properties invariant. See Appendix 3 for a longer explanation.

6.6.1 The XAO dataset

The dataset chosen is twenty-two years (21April 1986 to 18 April 2008) of the daily closing price of All Ordinaries Index (ticker XAO) of the Australian Securities Exchange (ASX). It is comprised of 5725 datapoints[23] and shown plotted in Figure 6.3. The first task was to is to find a way to overcome the non-resonance problem, as discussed in Hayward (1994) and referred to earlier in §6.5.2; one that transforms it to be suitably oscillatory while preserving the correlational integrity of the dataset. An equivalent problem is to be found in quantitative analysis, as observed by computer scientist Stephen Skiena (1994):

> The *price* of an asset as a function of time is perhaps the most natural financial time series, but it is not the best way to manipulate the data mathematically. The price of any reasonable asset will increase *exponentially* with time, but most of our mathematical tools (e.g. correlation, regression) work most naturally with linear functions. The *mean* value of an exponentially increasing time series has no obvious meaning. The *derivative* of an exponential function is exponential, so day-to-day changes in price have the same unfortunate properties.

The method chosen was to convert the data to market returns. For an asset whose price changed from p_t at time t to $p_{t+\delta t}$ at time $t+\delta t$, the simple linear return R_{lin} is defined as $R_{lin} = p_{t+\delta t} - p_t$

Because prices tend to move exponentially over longer timeframes time, that is, in percentage terms, a better measure than R_{lin} is the ratio of successive price differences to the initial prices. These ratios are known as net linear returns (Sornette 2003:36):

$$R_{net} = \frac{p_{t+\delta t} - p_t}{p_t}$$

Figure 6.4 is a plot of net returns of the XAO dataset, generated using the code in Appendix 6.1. The insert is of the first 500 samples, as in the insert in Figure 6.3. Table 6.1 summarises the statistical properties of these returns, clearly showing that they are not normally distributed. The single largest Net Return, clearly visible in and Figure 6.4 was on 20 October 1987 ("black" Tuesday), the largest one-day percentage decline in stock market history.[24] The difference between this largest minimum and the second-largest

---

23 The XAO is the broad Australian market indicator, a composite of the 500 largest companies, weighted by capitalisation, which are listed on the exchange. Contextual details are available at http://www.asx.com.au/research/indices/description.htm

24 See http://en.wikipedia.org/wiki/Black_Monday_(1987)

minimum is 62% of the total returns space. This is clearly illustrated in the histogram of Figure 6.5, which illustrates the frequency of net returns; the single minimum and second-largest minimum are circled but barely visible.

Figure 6.3. Graph of 22 years of XAO datapoints. The detail is of the 1st 2 years. [25]

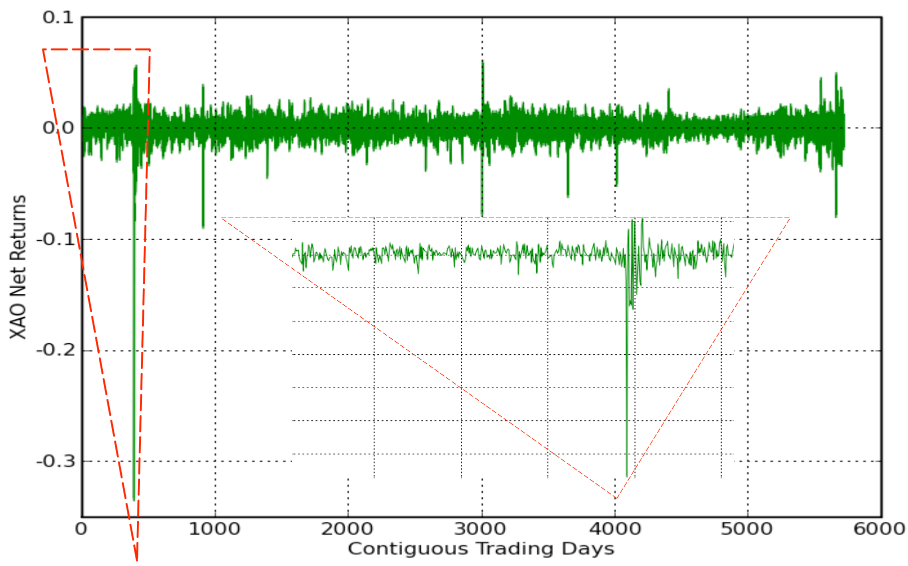


Figure 6.4. A plot of XAO net returns.

#samples	min sample	max sample	arith. mean	variance	skewness	kurtosis
5725	-0.333204213	0.058862074	2.174884e04	9.568588e-05	-7.6491182	241.72988

Table 6.1. Statistics for XAO net returns.

[25] All plots are produced from within *SoniPy* using *matplotlib*. Code can be found in Appendix 4.

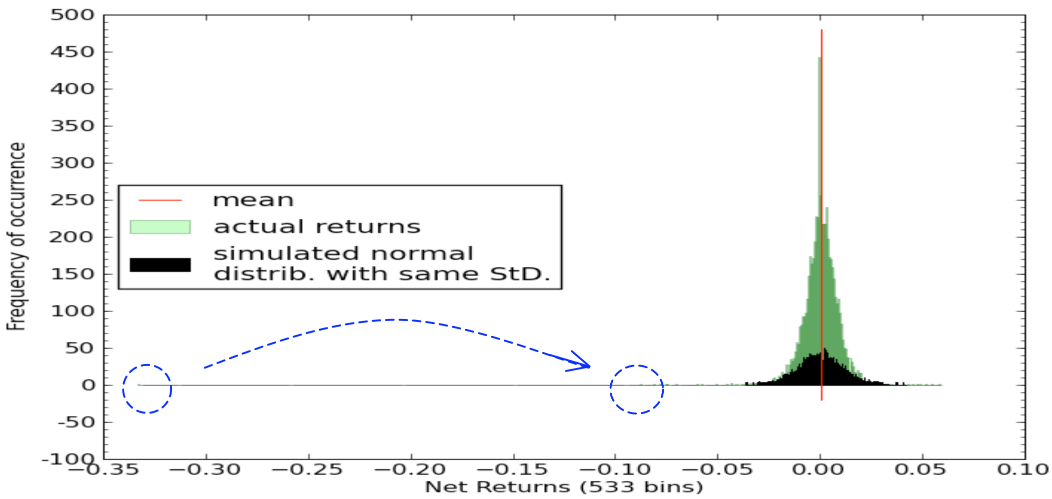


Figure 6.5. A histogram of net returns that illustrates the proportion of the dataspace allocated to a single negative outlier.

As the use of the audification technique in this context is to aurally compare this dataset with an uncorrelated dataset with the same statistical characteristics, moving a single sample will not only not invalidate the experiment but enhance the perceptibility, because it will make fuller use of the audio amplitude space and thus improve the signal-to-noise ratio.

So, despite its anecdotal interest, an 'audacious' clipping the largest minimum sample to that of the second-largest minimum, was performed and the resulting returns plotted in Figure 6.6 and the resulting histogram is show in Figure 6.7, which illustrates both the skewness and kurtosis of the dataset, when compared to a normal distribution. The *size* of the sample-bins of this histogram is kept constant those to those of the Figure 6.5 histogram by decreasing the *number* of bins. Of interest is the asymmetry of the outliers (the data at the extremities): there are more of the negative variety than positive, and negative ones exist further from the mean; even more so when considering this is the clipped dataset. Audio Example 1 is the direct audification of the clipped dataset.

For comparison, the net returns are decorrelated (by randomsing the time order of the samples). Figure 6.8 shows the plots of both the correlated and decorrelated datasets. A number of features are visually apparent. Both have long tails but they appear more evenly distributed throughout the decorrelated dataset, contributing to its more chaotic visual appearance, whilst the correlated dataset appears to have periods of increasing (trapezoid), low (circle), and ramped (diamond) volatility.

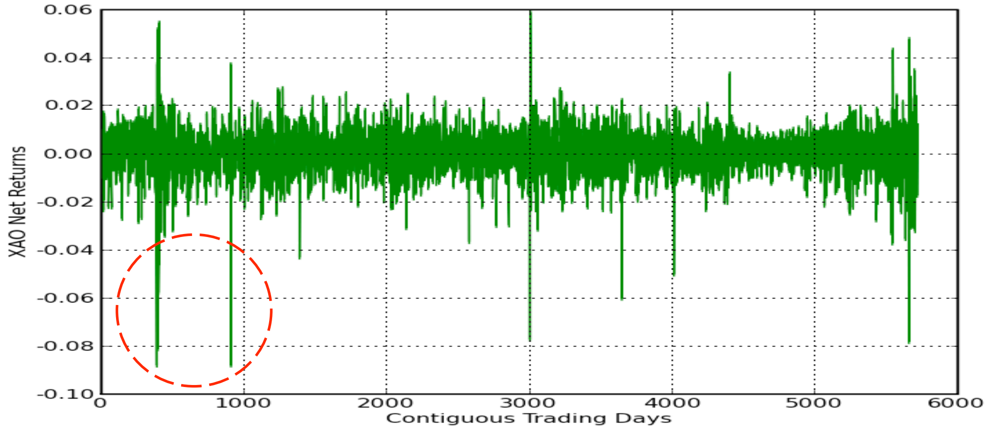


Figure 6.6 Plot of XAO net returns, clipped so as to be suitable for audification.

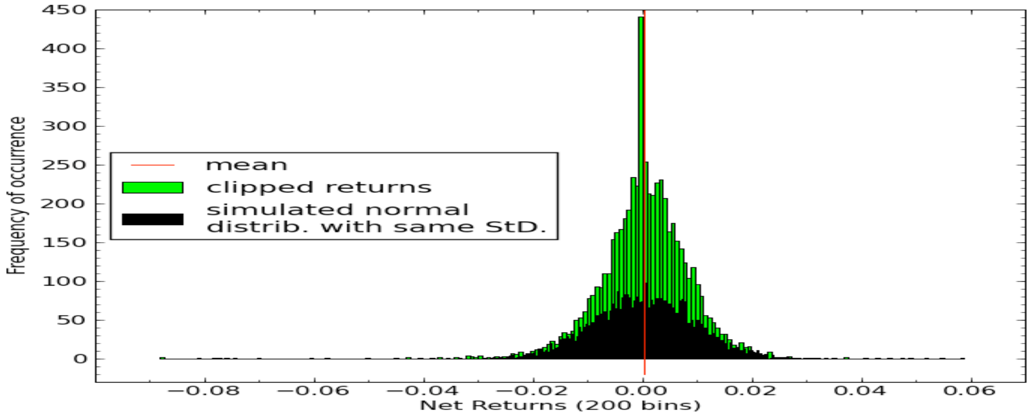


Figure 6.7. Histogram of the clipped net returns with a simulated normal distribution with the same standard deviation and number of datum is overlayed for comparison.

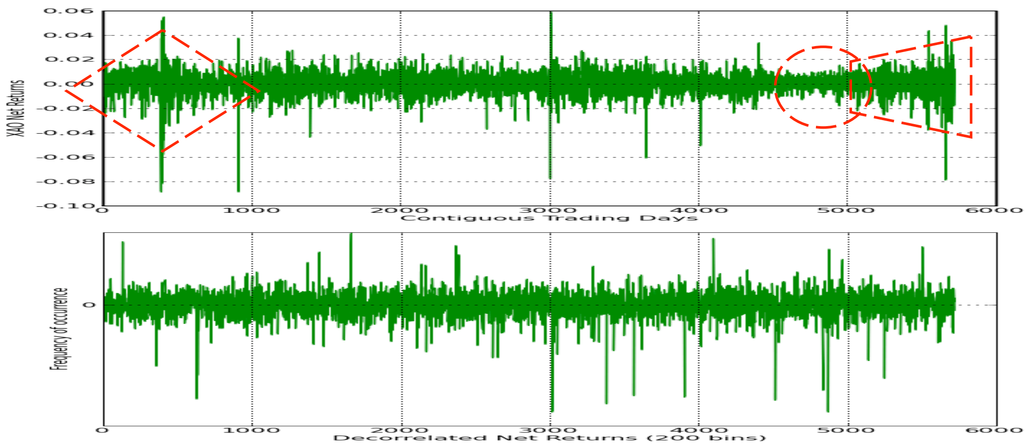


Figure 6.8. A plot of the correlated (top) and decorrelated net returns.

6.6.2 Experiment 1

In order to test whether the raw and decorrelated data sets could be distinguished aurally, a number of chunks of audio were prepared with the same number of samples as the Net Returns:

A. Uniformly distributed B. Normally distributed (Gaussian)
C. Decorrelated Net Returns D. Net Returns

Audio 6.1 A sequence of four differently formed chunks of audio noises repeated four times for evaluative comparison.

These can be heard four times at a sample rate of 8 kHz (in the order A-B-C-D) in AudioExample01. Each audio chunk is approx. 0.7 seconds duration. There is a one-second gap between each chunk and a three-second gap between repeats. The following informal observations can be made:

- The uniformly distributed noise (A) is clearly distinguishable from the Gaussian (B). This distinction is unsurprising: it is that between white and band-limited noise in electronic music parlance. As would be expected, the uniformly random noise sounds "brighter" because of the comparatively greater prevalence of higher frequencies.
- The raw and decorrelated returns (D and C) are clearly distinguishable from A and B: Qualitatively, they sound rougher or grainier, and they have less evenly distributed spectral energy than A or B. This can be interpreted as a result of the increase kurtosis, as reported earlier (Frauenberger de Campo and Eckel 2007).
- The use of an 8 kHz sampling rate was used after some initial experimentation with phigher and lower values. There appears to be an optimalisation compromise between longer duration and higher frequencies.

6.6.3 Experiment 2

A second experiment was conducted to ascertain whether or not the raw and decorrelated chunks could be aurally distinguished from each other. Another differently decorrelated returns (E) was generated in the same manner described for C, above, and three files were prepared with the following sequences in which the original raw returns (D) was placed in first second and third place respectively:

File 1. D-C-E File 2. C-D-E File 3. C-E-D

Audio 6.2 Three sequences each of three audio chunks, two of which (C & E) are decorrelated versions of the Net Returns (D).

These files can be found on the accompanying CD as AudioExamples02–04. The listening task, on random presentation of these audio files, was to determine, in each case, which of the three chunks sounded different from the other two. This was found to be (informally) possible, though a harder task than that of Experiment 1, and more dependent on a narrower band of sampling rates. Above 8kHz the characteristics described earlier seem to disappear. Below 3-4kHz the roughness created by the individuation of large-valued samples meant that the principal means of identifying the raw returns was probably more by its invariance across all chunk presentations than by direct chunk comparison. Between 4 kHz and 8 kHz sampling rate, a distinct, though subtle, amplitude modulation is observable in the raw returns that seems not to be present in the decorrelated ones.

6.6.4 Experiments 1 and 2: observations

The observation of amplitude modulation in the raw Returns in Experiment 2 suggests an empirical study to isolate the determining factors might be useful. This would require the preparation of additional, unrelated, raw returns datasets of the same sample size. In these experiments, sample rates were chosen heuristically as no other method for determining them was known. Yet the choice clearly influences the perceptibility of pattern, as was also observed by Dombois in his seismic audification studies (2002b).

It appears that the simple technique of using Net Returns is correlation-preserving and that this correlation is aurally perceivable, so it may be possible to apply the same technique to other similarly-structured time-series datasets. An interesting extension study would be the application of the technique to the functional *simulations* of financial-market-like time-series, an active field of econometric investigation in which large datasets can be generated when needed.

The effect of the size of the dataset also needs to be isolated, as, whether or not higher sampling rates on larger datasets reveal more or other distinguishing features is currently not known. Although the directness of audification makes it appealing as a technique, more might be gained through the use of other less 'data hungry' techniques, in consideration of which Mapping Experiments 3 and 4 and 5 were designed using the datasets of Experiment 1.

6.7 Experiment: Homomorphic mapping sonification

This experiment was designed to test a simple proposition: That the four datasets of Experiment 1 could be identified as distinct datasets under homomorphic mapping into pitch-time auditory space. The four datasets are

A. Uniformly distributed dataset B. Normal (Gaussian) distributed dataset
C. The set of XAO Net Returns D. The set decorrelated XAO Net Returns

A homomorphic mapping is one in which the changes in a dimension of the auditory space track changes in a variable in the dataset, with only as few mediating translations as are necessary for comprehension (Kramer 2004b: 26), and was discussed more fully in chapter 2. In this experiment, time in the dataset was mapped to time in the auditory display and sample value was mapped to pitch deviation (both positive and negative) from a centre frequency. A csound instrument, illustrated in Figure 6.9, was constructed to produce this mapping.

Structurally, this is a basic frequency modulation instrument in which an ADSR[26] for controlling modulation index is replaced by a sample buffer of the AIFF samples. These samples, which can be read directly from an audio file, are used sequentially to control the frequency deviation from a user-defined centre 'reference' carrier frequency. Figure 6.10 shows a MacCsound controller interface to this sample modulator. It provides a flexible means for heuristically adjusting the dimensional mappings. The setting shown is for frequency-modulating a 440Hz tone over a range of four octaves at the rate of 480 modulations per minute from the Net Returns distribution. Code 6.1 shows the csound implementation. Full implementation tools are available on the accompanying disk in the *Experiment3* folder.

[26] Computer music parlance: An ADSR is an Attach-Delay-Sustain-Release envelope shaper, a common tool for controlling the amplitude evolution of computed sounds.

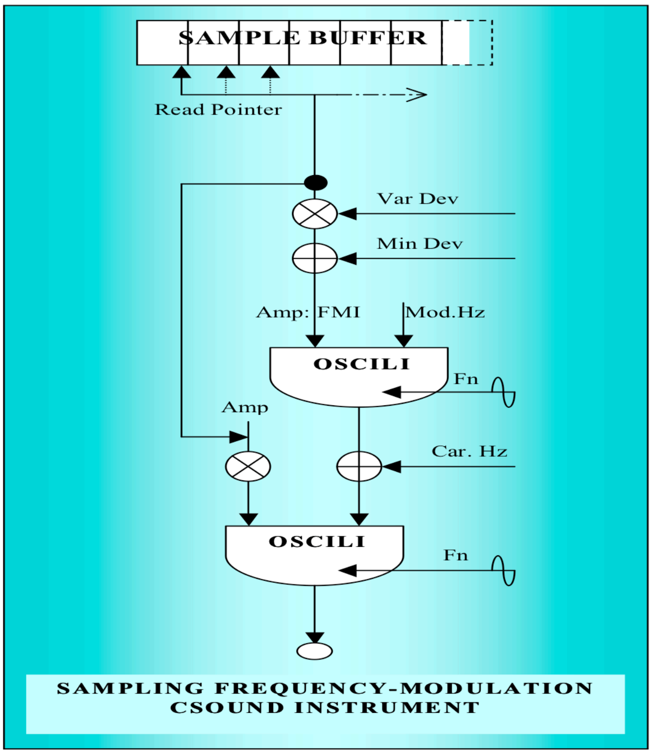


Figure 6.9. A graphic illustration of the csound instrument used for the homomorphic mappings experiments.

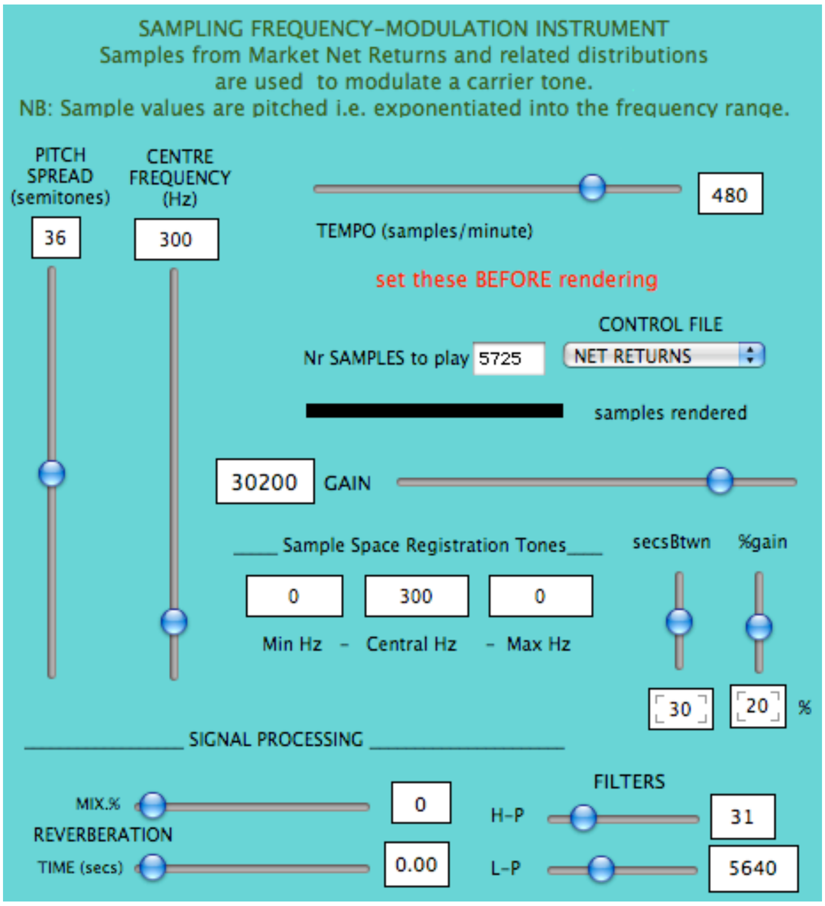


Figure 6.10. User interface of the sampling FM instrument.

```
; ----------------------- SAMPLING FREQUENCY-MODULATION INSTRUMENT ---------
; For reading audiofile into table so samples can be used as Freq Modulators
sr        = 44100
kr        = 4410
ksmps     = 10
ga1 init 0          ; init a global audio receiver/mixer
instr 1 ; ---------------------- INSTRUMENT 1  ----------------------------
;    ----------- controller inputs ---------------------------------------
     kgain     invalue    "gain"       ;  render time - if possible
     kspread   invalue    "spread"     ; get pitch spread from slider
     kCentreHz invalue    "centreHz"   ; centre freq

     ktempo    invalue    "tempo"      ; get tempo from slider
     itempo    = 60/i(ktempo)/2        ; get tempo from slider

     kNoiseOffset   = 3                ; weird listIndex of noisetypes
     kNoiseType invalue "noiseType"    ; choose which type is rendered
     kNoiseTypet    = kNoiseType + kNoiseOffset ; weird indexing
     iNoiseType     = 3 + i (kNoiseType)

     kNrSamples invalue "nrSamples"    ; get Nr samples to render
     iNrSamples = 2 * i(kNrSamples)    ; Nr samples in the file

     kmixpercent    invalue "reverb"   ; NB is also fed to reverb instr.
                                       ; To control OP % of dry signal
;    ----------- FM component  -------------------------------------------
     kCount    = 0
;    Create an index into the GEN01 sampleArray to be ++ed accord to tempo.
     kcps init 1/(iNrSamples * itempo)  ; Nrsamples in the file; 5725
     kndx phasor kcps                   ; kndx=(0-1) norm(sampTable (index))
     kCount = kCount +1                 ; pick new samp every 'beat'
     printks "\nknsx = %f, ", 10, kndx
     ktabOP    tab kndx,  iNoiseType,  1
     kFreq = kCentreHz * powoftwo (kspread/2 * ktabOP/12)
; printks "Count=%6.1f tableVal= %5.3f Freq=%5.3f\n",kCount,1,ktabOP, kFreq
     a1 oscili  kgain,     kFreq, 1     ; Sine: tableValss to FM centreHz.
     out a1 * 1-(kmixpercent/100)       ; and output the result
     ga1 = ga1 + a1                     ; OP to global buff
endin

instr 2 ; ---------------------- INSTRUMENT 2  ----------------------------
;    ----------- controller inputs/outputs -------------------------------
     kgain     invalue "gain"          ; gain of FM samples
     kgainPcnt invalue "regisGain"     ; %FM samp. gain for tick registrats
     kregisGain = kgain * kgainPcnt/100
     kregisRate    invalue     "regisRate" ; repeat rate of ticks
     kspread   invalue    "spread"         ; get pitch spread in ST
     kCentreHz invalue    "centreHz"       ; centre freq
     outvalue "lowerHz", kCentreHz - powoftwo (kspread/24)
     outvalue "upperHz", kCentreHz + powoftwo (kspread/24)
     a1 oscil kregisGain, kregisRate,1     ; FM kCentrHz by set amount
     a1 oscili a1,      kCentreHz,  1      ; A sine, using centreHz.
endin

instr 99 ;     --------- INSTRUMENT 99 -- REVER B---------------------
     kreverbtime   invalue     "reverbTime"
     kmixpercent   invalue     "reverb"
     a3 reverb ga1, kreverbtime      ; reverberate whatever is in ga1
     out a3 * (kmixpercent/100)      ; and output the result
     ga1 = 0                         ; empty receiver for next pass
endin
```

Code 6.1 The csound instrument used to implement sampling frequency modulation

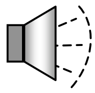
With this controller, it is possible to dynamically adjust the pitch spread and centre frequency during audition. In addition to the sample modulator, the sonification also uses has a simple audio 'tick' registration tone generator that acts as an auditory reminder of the upper, lower and central limits of the current render. Both its frequency-of-occurrence (*secsBtwn*) and relative loudness (*%gain*) are adjustable.

AudioExamples05–09 provide various sonic realisations of the homomorphic mappings of the Net Returns samples generated for Experiments 1 and 2:

A. Uniformly distributed dataset
B. Normally distributed dataset
C. The set of XAO Net Returns
D. The set of decorrelated C.

For consistency, all are rendered with the settings illustrated in Figure 6.10:

- AudioExample05 is a series of twenty-second 'snapshots' of each of the four sample sets A, B C and D.
- AudioExample06 is of A.
- AudioExample07 is of B
- AudioExample08 is of C.
- AudioExample09 is of D.



Audio 6.3 Audio examples of four audifications: uniform, normal, net returns and decorrelated net returns.

6.7.1 Experiment 3: observations

The following characteristics appear invariant under both translation (transposition) and linear scaling of the gamut if the standard psychophysical non-linearities, such as Fletcher-Munson equal loudness contours (Cook 1999) are allowed for:

- The difference between A, and B is easily noticeable, at least to a musically trained listener, as it was in the audification experiments, discussed earlier.
- The homomorphic mapping of A can be observed to be evenly spread across the pitch gamut, while the Normal distribution B can be observed to be more closely clustered around the centre of the gamut, the mean.
- C and D are noticeably different to A and B. Whilst both C and D appear to have short trending auto-correlative sequences, those of C (the net

returns) appear more consistently and when they do they appear to last for longer periods of time.

An attempt to 'enhance' the homomorphic mappings by smoothing the sample transitions using bandpass filtering and reverberation, added noisy artifacts that were distracting, not enhancing. This *had* been anticipated on the basis that such signal processors are designed for inputs with complex spectra, not simple sine-tone modulated signals, but it was useful to confirm that the 'starkness' of the carrier sine-tone with acoustic transients created by the transitions between the modulating data samples, created a superior sonification with less perceptual dissonance, than the results of using of a complex timbre, as a conventional musical aesthetic might suggest.

6.8 Experiment: Parameter mapping of intra-day data

The opportunity to work with intraday data trading engine data was afforded by an exploratory project with the Capital Markets Cooperative Research Centre[27], a private-sector/university partnership, through a formal agreement between the parties. The CMCRC had had some prior experience of a dynamic visualisation of intra-day trading engine data, which they reported their clients had found attractive as a foyer exhibit. Their initial non-committal position regarding the potential usefulness of sonification was qualified by their appreciation of the possible application for the visually challenged, so they were intrigued enough to fund a pilot study. They were not, at least initially, inclined to engage on a specific collaborative task[28], so a broad question was agreed to, namely:

> Can the sonification of data from the capital markets provide data domain experts and others with insights into activity in these markets not afforded by other display techniques?

6.8.1 The complete dataset

The complete dataset contains one financial quarter of a time-sequenced trading engine log of a generic market: approximately 5 Gigabytes of ASCII characters, in flat–file format. Structurally, this dataset is a multiplexion of all the on–market TRADEs with the incremental adjustments to the orderbooks of approximately 3000 trading instruments. It had to be extensively cleaned

---

27 http://www.cmcrc.com/

28 This appears to not be an unusual scenario.

before being statistically analysable in preparation for sonification. While the task was very time-consuming and laborious, it was essential to achieving a workable database. It also provided something of an opportunity to get to know the data itself before attempting to sonify it, the importance of which was discussed earlier (§5.2). Appendix 5 outlines the resultant dataset's Metadata Specification.

The data cleaning method was to iteratively develop and modify a series of active filters built around *Python*'s regular expression-matching capabilities, until one day's data parsed correctly. Most of the data from other days then required only minor adjustments to the filters; however, another day's data was particularly corrupt, and required extensive 'personal' attention. The final filter code is listed in Appendix 6.2. It does not reveal process used to arrive at it, nor the superiority of an interpreted rather than compiled language for undertaking such incremental fastidiousness.

Some of the limitations imposed to make the dataset generic ruled out certain types of internal comparisons, such as day–of–week, GICS[29] market sector or index membership, meaning that it was not possible to group these securities by other than by abstracted statistical characteristics such as value and frequency of trades.

6.9 Experiment 4: Information Sonification: $value at Time[30]

The aim of Experiment 4 was to provide a means by which a musically-naïve listener[31] could aurally observe something of the nature and extent of the way value, measured in monetary terms, changed ownership during a trading day. There is no simple relationship between the traded *price* of an individual unit of a security and that of an individual unit of another security[32]. However, one way to meaningfully compare securities being traded is by comparing the monetary value of trades ($value hereafter). To effect this

---

29 Global Industry Classification Standard. See Footnote 5.

30 Experiment 4, as reported here, was preceded by a trial of mapping each <securityID>'s trades at each moment into a 2D (frequency, amplitude) that then became a component of the overall spectrum for that moment in time.

31 That is, to use sound relationships which do not require musical training to identify.

32 Because the actual $price is related to the number units of a security a company has on issue, not to the value of each of such units.

sonification, intraday TRADE[33] data was adapted to a model of $value; that is, one that reflects the relative importance of trades involving various amounts of a finite resource (money). Without wishing to overstress the point, the task is thus of the sonification of (human-valued) information.

<u>The sonification task</u>
To sequentially sonify the total $value of all TRADES at each second in the trading day.

Perceptually, the size of an object is habitually related to its mass, as evidenced by the surprise when picking up a large piece of pumice. Portentous objects and events in the natural world are more usually associated with lower-pitched, less frequent sounds. For example, there appears to be many more small birds than large ones, and so on.[34]. A basic statistical analysis of the TRADE data reveals proportionally fewer high-$value trades than low-$value trades, indicating that $value is in line with the principles outlined, as symbolically illustrated in Figure 6.11.

Similarly, there appears to be an inverse relationship between the *duration* of an event type and its *frequency* (of occurrence). When compared to smaller events, larger events in the same domain occur less often and last for a longer period of time than smaller ones: the snap of a twig is more frequent and shorter than the crash of a tree, the scurry of mice more frequent than a stampede of elephants etc. So, when sonifying larger $valued trades at lower frequencies, it is necessary to increase their relative duration, comparatively.

These generalised relationships are also in evidence in the evolution of the logarithmic 'mappings' between psychophysical, physical and psychological phenomena. For example, it takes longer for the pitch of sound in the range of 30–50 Hz to be determined than it does for a pitch in the 300–500 Hz range. A similar psychological relationship can be observed in changes in perceived financial value; accounting for why $value is measured in the percentage gain or loss rather than in absolute amounts. This exponential relationship is also in evidence in the distribution of $value *between* securities in the market as a whole, as illustrated in Figure 6.12.

---

33 Capitalised words are Order Types and words enclosed with < and > are Order Type Fields as defined in the *Metadata specification for the CM generic dataset*.

34 For an expanded description of these phenomena as they relate to film, see Branigan (1989).

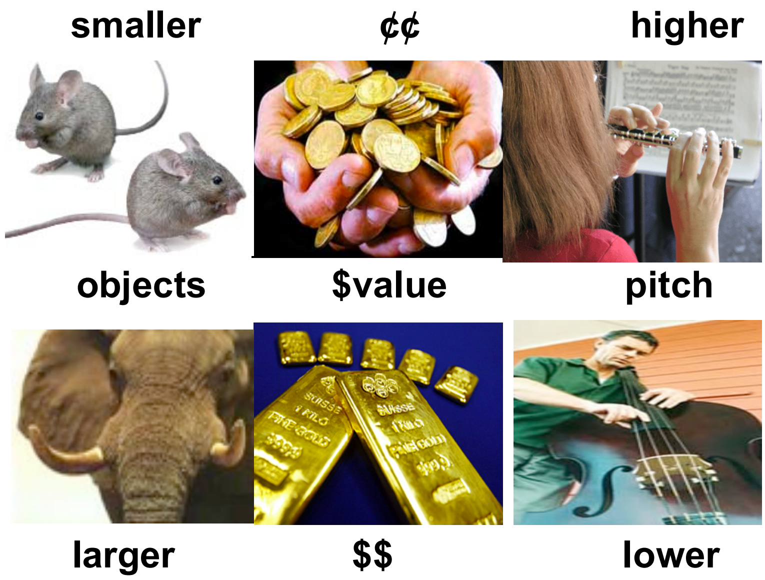


Figure 6.11. Symbolic representation of the relationship between size, value and pitch

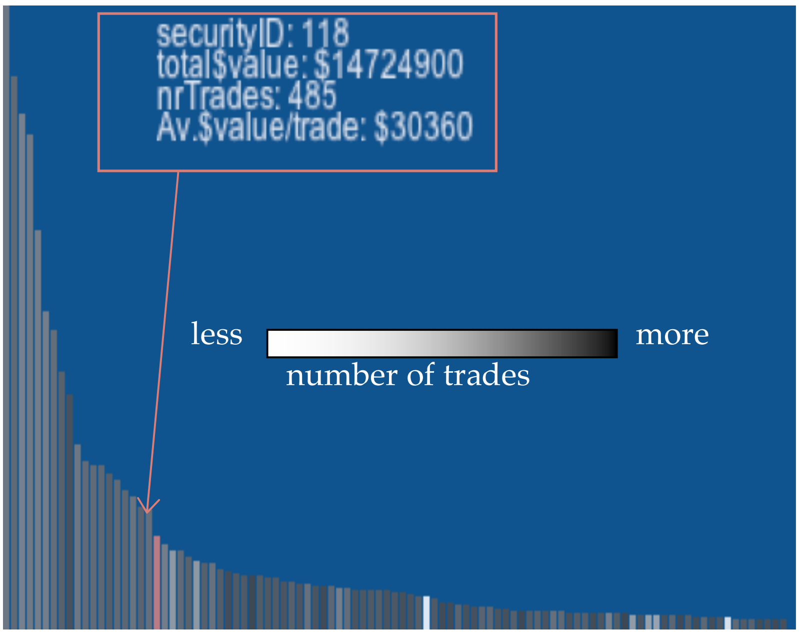


Figure 6.12. The distribution of $value traded for individual securities in a single trading day. Shading indicates the number of TRADEs for each security.

So, for the reasons just outlined, in this experiment, a proportional *inverse* mapping was applied between sound frequency and the total $value of all trades at each moment in the trading day. The psychoacoustic mapping as are illustrated in Figure 6.13 between pitch and duration, and in Figure 6.14, between a tone's onset–time and pitch.

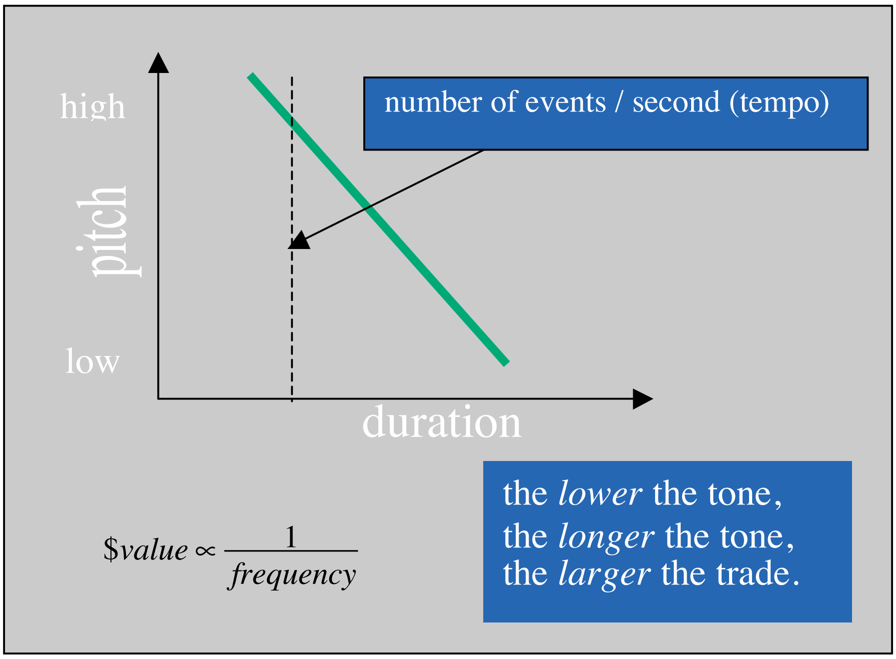


Figure 6.13 The principle information mappings $value is inversely–proportional to pitch and the lower the tone the longer the duration. Notice that the pitch (green line) is linear, implying and exponential frequency scale.

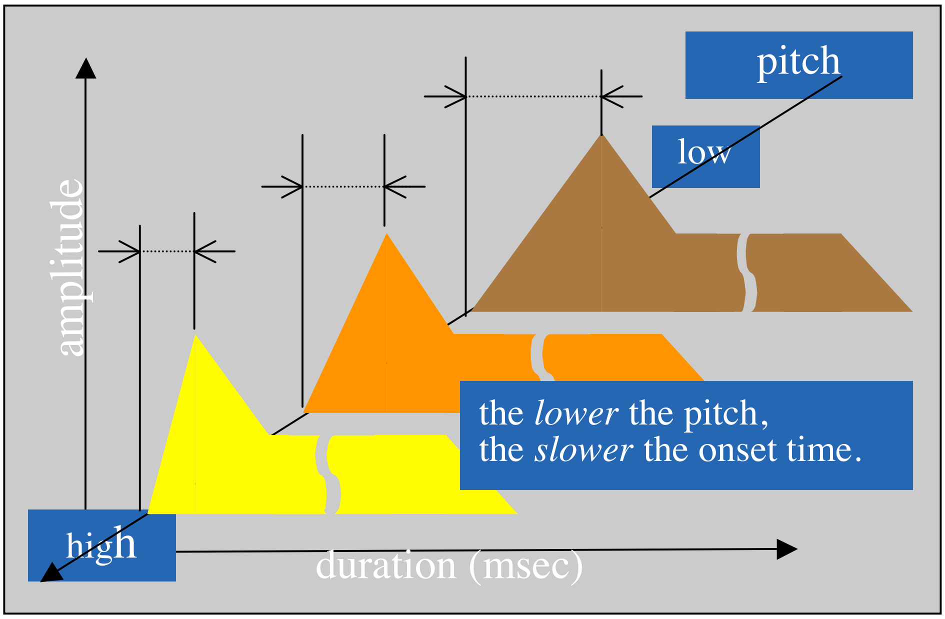


Figure 6.14 A second psychoacoustic adjustment: larger $value trades (lower–pitched) have slower onset–times in keeping with physical characteristics of material resonators.

A further psychoacoustic adjustment made was a very basic 'inverse Fletcher–Munson curve of equal loudness'; mapped to counter–balance the known psychoacoustic phenomena that the centre of the pitch gamut of human hearing is more amplitude–sensitive than the extremes (Mathews 1999;

Roederer 1973: 71-82). A more detailed illustration of this relationship is shown in Figure 6.15, below.

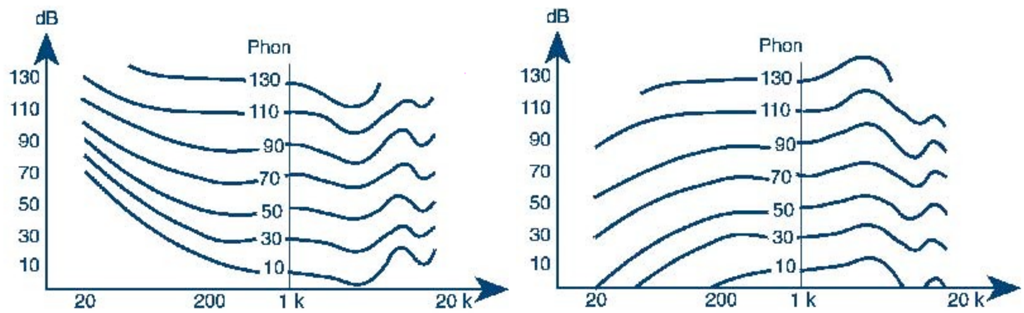


Figure 6.15 Fletcher Munson curves of equal loudness (left) and its inverse. Used for basic of frequency-dependent adjustment of amplitude to counterbalance this hearing non-linearity.[35]**=**

6.9.1 Data subset and sound rendering

The dataset for Experiment 4 is subset of a single metadata type (`TRADE`) of the complete dataset. This subset is used sequentially, day–at–a–time, second–at–a–time so the data can be extracted from the flat–file format in a single pass, thus not requiring the need for a separate, more sophisticated, persistence (database) model. This criterion would not change even if the technique were to be applied to a buffered real-time data feed. The subset extraction code is shown in Appendix 6.3.

The csound instrument used to render this new dataset to sound was designed for straightforward fine adjustment of the acoustic–psychoacoustic relationships. For the purpose of these experiments no graphical user–interface was implemented, though the one used in Experiment 2 could be easily adapted. Code 6.2 shows the shows the instrument itself and the head of the control (score) file, including the first 30 seconds of data to be rendered. Loading the instrument with the psychoacoustic adjustment tasks, rather than using an inflexible renderer and a pre–calculated score, affords rapid iterative adjustment of an essentially very basic instrument: the emphasis being on clarity of articulation rather than complexity of timbre.

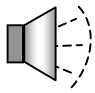
[35] Images From http://www.audioholics.com/education/acoustics-principles/physics-tutorial-2-the-physics-of-hearing/ were screen-scraped without permission and adjusted.

```
; instrument for playing score made using csnd.py.cSoundScore(adict)
sr = 44100
kr = 4410
ksmps = 10
nchnls = 1

instr 1 ; ----------- simple sine instrument with ADSR --------------
     ifreq     = 2048
     kampScale = 25      ; post 1stPB-adjust for max PB amp.
                         ;     val used = EOFoverallAmps=28148.7
     p5=p5/4             ; post 1stPB, push the freq spect up a M3
     kmaxamp   = kampScale *(1200 - (1100*p5/ifreq))   ; p5 =
frequency
     p3        = (p3*0.9) + (0.1*ifreq/p5)        ; p3 = dur
     idur         = 1/p3                          ; oscil freq = 1/dur
     kenv linseg 0, 0.01,1, P3-.01, 0
                  ; alternate for softer attack use f2=1st 1/2 sine
                  ; kenv      oscil 1, idur, 2
     asig oscili kenv, p5, 1   ; sine
     out asig * kmaxamp
endin

; scorefile for rendering sumVAT001. sums$value@time. made with
sumVAT.py

; choose one tempo (in BPM)
;t   0    60   ; 1  x time compression: 5 sec = 1 scond
;t   0    720  ; 12 x time compression: 5 sec = 1 minute
t    0    1800 ; 30 x time compression: 2 sec = 1 minute
;t   0    3600 ; 60 x time compression: 1 sec = 1 minute
f    1    0    16384     10    1              ; sinw
f    2    0    16384     9     .5    1      0 ; ½ sine 4soft envelope
i    1    0    1.0 1.0   633.431607803
i    1    7    1.0 1.0   7874.78575632
i    1    8    1.0 1.0   8162.64188795
i    1    10   1.0 1.0   8162.64188614
i    1    13   1.0 1.0   8075.19729843
i    1    16   1.0 1.0   8104.24089335
i    1    17   1.0 1.0   8104.24088033
i    1    18   1.0 1.0   8162.64187037
i    1    19   1.0 1.0   7988.6893679
i    1    20   1.0 1.0   8133.38893143
i    1    23   1.0 1.0   8046.25763285
i    1    24   1.0 1.0   8162.64184986
i    1    25   1.0 1.0   8162.64184804
i    1    26   1.0 1.0   8133.38890612
i    1    27   1.0 1.0   8162.64184414
i    1    29   1.0 1.0   8075.19713194
e
```

Code 6.2 csound instrument and first 30 seconds of the scorefile for rendering a $value TRADEs for Experiment 4.

AudioExamples10, 11 and 12 are each of 30 seconds duration with the same data (from Market Open) but with different temporal compressions, as indicated in Audio 6.4. AudioExample13 is a rendering of the entire day's TRADEs at a ratio of 1: 60. That is, 1 second represents 1 minute of TRADEs.

- AudioExample10. 30 secs. Compression ratio: 1: 1 (realtime)
- AudioExample11 30 secs. cRatio : 1:12
- AudioExample12 30 secs. cRatio: 1:30
- AudioExample13 Entire Day Compression ratio: 1:60(1 sec=1min)



Audio 6.4 Audio examples of the $value sum of the total market, moment–to–moment. Durations as time compression ratios are indicated.

6.9.2 Experiment 4: observations

AudioExamples10, 11 and 12 are provided to contextualise AudioExample13, the rendering of a full day's data at a 1:60 compression. 1:60 was chosen because (a) it is under the temporal 'fission boundary' for all frequencies of human hearing (Bregman 1994: 58–73) and (b) it is easy to identify the temporal occurrence of characteristics by watching a clock: one minute of audio is equivalent to one hour of trading. A trading day is six hours (10am–4pm), so one day's trading data can be listened to in six minutes.[36].

Described informationally, AudioExample13 presents a fluctuating, almost–continuous stream of small $value TRADEs (high frequencies) interspersed occasionally with those of larger $value. There are five high $value TRADEs, at the beginning of the day spaced five minutes apart. At the end, there are two, after a few seconds silence, the first of which is very $value large.

Each of the five large $value indications occur on Market Open. That is, the market is sequentially opened for trading in five groups, five minutes apart.[37] The initial TRADEs occurring simultaneously on a group's opening are a combination of those TRADEs identified in the orderbooks as Market–

[36] Such a compression rate is likely to offer a significant advance on the ability to represent individual datapoints on one screen of a visual display. single screen 6 x 60 x 60 = 21,600 pixels. For a monitor with a horizontal dimension of 12 inch, this translates to the need for 1800 pixels/inch, while the resolution of the eye is approx. 1200.

[37] This 'orderly open' approach is to mitigate against the risk of a trading engine crash when opening in a volatile environment. On the ASX, while an individual security always belongs to the same group, the order of opening is randomised in an attempt to suppress the exploitation of this 'feature'.

On–Open TRADEs and those that are initiated by brokers' trading software 'triggering stops' in the pre–market–open price–discovery phase.

Following Market Close, a 'price–determination' process is initiated in which the closing prices of all securities are calculated. Of the two large $value indicators that occur at the end of the day, the first represents the value of those TRADEs marked as <Market–On–Close>. In addition to the trades that occur 'on-market', brokers are permitted to trade between themselves without going through the exchange's trading engine, with the requirement that such trades must be registered through the engine within a short period of the market closing. The last $value indication is of the registration of those off-market trades.

While there was an expectation of an exponentially greater proportion of lower $value TRADEs than higher, the proportion of low–valued trades seemed out of proportion even to those expectations. An investigation of the reasons for this phenomenon initiated Experiment 5.

6.10 Experiment: Sonification of market volatility

An initial attempt was made to simultaneously sonify the TRADE data of individual securities by using the $value of each security's TRADE data when it occurred. This resulted in each TRADE contributing a component frequency to the overall spectrum comprised of all such TRADEs for that moment in time[38]. AudioExample14 presents the results. AudioExample15, applies the same techniques, except that all trades lower than $2000 value were filtered out, which accounts for the lower pitch.

- AudioExample14 Entire Day cRatio: 1:60 (1 sec=1min)
- AudioExample15 Entire Day cRatio: 1:60 (filter $value <$2000)

Audio 6.5 Audio examples of the use of simultaneous $value TRADEs to contribute to the composition of an evolving spectrum.

As for Experiment 4, the results were difficult to interpret, and appeared unsatisfactory for the same reasons. What was needed was a more detailed understanding of the structure of an order becoming a TRADE. To undertake

[38] These preliminary experiments, as exemplified by AudioExamples 14 and 15, were actually conducted prior to the commencement of Experiment 4, but they are reported here as they contributed to the results of Experiment 5.

the analysis to effect that, the data had to be made accessible in a manner appropriate for the analysis intended. It was not clear whether or not such analysis would have to include data other than TRADEs and so it was decided to store all the records in a single database. The first approach was to use a relational model[39] with a local *mySQL* server accessed via *pySQL*, the *Python* wrapper API. However process became difficult to manage as search times were unacceptable. The standard technique for this kind of problem in relational-database management is to create additional tables of indices (key, pointers) to reduce the query–response time (Coronel 2000: 89). These additional tables increase the size of the database as a whole. The technique was applied progressively in order to optimise the query–time/database–size configuration, but the model could not deliver query–times low enough to permit real-time rendering of the 1:60 time compression required. Perhaps, with continual refinement, a workable solution may have been found but the relational approach was eventually abandoned and an object-oriented b-tree structure was adopted, using HDF5[40] through the *pyTables* API. Code for translating the data to HDF5 is listed in Appendix 4.4.

A TRADE order type contains a number of fields[41]. Using the now more–easily–accessed data, analysis revealed long sequences of the following behaviour (in multiple securities simultaneously): Many small–$value TRADEs were initiated as Market Orders by a single trader through the same broker, either for accumulating or distributing holdings of security at the BID or ASK price. A plausible explanation for this pattern is that by so 'hitting the bid' or 'ask', these TRADEs remain relatively undetected by all except those with realtime access to the trading engine data[42]. When another trader entered the market 'significantly' on the same side, or when the price, or activity in the orderbook changed significantly, the sequence was often broken, sometimes to resume, sometimes not. Figure 6.16 graphically illustrates the

---

39 See Chapter 4 §4.3.2.2 for a discussion of the pro's and con's of different database models..

40 Hierarchical Data Format library version 5, (HDF5) is a versatile, mature scientific software library designed at NCSA supercomputing facility for the fast, flexible storage of enormous amounts of data. It provides a robust way to store data, organized by name in a tree-like fashion. With HDF5, extremely large datasets (hundreds of gigabytes in size) are organized in a filesystem-like hierarchy using containers called "groups" an accessed using the traditional POSIX /path/to/resource syntax.

41 see Appendix 5.3 for a description of the metadata structure of the trading–engine data.

42 Because, being 'At Market' TRADEs they can pass into and out of the order book almost instantaneously, without significantly shifting the BID or ASK price. An auditory alert would be an ideal way of drawing attention to such events.

structure through which the TRADEs were traced in order to arrive at this explanation. The activity described has the characteristics of computer-based algorithmic trading; in small amounts, presumably by broking firms that are not paying for each individual TRADE, in an attempt to avoid detection by the 'suppliers' of the other side of the TRADE; such knowledge in some situations, being sufficient to trigger a repositioning of their TRADEs in the market to their better advantage.

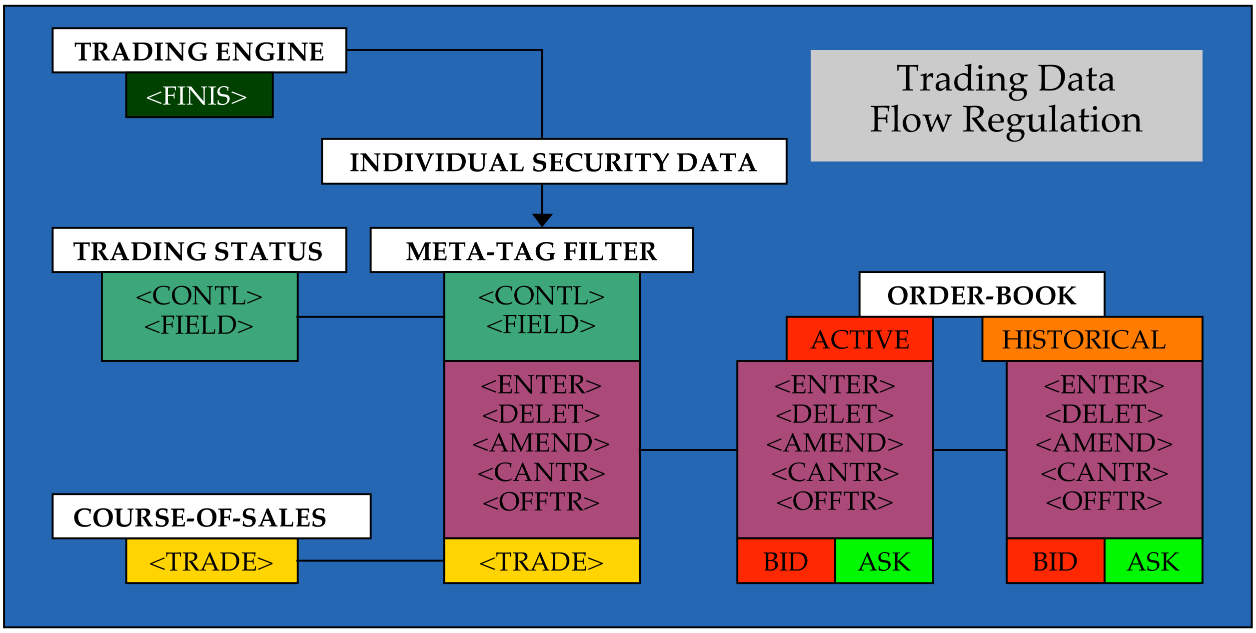


Figure 6.16 A graphic illustration of part of the HDF5 file structure used to trace the movent ot TRADE orders.

In order to provide a better representation of the way prices were fluctuating this small TRADE 'dispersion' characteristic of the dataset needed to be contained, or managed–whether or not the explanation just given is accurate or not, so an accumulation filter was applied, as follows. All trades in a security at the same price were accumulated in a buffer and the accumulated $value was sonified only when the price changed. Thus, this sonification became, in effect a monitor of market volatility: If the price of a security didn't change, it would not be sonified. It was necessary to implement inter–day buffers to accommodate this type of situation. Figure 6.17 graphically illustrates this process.

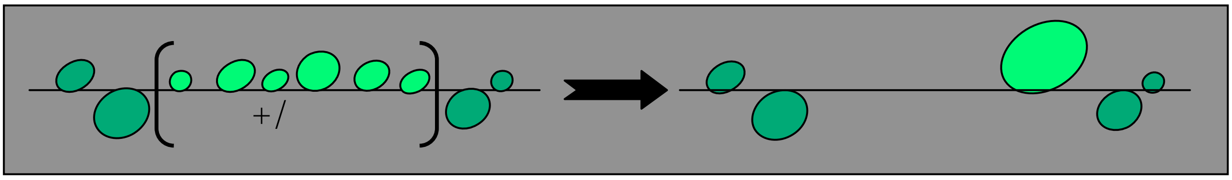
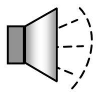


Figure 6.17 A graphic representation of the filter applied to TRADE data for the Experiment 5 sonifications. $value TRADEs in a security are accumulated until the price changes, at which point the accumulated value is sonified. On the LHS, The dark-green circles represent trades sonified without accumulation because price has changed beforehand. The smaller light-green circles represent TRADEs that are accumulating (+/). The RHS illustrates the overall result.

In order to demonstrate distribution of $value of 'Trades'[43] in using the accumulation technique just outlines, AudioExample16, 17 and 18 demonstrate the rendered results, with various setting of the filter. Figure 6.18 is a graphic representation of the relationship between them, including that the pitch gamut of AudioExamples17 and 18 are adjusted to be the same as AudioExample16, thus enabling a direct comparison of the distribution of $value in each of the ranges.

- AudioExample16 All trades (No filter)(full day) cRatio: 1:60
- AudioExample17 All trades of $value>$10K (30min) cRatio: 1:60
- AudioExample18 All trades of $value>$50K (30min cRatio: 1:60



Audio 6.6 Three examples of cumulative TRADE data, with $value filters, as shown.

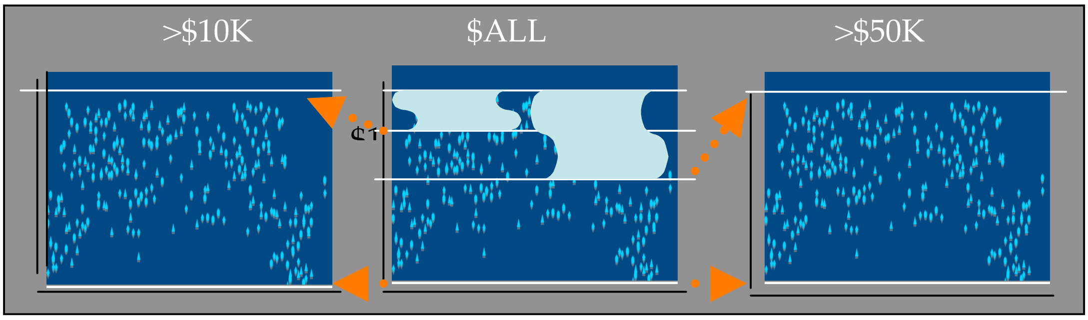


Figure 6.18 A graphic representation of the filtering of cumulative TRADEs below $10K and $50K and rescaling the results to the same pitch gamut before rendering.

[43] As the accumulation of TRADEs is not a TRADE , the term 'Trade' is used.

6.10.1 Experiment 5: observations

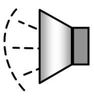
The opening is clearly audible, though because of the accumulation technique employed, the five Market On Open events are less pronounced. The middle of the day clearly has less activity – lunchtime perhaps – and the increased activity and volatility towards the end of the trading day is quite noticeable, especially in the last fifteen or twenty minutes of trading. On some days there are two Market On Close events, as discussed, however on the day presented in this example, there is only one. There is still a preponderance of smaller–$value Trades, but they appear more realistically weighted than in the previous experiments. The volatility of the market can be easily inferred from this mapping, and overall, the technique seems to work well.

6.11 Experiment: Separation of conflicting conceptual mappings

One further experiment was undertaken with the cumulative mapping technique employed in Experiment 5. A sonification event only occurs when the price changes so, to each event a second pitch was appended to indicate whether the TRADE that triggered the sonification event was a downward or upward movement in <price>. The pitch of that tone was made *higher* when the <price> increased and *lower* when it decreased; the size of the interval between the two tones indicating the extent of the price difference.

AudioExamples 19, 20 and 21 demonstrate the technique.

- AudioExample19 First hour of trading cRat: 1:60
- AudioExample20 Middle of Day(12h30-13h30) cRat: 1:60
- AudioExample21 Last hour of trading cRat: 1:60



Audio 6.7 Three examples of cumulative TRADE data appended be a shift in pitch to indicate whether the <price> of the following trade had increased or decreased and to what extent. A higher pitch indicates that the price rose.

6.11.1 Experiment 6: observations

There appears to be no difficulty in cognitively separating the two opposing mapping paradigms when they are superimposed. Having had the concepts explained to them, several people were asked informally where or not they could describe whether a specific event was higher or lower in $value than the previous and whether the TRADE following was higher or lower in pricee. None appeared to have any difficulty in separating the two paradigms. In fact, when it was pointed out that the two superimposed conceptual mappings were opposed psychoacoustically, most of those tested expressed surprise, indicating that they hadn't noticed.

This informal experiment illustrates that it is possible for superimposed concepts to be easily separated cognitively even when mapping of the two paradigms into the same psychoacoustic space is opposed: intention, that is information, can take precedence over perception and sensation when given the conditions to do so.

6.12 Summary

A number of techniques were explored in the experiments reported in this chapter. A new technique for the audification of securities market data by sonifying net market returns seemed to preserve the autocorrelation in the data that could be aurally detected. A lack of sufficient data to provide for extended listening was compensated for by the use of a frequency-modulation homomorphic mapping technique and the apparent distinctiveness of net returns from related datasets suggests that rigorous empirical testing of both techniques would be worthwhile.

The opportunity to work with a relatively large high-frequency data set was useful in testing the application of the *SoniPy* framework paradigm in the domain that traditional music composition software is the weakest; namely, the handling of external multivariate datasets. While the tools and techniques for handling large multiplexed datasets is not uniquely a sonification problem, the special skills that it requires need to be acquired for such sonifications to be possible. These experiments served to emphasise the flexibility of using an interpreted language when searching for solution rather than just coding a well-defined algorithm. The extensive collection of data handling tools *SoniPy* provided access to worked extremely well, both individually and in tandem. In fact, the exploratory techniques needed to clean, and then search and manipulate such a dataset would have been prohibitively arduous without them.

In the $value experiments, the conceptual mapping techniques employed using high levels of psychoacoustic redundancy seemed to work well, especially those sonifying market volatility. The lack of cognitive dissonance when two opposing conceptual paradigms were superimposed was a clear indication of the power of intention, that is, a seeking of information, over perception and sensation, given conducive conditions.

Chapter 7

EPILOGUE

Hamlet:	Do you see yonder cloud that's almost in shape of a camel?
Polonius:	By th' mass, and 'tis like a camel indeed.
Hamlet:	Methinks it is like a weasel.
Polonius:	It is back'd like a weasel.
Hamlet:	Or like a whale.
Polonius:	Very like a whale.
Hamlet:	… They fool me to the top of my bent…

(Shakespeare: *Hamlet* Act III, Scene 2).

7.1 Chapter 2

The whole field of sonification research was reviewed in Chapter 2. Because the study was not period-limited, keeping pace with a number of developing topics proved daunting at times, but was mitigated somewhat by their currently being relatively few avenues of publication; *ICAD*, *Human Factors* and *HCI* being the principal ones. This has both positive and negative effects. The negative effects are the tendency for a little 'inbreeding', particularly in citations. The positive effects include that most researchers in the auditory display community have easier access to a larger body of research in the field than they otherwise might. While it is a sign of widening community acceptance of the discipline when more papers are accepted elsewhere, the institution of a Journal, which seems to be on the horizon, will be welcomed as it seems other journals now consider auditory display as a defined territory.

The ICAD website, which has recently been upgraded, performed a useful function in this regard too, although not all the papers that are listed are available. Papers from ICAD '94 are available as PDF format documents, however the contents are not accessible to search engines as each page within each document is in image format. A scan and OCR produced enough of a workable copy for personal use but also revealed that some fine papers have been all but ignored by subsequent researchers.

An attempt at too-taxonomic a classification of sonification techniques was a dilemma until it became clear that there is not much to be gained from trying to make fine, often somewhat arbitrary distinctions in a field that is so fecund with possibilities. When reviewing so much material, perhaps it is part of the process to become as taken with possibilities as astounded by the depth of insight of those who build the foundations, but that occurred in this instance. There has hardly been a week pass without a new revelation arising from those two books from 1994: Kramer et al. and Bregman. Having the opportunity to write a review of the field for a book was an added stimulus and the sonification literature review for this thesis probably would not have been as broad otherwise.

In reviewing a number of other theses in the field it became clear that there was little to be gained by including a cursory overview of the physics or psychophysics of sound in this thesis. Another was the ability to reference personal material previously published that covers the topic more fully[1]. However, the most important reason was a sense that the discussion needed to move on. While physics or psychophysics is important from an analytic perspective, for it to be useful for synthesis it needs to be in the form of inverse filters, such as that for Fletcher-Munson described in Chapter 6 (§6.9). There is some peripheral work currently being undertaken in this area (Cabrera Ferguson and Schubert 2007) and it would be useful if were to be generalised. The concern of this thesis, however, was to look further forward, to try to find a theoretical basis for the better mental instantiations of multivariate datasets using sonification.

At the outset it was unclear how to best confine the discussion of sonification in such a way as to avoid having to deal with those now broad applications of the term to indicate a 'datasound' oriented media arts practice, in which the primary motivations are socio-cultural. Often such work is not concerned with informational relations within the data–and is sometimes specifically interested in obscuring it–yet in some compositions the concern for aural clarity in such matters *is* important. Coining a new term, *soniculation*, was useful in clarifying the different motivations. In any event, given the importance of (the action of) *intent* in defining information in a particular context, as discussed in chapters 3 and 4, the need for a nominal definition of sonification may be misplaced.

---

1 http://worrall.avatar.com.au/courses/PPofM/

7.2 Chapter 3

Chapter 3 was a difficult chapter to write, given that it finally covering two–and–a–half millennia of Western thought. It was as good to become reacquainted with Kant, as it was to discover the 'back-story' of Brentano and his mentors, Husserl, and Peirce's less–well–known work in phenomenology. Music's closest association with that philosophical movement is through the work of Pierre Schaeffer (1966), which, despite its importance in the history of Western music in the twentieth century, is eclectic enough to have resisted being translated from the original French.

So many different threads emanate from or react to Kant, that Mozart of philosophers, and a constant problem was what to do with those in the nineteenth and early twentieth century; particularly those influenced by Martin Heidegger and William James. Goethe, Schopenhauer, Hegel and Deleuze were entertained for a while, but eventually not included. Much of the literature on consciousness and the theory of mind that deals with sensation and mental imagery is a 'mind field' for the casual reader 'arriving' in the middle of a discourse, as finer and finer distinctions result in new terminologies that are invoked seemingly without a backwards glance. David Chalmer's websites are very useful resources in this regard.

It has never been clear whether it is ignorance of arrogance that keeps many scientists from reading philosophy, but after reading the professionals it is hard to entertain the clowns, especially those who believe they have discovered a resolution to the mind/body dualism in the application of systems theory. In any event, the distinction between data and information, how they function with each other, is now quite clear; as is the understanding that abstract immanent phenomenal objects of the type produced in the mind of a listener in response to conventional parameter mapping sonifications are unlikely to ever work very effectively until embodied in ways that afford their mundane comprehension with a low cognitive load. In fact it was only on reading Husserl's work, in particular, that the Platonism of his aims become so clear, and the reason why his desire for a theory of pure phenomenal immanence continually seemed to collapse in that direction. If this inadequacy is a reality, it is probably dealt with more effectively if it is considered a feature of the human condition rather than as a 'bug', as discussed in the introduction (§1.1.4).

While much of the chapter oscillated back-and-forth as appendices, part of the introduction and the conclusion, the essential threads eventually became clearer and in the end it seems to work. The biggest regret is that there wasn't the opportunity or time to pursue the work of Michael Polanyi on tacit knowledge, as mentioned in Appendix 1. For those who wish to pursue the thread, the Robert Innis' book (1994) is a gem.

7.3 Chapter 4

In some ways, Chapter 4 was a questioning of the findings of Chapter 3 as a broad basis for the principles needed for Chapter 5. In practice, Chapter 4 remained as a sketch until the other two were almost complete. Currently, most work on embodiment seems to be being undertaken in the context of artistic performance that, though interesting, is of limited value for the work of this thesis without any resulting empirical biomechanical data. Sports science is probably a more appropriate experimental context, though building models that include untrained actions as well as those of movement artists seems like a possible path forward, perhaps collaboratively with animators and the support of the film industry. There are quite a few possibilities and if this thesis was building on the back of what it eventually became, that is the direction it would take.

7.4 Chapter 5

The *SoniPy* chapter and the interim publications that came out of that work was the result of a lot of detailed, sometimes pleasant, work on the discovery of material in the public domain that was well documented, easy to install and did what it said it could. Equally invigorating, but negatively so, was the frustration of being caught rewriting code that seems to have been placed in the public domain for the express purpose of claiming credit for the work of others who happen to come across it. The design of some reasonably rigid selection criteria soon followed! There are good reasons why *Python* has become such a popular language in which to think. It is elegant and powerful and very well supported. To go back to a specialist computer music language after experiencing the breadth and depth of the tools that are available for the asking is not even worth entertaining.

SoniPy is clearly not a completed software application, nor even a mature toolkit. How quickly, and if, it moves past the current 'proof–of–concept' stage will depend on community response, of which there has been some. From a research perspective, there is a clear need for the seamless integration of a controlled experimental environment for empirical testing, including the addition of tools for Internet–based experiments. The components are there, they just need to be integrated.

7.5 Chapter 6

The early experiments in sonifying securities data and the discovery of the simplicity of the discovery of the Net Returns approach to transforming the data into an oscillation was a positive beginning. It may be a point of conjecture how often one human being can listen to countless 600 millisecond examples of various types of noise before insanity sets in, but to some, such as this writer, it never ceased to be enthralling! It was surprising to find so little work has been undertaken to empirically test the differentiability of sonified stochastic functions, despite them having been used so extensively in the computer music community.

If the 'joy of *Python*' was ever going to be tested, it was during the weeks spent using it to clean the high-frequency trading engine data. Both parties survived, however, and the tools developed for *mySQL* and *HDF5* databases permitted searching that would have been extremely difficult using a contiguous flat-file format. In the beginning, it was a mystery why the $value sonifications were so heavily loaded towards small trades. Tracing the probable cause and developing and implementing a sonification scenario that *soniculated* the volatility model was, in the end, quite successful and satisfying.

From the beginning of work for this thesis, a decision was made to keep the sound synthesis models as simple as possible. Perhaps it is perverse, but the structures always appear more lucid that way. And partly as a result, it is clearer now than ever before that the way forward for producing more powerful multivariate sonifications is likely to lie in the incorporation, along with better psychoacoustic models, of subtle queues to an embodied mind; perhaps through coherently resonant physical simulations that afford attention, or perhaps through models that modulate the vestibules and proprioceptive centres of the body, but it is unlikely to be through the application of more synthetic reverberation to disembodied multidimensional sample-realised abstractions. Some well-formed synthetic physical models of musical instruments do exist, however it was not clear at this stage whether they will be adequate without the inclusion of their other halves–their players. There are many examples of timbrally-challenged synthetic resonators that seem to function perfectly adequately for *Gebrauchsmusik* and if the postulation that embodiment is to be an important and workable design feature of sonifications in the future, it seems unlikely that it will matter whether or not they will succeed simply as beautiful resonators.

The floating ephemeral nature of immanent objects means that they are susceptible to being 'grounded' by ideologies, and the current Cartesian, timbre-fetish orientation of much computer music software will need to embrace a sophisticated sense of embodiment if it is to be of real use in multivariate data sonification. It is not too hard to imagine that the necessity that is driving this may, in turn, be a lasting legacy to music composition.

Appendix 1

KNOWLEDGE: TYPES AND METHODS OF ACQUIRING

A 1.1 Introduction

There is a distinction is made between the terms *information* and *knowledge*, despite them being frequently used to mean the same thing in common parlance[1]. Knowledge is considered to be *a theoretical or practical understanding of a subject* (AOED); a coherent understanding of a way accepted facts about the subject relate to each other[2], whilst *information* is used to mean items of communicable knowledge (ideas). *Knowledge* is considered to be *a theoretical or practical understanding of a subject* (ACOD); a coherent understanding of a way accepted facts about the subject relate to each other. Accepted facts contribute to *truth,* that can be understood, after Fromm (1947: 238), to mean a functional approximation to reality; itself understood to be relative to an accepted body of knowledge. Descriptively, the term *knowledge* implies that its content is meaningful, 'acknowledged', and veridical. Strictly, an individual *S* knows a proposition *P* only if *S* believes *P* and *P* is true. If P is false, *S* does not know *P*, even though *S* might know that they *believe* that they know *P*. However, Truth is currently acknowledged to have multiple, and developing, ontologies.

This appendix outlines the major methods of acquiring knowledge. It does not account for the *reasons* an individual, or community, or culture use one method, or mixture of methods, rather than another, in a particular circumstance. Also, whilst it is not an historical account of the *getting* of knowledge, the methods are presented in an order which reflects the occidental procession from a reliance on authority and revelation as the principle source, stimulated by occasional revelatory experiences, to individual discovery that is verifiable by others, whether through intuition or more formally, through inference.

---

1 Especially in English, where the Anglo-Saxon distinction between *witan* ('wit') and *cnawan* ('to know') barely survives. The distinction remains, in German, for example, with *wissen*, *kennen*, *erkennen* and in French, with *connaître* and *savoir*. Seen Chapter 3 Footnote 17 which relates 'wit' to the Greek *eide*.

2 Accepted facts contribute to 'truth' that can be understood, after Fromm (1947: 238), to mean a functional approximation to reality; itself understood to be relative to an accepted body of knowledge.

In an age when many thinkers were still wrestling for intellectual[3] independence from the dominance of the church in matter of knowledge, understanding and wisdom, the eighteenth century philosopher Immanuel Kant considered understanding to arise rationally:

> Understanding may be regarded as a faculty which secures the unity of appearances by means of rules and reason as being the faculty that secures the unity of the rules of understanding under principles. Accordingly, reason never applies itself directly to experience or to any object, but to understanding, in order to give to the manifold of knowledge of the latter an *a priori* unity by means of concepts, a unity of which may be called the unity of reason, and which is quite different in kind from any unity that can be accomplished by the understanding (Kant 1787/1929: 303).

Contemporary methods in scientific research, and particularly if they involve interdisciplinary practices, are usually an evolving mixture, and Reliabilism is seen to be a reasonable account of the methodology many contemporary practices, including many that are described by their users as empirical.

A.1.1.1 The logic and psychology of knowledge

There is a frequently confused epistemological distinction between the *logic* of knowledge that is concerned with logical relations, and the *psychology* of knowledge that deals with empirical discoveries, about which Immanuel Kant was clear:

> It is of the utmost importance to isolate the various modes of knowledge according as they differ in kind and in origin, and to secure that they be not confounded owing to the fact that usually, in our employment of them, they are combined. … It must be admitted, however, that the two elements of our knowledge -- that which is in our power completely a priori, and that which is obtainable only *a posteriori* from experience -- have never been very clearly distinguished, not even by professional thinkers and that they have therefore failed to bring about the delimitation of a special kind of knowledge, and thereby the true idea of the science which has preoccupied human reason so long and so greatly (Kant 1787/1929: 660). [4]

Called psychologism, Karl Popper made the same point, more than a century-and-a-half later:

> The initial stage, the act of conceiving or inventing a theory, seems to me neither to call for logical analysis nor to be susceptible of it. The question how it happens that a new idea occurs to a man … may be of great interest to empirical

---

[3] The noun use of the term *intellectual* for persons arose at this time (ACOD).

[4] By *professional thinkers* Kant is referring to Lock and Hume, neither of who make the distinction.

> psychology; but it is irrelevant to the logical analysis of scientific knowledge (Popper 1959/1972: 31).

Different methods of acquiring knowledge provide access to different *types* of knowledge and need not be employed in mutual exclusion. For example, empirical science, as practiced, increasingly relies on authority in the form of peer review as well as observation, and intuition can play a large part in deciding which possible consequences to use inductively. From a deductive perspective, abduction is a logical fallacy, yet it has proved to be a successful method in artificially intelligent agents' interaction within their environments and is now considered by some to be at the root of human perception and cognition.

A 1.1.2 Explicit and implicit knowledge

There is a further epistemological distinction between explicit and implicit knowledge; between *knowing-that* (2+2 = 4; Tuesday follows Monday etc) and *knowing-how* (to play the violin; ride a bicycle etc). The distinction can found in the Greek distinction between *episteme* (theoretical truth) and *tekhne* (practical methods for effecting results). By the time the Latin Scholastics of the Middle Ages had rediscovered the Greek language and the philosophy of Aristotle, the term 'science' had come to imply both, differentiated as the theoretical sciences[5] (philosophy) and the manual arts[6] (practice).

Explicit knowledge is also known as *declarative*, *descriptive* or *propositional* knowledge. Such knowledge is usually acquired reflectively by logical reasoning, mathematical proof and scientific methods, or by reference to historical or cultural practices. By the time Descartes formulated his theory of the separation of the mind or soul (*res cogitans*) and the 'extended' world outside it (*res extensa*), now known as Cartesian Dualism, the mind was clearly thought of as the seat of reason, God and the sacred, while the body was a fleshy machine and clearly inferior, or at least secondary. This attitude began to change slowly through a philosophical 'bottom up' exploration of

---

5 In its oldest sense (c.1300), 'science' meant knowledge (of something) acquired by study, also a particular branch of knowledge, from O.Fr. science, from L. scientia knowledge, from sciens (gen. scientis), prp. of scire to know, probably originally to separate one thing from another... Main modern (restricted) sense of body of regular or methodical observations or propositions ... concerning any subject or speculation is attested from 1725; in 17c.-18c.; this concept commonly was called philosophy. (OLED).

6 *ars*, from the Greek *artios* 'a complete skill as a result of learning or practice'.

the relationship between sensation and knowledge of the world, culminating in Kant's *transcendental idealism* as a way of resolving the idealist's dilemma of how true knowledge of the world was possible with the obvious success of empirical methods.

In an historically common 'spectator view' of knowledge, human experience is primarily a matter of contemplation. The American humanist philosopher Corliss Lamont suggests that this position is largely derived from an overemphasis on the role of vision. Further, he emphasizes the location of truth with respect to such knowledge:

> [W]orkability is the *test* of a truth, not the *source* of it. The truth of an idea does not lie *in* verification; we are able to prove it true *through* verification. An idea is true *if* it works, not because it works; for it already *was* true and corresponding to objective reality. New truths lie all about us waiting to be discovered by persons wielding scientific techniques; but the process of discovering does not *make* ideas true (Lamont 1949/1997: 243).

The polymath Michael Polanyi thought that non-explicit *know-how* type knowledge is 'tacit' in that it is *embodied* and cannot be fully described in words. Attempts to perform such analyses are often laborious, difficult and 'destructive' (Polanyi 1966). The fact that we know more than we can clearly articulate contributes to the conclusion that much knowledge is passed on tacitly by practical means. In such an operational approach to knowledge, truth is functional; an idea or hypothesis is true if and while it works.

A 1.2 Methods of acquiring knowledge

Knowledge acquisition, the process of gaining knowledge from information, can be understood as the integration of new information into that which is already coherently embodied. Considered in this way, the trans*form*ation of in*form*ation into knowledge is an internal process–whether to an individual, a group or a community, and while there may be sonification techniques to enhance that process[7], they lie outside scope of the current thesis. What follows is a pragmatic summary of different modes of knowledge acquisition. With the exception of Reliablism, so apt an epistemological description of current scholarly practice, most of its contents are widely understood, especially from an individual point–of–view.

---

7 Such as those used to enhance learning by entraining the brain's beta frequencies.

A 1.2.1 Authority

Leaders, considered knowledgeable and wise, decide what is true for everyone, sometimes during periods of special inspiration, insight or perhaps revelation. The surrender to authority is common in most arenas of human endeavour. It acts to stabilise and encapsulate a corpus of traditional knowledge against which new ideas can be tested.

A 1.2.2 Revelation

Revelation is a method is often when seers and prophets employ magic and divination techniques. What is usually assumed is that the practitioners have ability to access knowledge through inspired communion with supernatural being(s); such access usually requiring mindful techniques such as faith (belief) sometimes induced by body renunciation techniques.

A 1.2.3 Intuition

Intuition is the direct, immediate and certain apprehension of truths without the intervention of conscious reasoning or related sensory perceptions. The difference between intuition and a revelation is the person doing the intuiting does not assume that the source of the knowledge is external to workings of their own brain.

A 1.2.4 Heuristics, folklore and commonsense

The generalizations we apply in everyday life in predicting and explaining each other's behavior, often collectively referred to as *folk psychology*, is both remarkably successful and indispensable. A person's 'personal knowledge', what they believes doubt, desire, fear, etc. is a highly reliable indicator of what they will do, and we have no other way of making sense of each other's behavior than by ascribing such states and applying the relevant generalizations. This theory of knowledge is also known as *intentional realism*[8], which recognizes that we all, in some way, committed to the basic truth of commonsense psychology and, hence, to the existence of the states its generalizations refer to (Dretske 2000). Some, such as Fodor, also hold that commonsense psychology will be vindicated by cognitive science, given that

8 For an extended description, See http://plato.stanford.edu/entries/ mental-representation/

propositional attitudes can be construed as computational relations to mental representations (Fodor 1987).

Churchland (1981) thinks that, as a theory of the mind, folk psychology has a long history of failure that can't be incorporated into the framework of modern scientific theories, including cognitive psychology. He argues that the states and representations folk psychology postulates simply don't exist; that it is comparable to alchemy and ought to suffer a comparable fate. On the other hand, Dennett (1987) seems prepared to admit that the generalizations of commonsense psychology are true and also indispensable, but denies that this is sufficient reason to believe in the entities to which they appear to refer. He supports this stance on that basis that there is nothing more to having a propositional attitude than to give an intentional explanation of a system's behavior by adopting an the *intentional stance* toward it. Assuming a system is is rational[9], if the strategy of assigning contentful states to it and predicting and explaining its behavior is successful, then the system is intentional and the generalised propositional attitudes we assign to it are true (Dennett 1987: 29.)

A 1.2.5 Inference

Inference is a term covering a number of forms of reasoning in which conclusions are drawn or judgments made on the basis of circumstantial evidence and prior conclusions rather than purely on the basis of direct observation or knowledge arrived at by direct observation. The conclusion may be correct, incorrect, partially correct, correct to within a certain degree of accuracy, or correct in certain situations. Five distinct inferential methods are recognised: deduction, induction, Bayesian inference (which is really a form of induction), abduction and reliability. Inferential methods are the principal methods of science, although not all types are undisputedly considered applicable to all fields of inquiry.

A 1.2.6 Deductive inference

A way of reasoning in which a collection of ideas is built into a coherent whole through the rigorous deductive application of certain axioms or

---

9 Something is rational if behaves in accordance with the truths and falsehoods afforded by its environment.

postulates using propositional, predicate, modal, and/or fuzzy logics (Lemmon 1965, Hunter 1971, McNeil and Freiberger 1993). The appeal to deductive reason (syllogism) as a source of knowledge or justification is known as rationalism. Strictly, rationalism submits neither the original propositions, which may be selected through intuition, nor final conclusions, to experimental verification. Two important philosophers of the European Age of Reason (seventeenth century) were Descartes and Leibniz, who, after Plato and Spinoza, considered knowledge of eternal truths, including the epistemological and metaphysical foundations of the sciences and mathematics, could be attained deductively, i.e. syllogising without recourse to inference from any sensory experience.

Summary
If *a* then *b*. [*b* is a consequence of the assumption of antecedent *a*].

A 1.2.6.1 Inductive inference

A form of reasoning that makes generalizations based on individual instances. It is used to ascribe properties or relations to types based on observed instances; to formulate laws based on the results of a limited number of experiments or the direct observations of recurring phenomenal patterns. In its rudimentary form, it is the process of learning through trial and error experience. In connection with the natural and social sciences, empiricism refers to the use of working hypotheses that are testable using observation or experiment. The doctrine of empiricism is discussed in more detail in §3.3.2.1.

Summary
- Simple: All observed *a* are *b*, therefore all *a* are *b*. (enumerative induction).
- Proportional: P(*g*), a percentage of known *g*'s in group *G*, have attribute *A* Individual *i* is another member of *G*, therefore there is a P(*i*), corresponding to P(*g*), that *i* has attribute *A*
- Analogic: *a* is similar to *b*. *a* has attribute *X*, therefore *b* has attribute *X*.

A 1.2.6.2 Bayesian inference

Bayesian inference is a type of inductive reasoning using the probability of dependence between events. If A and B, two random events, are *independent* of each other, then the probability (P) that they will appear jointly (i.e. together) is simply the product of the probabilities of each of them occurring: P (A and B) = P(A) . P(B). However, if C and D are *not independent* of each other, then the probability of the dependence can be expressed as a P(C given

D) and P(D given C), which need not be the same. In the eighteenth century, the British mathematician Thomas Bayes developed a theorem to compute the latter, given the former (see below).

<u>Summary</u>

If $P(b \mid a) = x$	(if the probability of *b* given *a* is *x*)
then $P(a \mid b) = x \cdot P(b) \; / \; P(a)$	(then the probability of *a* given *b* equals *x* multiplied by the probability of *b* divided by the probability of *a*)

A 1.2.6.3 Abductive inference

Abductive inference, or 'Inference to the Best Explanation', is a method of reasoning that infers the most likely explanations from data describing something. Abduction allows the precondition *a* of "*a* entails *b*" to be inferred from the consequence *b*. Deduction and abduction thus differ in the direction in which a rule like "*a* entails *b*" is used for inference. Abduction is formally equivalent to the logical fallacy affirming the consequent (*post hoc ergo propter hoc*), because there are multiple possible explanations for *b*. Unlike deduction and induction, abduction can produce results that are incorrect within its formal system. However, it can still be useful as a heuristic, especially when something is known about the likelihood of different causes for *b*.

The philosopher Charles Peirce introduced abduction into modern logic. In his works before 1900, he mostly uses the term to mean the use of a known rule to explain an observation, e.g., "If the bell rang, someone is at the door," or more accurately, "If the bell rang, then the most probable explanation is that it someone was at the door." Peirce changed his use of the term in later life to mean something similar to induction, though that need not concern us here.

From an abductive perspective, an explanation is valid if it is the best possible explanation of a set of known data. The 'best possible explanation' is often characterised, for scientific or technological purposes, for example, using Occam's razor, that is, in terms of simplicity and elegance. The philosopher Peter Lipton, in an attempt to avoid 'inference to the best explanation' being reduced to 'inference to the *likeliest* explanation' followed by probabilistic (Bayesian) analysis, suggests 'inference to the *loveliest* explanation'; the loveliest explanation to be the one that, if correct, provides the most understanding (Lipton 2004: 59). Perhaps originally, Lipton defends the thesis that Bayesian induction and 'inference to the best explanation' are broadly compatible because of their common concern for and explanationist approach (Lipton 2004: 106-107) and so attempts to build a bridge between the two by exploiting an overlap between the *loveliest* and the *likeliest* explanations.

Summary

If D is a collection of data (facts, observations, *a priori* assumptions) and
H is a collection of possible hypotheses (H_1, H_2, H_3, … H_n) for explaining D, then
the H_n that explains more D in the best, most elegantly is probably true.

A 1.2.7 Reliableness

Reliableness is a method for acquiring knowledge based on various belief-forming processes. It justifies the belief in the veridicity of perceptual sensations if the resulting perception is known to lead to a suitably high proportion of true beliefs. It is not a requirement of users of the method, or anyone else, know that the process is reliable or have any sort of knowledge of its reliability–all that is required is that it *is* in fact reliable. Thus, no appeal to sensory experience is required, thus effectively short-circuiting the issue that divides representationalism and phenomenalism. Reliabilism thus rejects the issue on which all three of the more traditional theories attempt to respond to: the issue of how sensory experience provides a reason for thinking that perceptual beliefs are true.[10] On the assumption that our perceptual processes are in fact reliable in the way that we take them to be, it offers a seemingly straightforward and account of how perceptual beliefs about physical objects and the physical world are justified.

Reliabilism emphasizes the properties of the processes used to arrive at truths. In reliabilist approaches to knowledge acquisition, noticing a static relationship between a conjecture and a body of evidence, knowing that an hypothesis does not contradict the evidence, or even is in-accord with it, for example, is insufficient to warrant support for it from the evidence; additional account must be taken of how reliable the method that produce the hypothesis is known to be in producing truthful hypotheses. Reliabilist thinking underpins the greater acceptance of the diagnostic judgements of

10 For an extended description, See http://plato.stanford.edu/entries/reliabilism/

experts over laypersons, the preferential support for research programs with fecund histories and the scorn of *ad hoc* hypotheses.

The first, then unrecognised as such, formulation of a reliability account of knowledge was in the mathematician Frank Ramesy's writing on knowledge (Ramsey 1931). Several similarly subjunctive theories, such as tracking theory and contexturalism, were developed in the latter part of that century, as discussed by Goldman, who notes

> Reliability theories of knowledge of varying stripes continue to appeal to many epistemologists, and permutations abound. … [Some theories] focus on modal reliability, on getting truth or avoiding error in possible worlds with specified relations to the actual one. They also focus on local reliability, that is, truth-acquisition or error avoidance in scenarios linked to the actual scenario in question Goldman (2008).

Reliabilism is still in active development and, according to Goldman, seems to have considerable robustness and flexibility.

<u>Summary</u>
S knows *P* if and only if *S* truly believes *P*, and *S*'s belief that *P* was produced by a reliable belief-forming process.

Appendix 2

Probability, statistics & time–series: basic principles[1]

A 4.1 Probability distributions & random variables

A 2.1.2 Random variables

A *sample point* is a point that is associated with each outcome of an experiment.

A *sample space S* is the totality of points $s_i (i = 1,2,...,n)$ that correspond to all possible outcomes of an experiment.

Finite sample spaces contain a countable number of sample points.

A *countably,* or *denumerable*, *infinite sample space* contains an infinite number of sample points that can be placed in one–to–one correspondence with the set of natural numbers $\mathbb{N}$. ($\mathbb{N}$ ={ 1, 2, 3 …})

A *discrete sample space* is a space that contains a finite or countably-infinite number of sample points.

A *continuous sample space* is a space that contains sample points that form a continuum.

A *random variable* is a function $x(s)$ whose value is defined at each sample point $s_1, s_2, s_3, ..., s_n$ in a sample space *S*.

A 2.1.3 Probability density functions

If *x* represents a continuous random variable in a sample space *S*, the probability density function (or simply a probability density), *p(x)* is a function that satisfies the following conditions:

(1) $p(x) \geq 0$, for all $x \varepsilon S$. [The probability of every sample in S is ≥ 0.]

(2) $\int_S P(x)dx = 1$ [These probabilities all sum to 1.]

(3) For any $x_1 \leq x_2$ in *S*, the probability of *x* lying in the range $x_1 \leq x \leq x_2$ is

[1] While no originality is claimed over these brief notes, no identifiable individual sources are acknowledged. They are informally compiled here for the benefit of those reading Chapter 5 who may not be familiar with the underlying statistical principles. One of the difficulties in reading mathematics is the pronunciation of unfamiliar symbols. The intention behind the inclusion of the Greek alphabet in the footer is to assist the reader in that process.

$$\int_{x_1}^{x_2} p(x)\,dx = P(x_1 \le x \le x_2).$$

Note that $P(x_1 \le x \le x_2)$ represents the area under the graph $y = p(x)$ in the xy-plane between two arbitrary points x_1 and x_2 as illustrated in Figure A2.1.

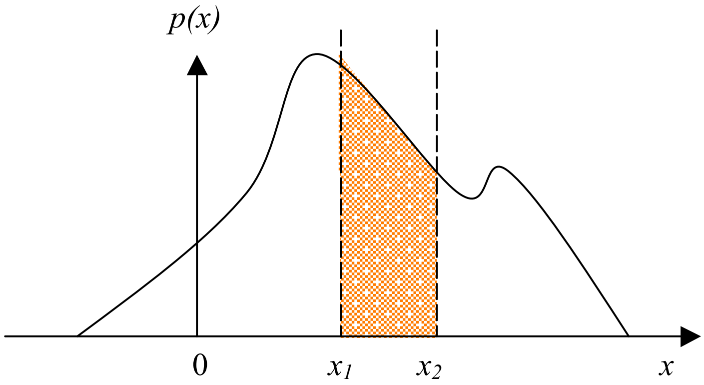


Figure A2.1. The probability of x being between x_1 and x_2 is the area under the curve of the probability density function *p(x)*.

The cumulative probability density function, or 'probability distribution function, or just the 'distribution function' (CDF for short), is a function which completely describes of a real-values random variable, a function When the random variable takes values in the set of real numbers, the probability distribution is completely described by the cumulative distribution function, whose value at each real x is the probability that the random variable is smaller than or equal to x.

A 2.2 Central tendency

The term *central tendency* refers to the way values in a distribution cluster. three common measures of central tendency are the

- **mode**, or the value occurring most often. Distributions with two or more different such points are called *bimodal* or *multimodal*.
- **median**, or middle value in an ordered list of them, and
- **average** or **arithmetic mean**, is the sum of all the values divided by how many of them there are. The arithmetic mean, $\bar{x}$, of a data distribution of n scores is defined as

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

A 2.3 Moments of dispersion about the statistical mean

The statistical moments of a distribution about this mean, also called *central moments*, are all defined by

$$\mu'_n = \sum_{i=1}^{n} (x_i - \alpha)^n P(x)$$

where n is the order of the moment, α is the value around which the moment is taken and *P(x)* is the probability function. Moments can be thought of as forces. Torque, the twisting force, for example, is described as the turning-moment. Other physical examples are the moment of inertia, angular momentum and magnetic moment, which is a measure of the strength and direction of a magnetic source. The moments about the mean of a probability density function (see § later) are called central moments and they describe "forces" that shape the function, independent of translation–shifting its location in the x-y plane, for example.

A 2.3.1 The 1st central moment (μ_1 : 0)

The first moment about the mean is the sum of the deviations of the scores from the mean, divided by the number of number of scores. The mean is the "balancing point" of the sample scores so the first moment is always 0.

$$\mu_1 = \frac{1}{n} \sum_{i=1}^{n} \left(x_i - \bar{x} \right)$$

Note that the convention is to use the expression $(x_i - \bar{x})$ not $(\bar{x} - x_i)$, so that all positive resultants are above the mean.

A 2.3.2 The 2nd central moment (μ_2, s^2 or σ^2 : the variance)

The second moment about the mean is the sum of the *squares* of the deviations of the scores from the mean, divided by the number of number of scores.

$$\mu_2 = s^2 = \sigma^2 = \frac{1}{n} \sum_{i=1}^{n} \left(x_i - \bar{x} \right)^2 .$$

The variance measures the spread of the distribution by averaging the effect of large and small deviations from the mean. Only the absolute value of these deviations are captured. That is, because these deviations $(x_i - \bar{x})$ are squared, the side of the mean on which they occur is lost. The relative location of the location of the mean in the range of the distribution is addressed.

The square-root of the variance, $\sqrt{\mu_2} = \sigma$ or s, is known as the *standard deviation*. Statistical analyses will often report the standard deviation because its value is more intuitively grasped: it represents the average deviation between the mean and the observed values.

A 2.3.3 The 3rd central moment (μ_3 : skewness)

The third moment about the mean is *skewness*, sometimes abbreviated to *skew*, is a measure of the degree of asymmetry of a distribution. The tapering sides of a distribution are called *tails.* Skewness is defined as

$$\mu_3 = \frac{\frac{1}{n}\sum_{i=1}^{n}(x-\bar{x})^3}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x-\bar{x})^2}} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x-\bar{x})^2}{\sqrt{\mu_2}}$$

where μ_2 is the second moment, the variance. In Figure A2.2 the curves on the right side of the distributions taper differently to those on the left. The skew of

(a) **is *negative*:** The left tail is longer than the right and the mass of the distribution is concentrated on the right of the figure. It has a few relatively low values. The distribution is said to be left-skewed. In such a distribution, the mean is lower than median, which is lower than the mode (i.e.; mean < median < mode). In this case the *skewness coefficien*t is lower than zero.

(b) **is *positive***: The right tail is longer than the left; the greater mass of the distribution is concentrated on the left of the figure. It has a few relatively high values. The distribution is said to be right-skewed. In such a distribution, the mean is greater than median, which is greater than the mode (i.e.; mean > median > mode); in which case the skewness coefficient is greater than zero.

If there is no skewness or the distribution is symmetric, as the bell-shaped normal curve, then the mean = median = mode.

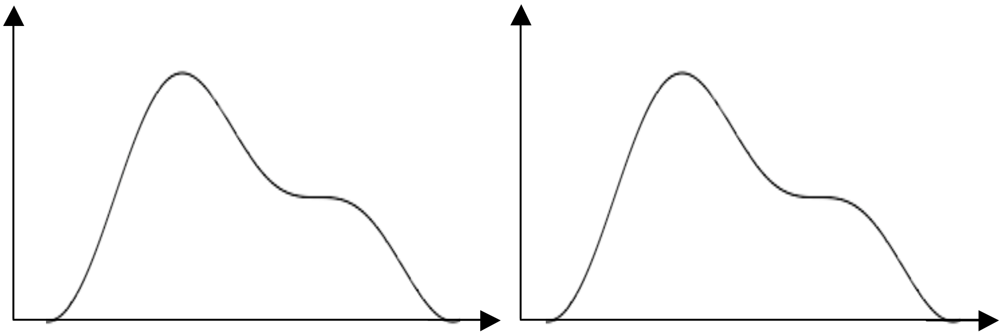


Figure A2.2. (a) Negative skew **(b) Positive skew**

In a skewed (lopsided) distribution, the mean is farther out in the long tail than is the median. Skewness is the third standardised moment, symbolised as γ_1 and defined as

$$\gamma_1 = \frac{\mu_3}{\sigma^3}$$

where μ_3 is the third moment about the mean and σ is the standard deviation. For a sample of n values, the sample skewness, g_1, is

$$g_1 = \frac{m_3}{m_2^{3/2}} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2)^{3/2}}$$

where x_i is the i^{th} value, $\bar{x}$ is the sample mean, m_3 is the sample third central moment, and m_2 is the sample variance.

Karl Pearson suggested two simpler calculations of the approximate measure of skewness:

1. $\dfrac{\text{(mean - mode)}}{\text{standard deviation}}$ and 2. $\dfrac{\text{3(mean - median)}}{\text{standard deviation}}$

A 2.3.4 The 4th central moment (μ_4 : kurtosis)

The kurtosis is the statistic that indicates whether the samples are clustered closely around the mean the mean or spread out over a wide range with many scores at the extremes. The kurtosis is defined as

$$\mu_4 = \frac{1}{n}\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^4$$

A distribution that had a disproportionate number of scores at the extremes of their range is know as *fat-tailed* or *leptokurtotic*, as illustrated in Figure A2.3.

Nu	*Xi*	*Omnicron*	*Pi*	*Rho*	*Sigma*	*Tu*	*Upsilon*	*Phi*	*Chi*	*Psi*	*Omega*
Ν ν	Ξ ξ	Ο ο	Π π	Ρ ρ	Σ σ	Τ τ	Υ υ	Φ ϕ	Χ χ	Ψ ψ	Ω ω

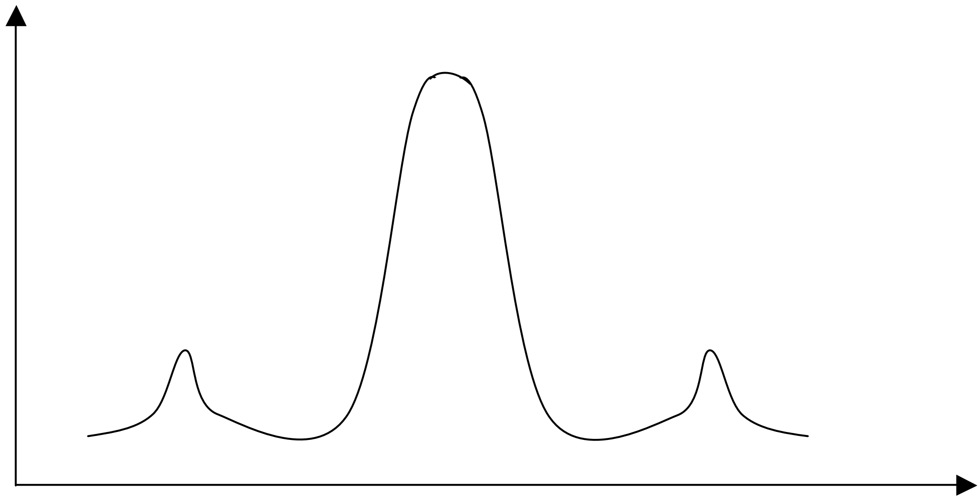


Figure A2.3. Graph of a leptokurtotic function with a small positive skew.

A 2.4 Stationary processes and time series analysis

A stationary process is a stochastic process whose joint probability distribution does not change when shifted in time or space. As a result, parameters such as the mean and variance, if they exist, also do not change over time or position.

Stationarity is used as a tool in time series analysis, where the raw data are often transformed to become stationary. For example, economic data are often seasonal and/or dependent on the price level. Processes are described as *trend stationary* if they are a linear combination of a stationary process and one or more processes exhibiting a trend. Transforming this data to leave a stationary data set for analysis is referred to as de-trending.

A uniform distribution, also known as *white noise*, is stationary. However, the sound of a cymbal crashing is not stationary because the acoustic power of the crash, and hence its variance, diminishes with time. An example of a discrete-time stationary process is a Bernoulli scheme, where the sample space is discrete, so that the random variable may take one of N possible values. Other examples of a discrete-time stationary process with continuous sample space include autoregressive and moving average processes–both of which are subsets of the autoregressive moving average model.

A 2.5 Independent random variables

The previous section describes independence of events. this section considers the independence of random variables. If X is a real–valued random variable and a is a number, then the set of outcomes that corresponds to the value of X being less–than–or –equal–to the value of event a is written $\{X \le a\}$.

Since such a set of outcomes have probabilities, it makes sense to refer to them as events that are independent of other events of this sort. Two random variables X and Y are independent if–and–only–if, for any numbers a and b, the events $\{X \le a\}$ (the set of outcomes where values of X are less –than–or–equal–to a) and $\{Y \le b\}$ are independent events as described above. Similarly, an arbitrary collection of random variables—possibly more than just two of them—is independent precisely if, for any finite collection X_1, ..., X_n and any finite set of numbers a_1, ..., a_n, the events $\{X_1 \le a_1\}$,..., $\{X_n \le a_n\}$ are independent events as defined above.

A 2.6 Correlation

In statistical usage, the strength and direction of the linear relationship between two random variables is known as their correlation, and the measure of it is called their *correlation coefficient*. Essentially, the correlation coefficient of two random variables is a measure of the extent to which they are not independent. The correlation of two random variables is defined only if both of their standard deviations are finite and nonzero. Correlation strength varies from 0 (no correlation) to 1 (identical) and the *direction* of the correlation varies from 1 (same direction) to -1 (opposite direction). The *value* of the correlation coefficient varies from -1 (negatively correlated) to 0 (no correlation) to 1 (highly correlated). The closer the correlation coefficient of two random variables is to -1 or 1, the stronger is the correlation between them.

It is important to note that the *causes* underlying a correlation may be unknown, and establishing a strong correlation between two random variables is not a sufficient condition to establish a causal relationship, positive or negative.

Cross-correlation is a term used in measuring the similarity of two time series. In signal processing, its value represents the measure of similarity of two waveforms as a function of a time-lag that is applied to one of them. Because the technique employed to calculate cross-correlation involves the taking of successive inner–products of the two time signals, it is also known as a *sliding dot product* or *sliding inner-product*. Cross-correlation techniques are commonly used to search for a short feature in a longer duration signal, such as a motive or phrase in a birdsong extemporisation, for example.

Autocorrelation is a technique for finding repeated and/or underlying patterns in a time–series; the cross-correlation of a signal with itself. Autocorrelation is frequently used, for example, to find the ‘missing fundamental’ frequency as implied by the presence of its harmonics.

Decorrelation is a term for any process that is used to reduce autocorrelation within a signal while preserving other aspects of the signal such as its statistical distribution characteristics. An example of an uncorrelated time series is Brownian motion; an independent random walk in which the size and direction of each move is not dependant on the size or direction of any previous moves.

A statistical analysis of time series data is concerned with the distribution of values without the *sequence* of values in a time series being taken into account and so it completely disregards any spectral (correlation) information. Thus, Bachelier’s hypothesis that market prices are a random walk was an hypothesis, recently shown by Mandelbrot and others not to be true, that there is no autocorrelation (such as trends) in market prices.

A simple method for decorrelating a time-series while preserving its statistical properties is to randomly shuffle the order of its samples.

Appendix 3

SPECIFICATIONS: CAPITAL MARKET TRADING DATA

A3.1 Introduction

Put simply, Exchange Trade Data it is any time-ordered data generated by a trading exchange as a record of the trading activity on that exchange and as such it falls into the broad category of multivariate time series. Typically, the complete set of all exchange data is quite voluminous. For example, the ASX, about 1% of the global economy, generates over 100 Mb of ASCII data daily. The structure of trading data from other trading engines is similar, though not identical, with that described here.

A TRADE is the exchange registration of the movement of a specified number of a security from one member of the exchange to another, including possibly from a member to themselves. Members are typically brokers, and all TRADEs are novated[1] by the exchange. All off-market transactions between brokers have to be reported to the exchange within a specified time period, as determined by the rules of the exchange, usually on the same day of trade and usually after normal trading times.

A TRADE is an amalgam of individuals' trading activity. TRADEs in an exchange's Orderbook contains <brokerID> but not <traderID> tags. Under some circumstances <brokerID> is undisclosed, either for a period of time (say 3 days) or until the outstanding monetary value of the order is less than a fixed amount, as set by the exchange.

Trading engine data can be divided into a few broad categories:

- Market Depth Data. Records of Buy and Sell Bids [<Price>, <Volume>, <TraderID>] generated by potential traders “positioning” themselves around a price, frequently that of the last trade, with instructions which are executed under certain circumstances, such as when the last trade is at the same price as that of another order in the orderbook.
- Trade Data. For all instruments on a exchange, each Trade Data packet (called a Tick) contains information about the date-of-trade (Date). time-of-

[1] Novation is a process in law in which, with the consent of all parties, one contract is entered into in substitution for and supersession of another contract.

trade (Time), trading session (Day, Night), number of units traded (called the Volume), and the Price at which they are traded. In addition, the Broker ID code and for large (institutional) traders, the Trader-ID-Code is sometimes revealed.

- Open Interest. For derivative instruments such as Futures contracts and Options, Open Interest reflects the number of currently open contracts. Data for each trading session is available sometime after market close and before the next trading session commences.
- Pre– and End–Of–Trading Events that contains information about the Time (Duration), Price Statistics (Open, High, Low and Close) Period Volume.

Intraday trading data: Cleaning procedure and metadata structure

The dataset used for the intraday trading data sonifications is of one financial quarter (sixty-five trading days) of a generic market's order-book day files. The data contains the multiplexed orderbook activity of approximately 3000 thousand trading instruments, listed firstly in action-time sequence at the resolution of one second and then by unique number according to their entry into the order-book. This data had to undergo extensive cleaning before being usable. The magnitude of the dataset (each day file was approximately 100 MB of ASCII data) demanded that the data–cleaning be undertaken algorithmically. The Metadata Specification here is from the cleaned dataset. Like the dataset itself it represents a generic structure as all references to a specific exchange, specific dates and specific securities have been omitted in–keeping with the confidentiality agreement between the parties signatory to the research proposal.

Metadata: Description of order types

<date code>.csv files contain one message per line (separated by a newline character (<\n>). Lines in each file are sorted in the order that the messages were sent from the trading engine. This should be in the same order as the <timestamp> attached to each message, but this has not been verified.

Descriptions of the fields used in the ORDER TYPES follow them.

ENTER:

```
<timeStamp>,ENTER,<securityID>,Bid|Ask,<orderID>,<price>,
<disclosedVolume>,<disclosedValue>,<flags>,U,<brokerTrade
rRefs>,<\n>
```

Description: A new order, or the untraded volume & undisclosed volume of an order carried forward from the last trading day (will have 00:00:00 timeStamp). Ignore the "U" field since this is for undisclosed volume (which is not provided in this dataset).

DELET:

```
<timeStamp>,DELET,<securityID>,<orderID>,B|A,<\n>
```

Description: Deletion of the untraded volume and untraded undisclosed volume of an order. i.e. an order cancellation. B for a bid order, A for an ask order.

AMEND:

```
<timeStamp>,AMEND,<securityID>,Bid|Ask,<oldOrderID>,<newO
rderID>,X,<newPrice>,<newDisclosedVolume>,<newDisclosedVa
lue>,<flags>, <traderID>,<\n>
```

Description: An amendment to an order. After an order is amended, it is referred to by its new ID, not its old ID. The traderID is that of the trader who amended the order.

TRADE:

<timeStamp>,TRADE,<securityID>,<tradeID>,<price>,<disclosedVolume>,
<disclosedValue>,<flags>,<bidorderID>,<askorderID>,<\n>

Description: An on-market trade.

OFFTR & CANTR:

```
<timeStamp>,OFFTR|CANTR,<securityID>,<offtrID>,<executeTi
meStamp>,
<price>,<disclosedVolume>,<disclosedValue>,<flags>,
<bidBrokerTraderRefs>,<askBrokerTraderRefs>,<\n>
```

Description: An off-market trade, or a cancelled trade (the orders involved are not re-inserted into the order-book).

FIELD:

```
<timeStamp>,FIELD,<securityID>,<Security><[1-5]>
```

Description: An update to the group number of a security. The last field contains the group number.

CONTL:
`<timeStamp>,CONTL,<securityID><statusText>,<\n>`
Description: Pre-open/opening/open/suspend/close/adjust period/trading halt for a particular security or a particular group or all securities.

FINIS:
`<timeStamp>,FINIS,<\n>`
Description: Signifies that there are no more messages for this day (can probably be ignored – only useful in real-time data). `<timeStamp>` is usually `24:00:00`.

Field	Description
<timeStamp>	Message time stamp reported by the trading engine. HH:MM:SS
<securityID>	Same as <securityID> in securityinfo.txt file. If this is "-" then there is a mapping problem in the data. In this case, we suggest you ignore these messages.
<orderID>	Order id. A natural number. Unique for the current day.
<price>	Price to exactly 3 decimal places, prefixed by '$'.
<disclosedVolume>	Disclosed volume. A natural number.
<disclosedValue>	Disclosed value, rounded to exactly 2 decimal places.
<flags>	Containing a list of flag codes. If there are no flags a double-quoted blank string is emplaced. Refer to market.cfg for descriptions of flags.
<brokerTraderRefs>	Broker reference and trader reference of the order (if they exist). ()|(@<brokerID><" ">&<traderID>)| (@<brokerID>)|(&<traderID>)
<newOrderID>	New order id of an amended order. After an order is amended, it is referred to by its new id (not its old id).
<newPrice>	New price of an amended order, to exactly 3 decimal places, prefixed by '$'.
<newDisclosedVolume>	New disclosed volume of an amended order. A natural number.
<newDisclosedValue>	New disclosed value of an amended order, rounded to exactly 2 decimal places.
<brokerID>	Broker id. A natural number prefixed by '#'.
<traderID>	Trader id. A natural number prefixed by '#'.
<tradeID>	Trade id for an on-market trade. A natural number.
<bidorderID>	<orderID> of the bid order.
<askorderID>	<orderID> of the ask order.
<offTrID>	Trade id for an off-market trade. A natural number.
<executeTimeStamp>	TimeStamp for the time the trading engine reports an off-market trade as happening (will probably be different to <timeStamp>). HH:MM:SS|H:MM:SS
<bidBrokerTraderRefs>	Broker reference and trader reference of the bid order (if they exist). B()|B(@<brokerID><" ">&<traderID>) |B(@<brokerID>)|B(&<traderID>)
<askBrokerTraderRefs>	Broker reference and trader reference of the ask order (if they exist). A()|A(@<brokerID><" ">&<traderID>) |A(@<brokerID>) |A(&<traderID>)
<statusText>	Control message state for this security, and whether it was initiated for a particular security (in this case no control type is stated), a security group or all securities. <status>[<" "><controlType>]
<status>	The status of the security., Can be either: Open, Closed or Suspended Opening: Not open for trading nor order entry, because opening-initiated trading is occurring. PreOpen: Open for order entry but not trading. Adjust-Period Trading-Halt
<controlType>	Whether the control was initiated for a particular security group ("group"), or for all securities ("system").

Appendix 4

CODE LISTINGS

INCLUSIONS

A 4.1 TO GENERATE & PLOT NET RETURNS OF THE XAO DATASET (SEE §6.6.1)

THIS PAGE HAS INTENTIONALLY BEEN LEFT BLANK

```python
# -*- coding: utf-8 -*-
# ::::::::::::::::::::::::::::::::::::::::::::: Start of returns.py :::::::::::::::::::::::::::::::::::::::::::::::::::::::
# ::::::1::::::::2::::::::3::::::::4::::::::5::::::::6::::::::7::::::::8::::::::9::::::::0::::::::1::::::::2
#:set printoptions=header:0,portrait:n,number:y,left:2pc,right:2pc,top:25mm,bottom:25mm
"""
#--------o---------o---------o---------o---------o---------o---------o---------o---------o---------o
#--------o---------o-- Returns generation, plotting and soundfile generation --o---------o---------o
#--------o---------o---------o---------o---------o---------o---------o---------o---------o---------o

- Converts trading or similar data to returns
- Produces correlated and decorrelated time series data
- Generates stats and graphs
- Outputs to AIFC, or perhaps WAV or direct audio streaming

history:
v0.00   20040902 initial draft
v0.01   20040922 minor bugs + rework w. better stats
v0.02   20041024 add toAIFC
v0.03   20041024 minor bugs + improve toAIFC
v0.20   20051209 objectify
v0.21   20070126 reorder for np, scipy and matplotlib
v0.22   20090110 redo plotting routines
v0.23   to sonification.com.au/audification for ICAD paper review
"""
__author__      = "David Worrall ¿ 2004++"
__version__     = "0.2.3"
__docformat__   = "restructuredtext en"

import  string, ctypes, struct
import  os.path as path                 # for file paths and universal separators.
from    datetime   import datetime
from    scipy      import stats
import  numpy   as np                   # hurray for arrays!

import  aifc, pyaudio                   # for file writing and rt audio streaming
import  pylab
import  matplotlib.pyplot   as plt      # interface to matplotlib plotting routines
import  matplotlib.ticker   as ticker   # for controlling the axis labelling
import  matplotlib.mlab     as mlab     # for plotting returns and histograms

EOF = ''                                # End of File ID for readline
BITS_15 = pow (2, 14)                   # for bit-shifting audio samples from -1 to 1 into 16 bit gamut

#--------o---------o---------o---------o---------o---------o---------o---------o---------o---------o
#--------o---------o---------o-- Generic Functions --------o---------o---------o---------o---------o
```

```python
#--------o---------o---------o---------o---------o---------o---------o---------o---------o---------o
def printStringList (listy, stringy=None, ):
    """
    Pretty-prints a list of strings.
    Good for looking for directory entries containing whatever stringy is assigned to.
    Use Eg.: printStringList (dir(), stringy = 'xxx') to find out if xxx is in the dir()
    """
    beg =''
    for i in dir(listy):
        if i[0] != beg:
            beg=i[0]
            print
        else:
            if stringy != None: print i ,
            else:
                if stringy in i:    print i

def marketTimeString (dtTuple):
    """
    Converts datetime tuple to a string in the form yyyymmdd.hhmmss.
    # Eg. datetime.datetime(2009, 1, 26, 18, 20, 5, 198934) ==> '20090126.182005'
    """
    res = str([dtTuple.year, dtTuple.month, dtTuple.day, dtTuple.hour, dtTuple.minute, \
                                           dtTuple.second])[1:-1] # dtTuple elts as a single string
    res=res.split(',')
    res = [res[0]] + [('0'+i.strip(' '))[-2:] for i in res[1:]]         # res = [yyyy,mm,dd,hh,ss]
    return string.join(res[:3], '')+'.'+ string.join(res[3:], '')       # returns "yyyymmdd.hhmmss"
# marketTime (now)

def scaleArray (np1array, miny=-1, maxy=1, trace=False):
    """Scales a numpy array to between miny and maxy. Typically, npArray is a row of the returnsArray
    Why isn't this a numpy method?"""
    np1array = np.array(np1array)           # just in case! redundant if already a numpy array
    newrange = abs(float(maxy-miny))        # abs range of IP.The abs could be avoided -
                                            # it's there in case max and min are swapped on IP
    scaler = newrange/abs((np1array.max() - np1array.min()))
    minAdjust=(scaler * np1array.min()) - miny
    np1array = (scaler * np1array) - minAdjust
    if trace:
        print "newrange:", newrange, "--- scaler:", scaler, "--- minAdjust:", minAdjust
        print "Xchecks. scaled returnsMinMax:", (np1array.min(), np1array.max())
    return np1array

#--------o---------o---------o---------o---------o---------o---------o---------o---------o---------o
#--------o---------o---------o---------o- Returns class ---o---------o---------o---------o---------o
```

```python
#---------o---------o---------o---------o---------o---------o---------o---------o---------o---------o

class Returns:
    def __init__(self, IPdataFname, begDate=0, endDate=0, nr=None, field="close"):
        self.IPdatafile = IPdataFname    # the csv file of the market data. The shortname - extension
                                         # is used as part of the OPfilename. See returnsToAif method
        self.shortFname = self.IPdatafile.split(path.sep)[-1].split(path.extsep)[0]

        self.nrRecords = nr              # max number records to be read btwn begDate & endDate
        self.fieldName = field.lower()   # the dataField (column ) we want + drop all caps
        self.datefield = 'date'          # should be 0. asign to null to tell if it is present
        self.fieldNr =-1                 # init for column nr of data to be pulled
        self.sep = ','                   # for csv files. Could be an IPvar, or more sophisticated

        self.returnsArray = None         # set later when size is known
        self.origDataIx         = 0      # self.returnsArray[0] Use named ix's for code readability
        self.returnsIx          = 1      # self.returnsArray[1] overwritten for clipped returns
        self.decorrReturnsIx    = 2      # self.returnsArray[2] for decorrelated version of [1]
        self.uniformIx          = 3      # self.returnsArray[3] for uniform distribution & returns
        self.normalIx           = 4      # self.returnsArray[4] for uniform distribution & returns

        self.returnsStats = None         # Stats for the returns. See self.calculateReturns()
        self.begDate = begDate           # default is 0, ergo first record if not set
        if endDate == 0:
            self.endDate = datetime.today() # default is current date, ergo  EOF if not set
            self.endDate = self.endDate.replace(hour=0, minute=0, second=0, microsecond=0)
        else:
            self.endDate = self.secDateToDT(endDate)
        try:
            self.IPfile = open(self.IPdatafile, 'r')
        except:
            print "Unable to locate ",IPfname,". If file is not in cur. dir, use full pathname."
            return
        self.getDataFormat()
        self.getDataFromFile()
        self.IPfile.close()

#---------o---------o---------o---------o---------o---------o---------o---------o---------o---------o
    def getDataFormat (self, trace=False):
        """
        Reads first line of datafile for data format. Called on class instantiation.
        self.dataFormat is not referenced outside this method - so could be a local.
        """
        try:
            self.dataFormat = self.IPfile.readline()   # read first line of file
```

```python
                            # should be in <date>,<open> etc format or
                            #string of the form "Date,Open,High,Low,Close,Volume,OpenInterest"

            # convert dataFormat into a simple csv string
            if self.sep in self.dataFormat:
                self.dataFormat.replace('<', "") # toss the <fieldnme> formatting, if present ...
                self.dataFormat.replace('>', "") # no harm done if not.¿check  for empty space?
            if self.fieldName in self.dataFormat.lower():
                                                 # find field nr of the field (eg 'close) reqd
                self.fields = self.dataFormat.lower().split(self.sep) # drop caps, split to list

            # get the field numbers
            for i in range (len(self.fields)):
                if self.fields[i] == self.datefield:
                    self.datefield = i  # ugly type-change - but saves a var!
                if self.fields[i] == self.fieldName:
                    self.fieldNr = i
            if self.fieldNr == -1:
                print "Unable to locate ", self.fieldName, " in ", self.dataFormat, \
                        ". Check (non case-sensitive) spelling."
            if trace:
                print "datafields   :", self.dataFormat
                print "Datafield '", self.fieldName ,"' is col ", self.fieldNr, ". File is closed."
        except:
                print "Problem with data read."

#---------o---------o---------o---------o---------o---------o---------o---------o---------o---------o
    def secDateToDT (self, dateString):
        """
        Converts a security date to eight-digit in the form yyyymmdd to a datedtime structure
        six-digit dates are common so we have to convert 86 to 1986 and 05 to 2005
        dateString is in eithe '861025' or '19861025' or '081025' or '20081025'
        """
        dateString = dateString.replace('/','')         # get rid of separators
        dateString = dateString [-6:]                   # toss  yyyy if present, we'll calc it.'
        if (2000 + int (dateString[:2]) )> datetime.today().year:   # it must be a 1900 date so
            dateString = int('19'+ dateString)
        else:
            dateString = int('20'+ dateString)
        tmp=divmod(dateString,10000)                    # leftmost 4 digits is the year
        m, d = divmod(tmp[1],100)
        return datetime (tmp[0],m,d)

    def getDataFromFile (self, trace=False):
        """
```

```
        Reads data from file, sorts out the date and keeps the required field.
        Uses secDateToDT() to put date in datetime format.
        """
        tbuff=self.IPfile.readline()
        count = 0
        datafromFile = []                       # list into which file data is read, elt by elt
        while tbuff != EOF and self.nrRecords < count:
            tbuff=tbuff.split(self.sep)                         # NB default for self.nrRecords = None
#            dayte = tbuff[self.datefield].replace('/',"")      # convert date string into an int
            dayte = self.secDateToDT(tbuff[self.datefield])
            datafromFile.append(string.atof(tbuff[self.fieldNr]))  # pull fielddata requird to array
            tbuff=self.IPfile.readline()                        # read 1 new line from file
            count += 1
            if trace:
                print "record Nr:", count, dayte
#       --------------- Make the np array for all the data --------------# row0 4 data, row1 4 Rnet,
        self.returnsArray=np.empty([5, len (datafromFile)], dtype=float) # row2 4 normalised
        self.nrRecords = len (datafromFile)
        if trace:
            print "Length of returnsArray = ", self.nrRecords       # len (datafromFile)
        self.returnsArray[self.origDataIx] = datafromFile           # transfer list to np array

    def calculateReturns (self):
        """
        Calculates the Net returns of in col0 and stores them in col1 of the np returnsArray.
        """
        self.returnsArray[self.returnsIx,0] = 0                     # set  return of zeroth sample
        for i in range (1,len(self.returnsArray[self.returnsIx])):
            self.returnsArray[self.returnsIx,i]=(self.returnsArray[self.origDataIx,i] \
                    - self.returnsArray[self.origDataIx,i-1])/self.returnsArray[self.origDataIx,i]
        self.doStats()          # set the records straight!
#        self.returnsArray[self.returnsIx] = scaleArray(self.returnsArray[self.returnsIx])

    def decorrelateReturns (self):
        """
        Stores a decorrelated form of the net returns in returnsArray[sourcecol]
        in returnsArray[targetcol].
        """
        self.returnsArray[self.decorrReturnsIx]= \
                                        np.random.permutation(self.returnsArray[self.returnsIx])
    def makeUniformReturns (self):
        """
        Generates a uniform random distribution over the interval [-1.0,1.0)
        Calcs the returns and stores in self.returnsArray[uniformIx]
        This is the base 'flat' distribution aganst which others canbe compared
```

```
        """
#        self.returnsArray[self.uniformIx] = scaleArray(np.random.uniform(low=-1.0, high = 1.0,\
#                                                         size = self.nrRecords), trace=True)
        self.returnsArray[self.uniformIx] = np.random.uniform (low=-1.0, high = 1.0,\
                                                        size = self.nrRecords)
    def makeNormalReturns (self):
        """
        Generates a normal random distribution over the interval [-1.0,1.0)
        Calcs the returns and stores in self.returnsArray[normalIx]
        This is the distribution against which random (uncorrelated) walks can be compared
        """
        self.returnsArray[self.normalIx] = scaleArray(np.random.normal(size = self.nrRecords),\
                                                      miny=-1.0, maxy = 1.0)

#--------o---------o---------o---------o---------o---------o---------o---------o---------o---------o
    def doStats (self, trace=False):
        """
        Calculates a number of common statistical properties of the returns data. Prints if nesess.
        """
        self.returnsStats = list(stats.stats.describe(self.returnsArray[self.returnsIx]))
        self.returnsStats[0] = len (self.returnsArray[self.returnsIx])
        self.returnsStats[1] = (np.min(self.returnsArray[self.returnsIx]) ,\
                                        np.max(self.returnsArray[self.returnsIx]))
        self.returnsStats[2] = np.mean(self.returnsArray[self.returnsIx])
        self.returnsStats[4] = np.var(self.returnsArray[self.returnsIx])
        self.returnsStats.append(np.median(self.returnsArray[self.returnsIx]))
        if trace:
            print "Description\n-----------"
            print" nr samples ", self.returnsStats[0]
            print" min sample ", self.returnsStats[1][0]
            print" max sample  ", self.returnsStats[1][1]
            print" arith mean  ", self.returnsStats[2]
            print " median       ", self.returnsStats[6]
            print " mode         ", stats.stats.mode(self.returnsArray[self.returnsIx])[0][0]
            print" variance     ", self.returnsStats[3]
            print" skewness     ", self.returnsStats[4]
            print" kurtosis    ", self.returnsStats[5]
#           print " kurtosos test", stats.stats.kurtosistest(self.returnsArray[self.returnsIx])
#           print " normal test", stats.stats.normaltest(self.returnsArray[self.returnsIx])

    def clipReturns (self, nrMin=0, nrMax=0, trace=False):
        """
        Clips all min values of the returnsArray to the size of the nr Min'th & nrMax'th vals.
        Used to 'limit' the returns to be audified. The default position is no clipping.
        Could be used for autoclip based on the 'significant' of SNR improvment for soing so.
```

```
        Eg clipReturns(nrMin=1,nrMax=0) limits the most neg. val to the 2nd most neg.
        Used to clip  outliers to improve SNR of sample space prior to sonification.
        If zeroing the samples is more appropriate, use threshold like this:
            tmp2array=stats.stats.threshold(tmp2array, negLimit,posLimit)
        In this usage, better to clip to nearest to maintain firmer recog. of ss.
        nrMin, nrMax are the number of samples on either end of a sorted ss to be clipped.
        """
        tmpArray = np.copy(self.returnsArray[self.returnsIx])
        tmpArray.sort()                                 # necess to make for independent sort
        negLimit = tmpArray[nrMin]                      # the value below which all are clipped
        posLimit = tmpArray[-nrMax-1]                   # the value above which all are clipped

        # fill outliers with closest inlier vals as dictated
        negIndicesToLimit = []                          # for indices of -ve vals to fix
        posIndicesToLimit = []                          # for indices of +ve vals to fix
        for index in range(len(self.returnsArray[self.returnsIx])):
            if self.returnsArray[self.returnsIx][index] < negLimit:
                negIndicesToLimit.append(index)         # find a -ve outlier
                self.returnsArray[self.returnsIx][index]=negLimit    # fill it with most -ve inlies
            if self.returnsArray[self.returnsIx][index] > posLimit:
                posIndicesToLimit.append(index)         # find a +ve outlier
                self.returnsArray[self.returnsIx][index]=posLimit    # fill it with most +ve inlies
        self.doStats()                                  # set the records straight!
        if trace:
            print "Neg limit", nrMin, " vals to", negLimit
            print "Pos limit", nrMax, " vals to", posLimit
            print "-ve indexes to fix with %f:" % (negLimit)
            for i in negIndicesToLimit:
                print "\treturns [%i] is currently %f" %(i, self.returnsArray[self.returnsIx][i])
            print "+ve indexes to fix with  %f:" % (posLimit)
            for i in posIndicesToLimit:
                print "\treturns [%i] is currently %f" %(i, self.returnsArray[self.returnsIx][i])

    def arrayToAIFC (self, np1array, OPfname,
                     nrchans    = 1,
                     SR         = 44100,
                     nrReps     = 0,
                     gapSecs     = 0,
                     buffsize   = 4096,
                     trace      = False):
        """
        Writes npArray to fname.AIFC. Filename contains original DataFilename datetime-stamp.
        buffsize is size of AIFC write buffer. File writes are in buffsize increments.
        Write the arran nrReps number of times, with gapSec seconds between
        """
```

```
        OPfileID = aifc.open(OPfname+'.aifc', 'wb')         # open the file and set up the header
        OPfileID.setnchannels(nrchans)                      # the number of channels (1=mono)
        OPfileID.setsampwidth(2)                            # the sample width in bytes(16bit fmt)
        OPfileID.setframerate(SR)                           # the frame rate. for mono, FR =SR
                                                            # scale data -1 to 1
                        # scale (-1,1) the np.array, shift it 15 bits and overwrite it as int16s
        npSarray=BITS_15*np1array                          # NB can be a np.array. Doesn't have 2B
        npSarray=npSarray.astype('int16')                   # cvt to 16 bit ints.
                                                            # Use nparray.dtype say what type it is
                        # to python list - has to be a better way!!!!! ##
                        # try to pack it directly from the np1aray, as per pyAudioEg.py
        tmp = npSarray.tolist()
        dataString = "".join(struct.pack('>h', i) for i in tmp) # pack the array data into a buffer
        OPfileID.writeframes(dataString)                        # write buffer to file
                        # if the sound it to be repeated,insert a gapsSec silence b4 writing again
        if nrReps > 0:
            tmp = np.array (np.zeros(gapSecs * SR * nrchans, dtype='int16'))
            silentString = "".join(struct.pack('>h', i) for i in tmp) # pack zeros into a buff
            for i in range (nrReps):
                OPfileID.writeframes(silentString)              # write the silence buffer
                OPfileID.writeframes(dataString)                # write the non-silent buffer again
        OPfileID.close()
        if trace:
            numsamps = int(len(np1array)) + (gapSecs * SR * nrchans)
            print "Wrote %s.aifc, nrChans: %d, nrSamps: %d." % (OPfname, nrchans, numsamps)

#--------o---------o---------o---------o---------o---------o---------o---------o---------o---------o
    def outputAudioFiles (self, OPdir=None, SR = 44100, gapSecs = 0, trace=False):
        """
        Write the corr and decorr rtns and  same size uniform and normal returns
        OPdir is a string of the directory to which the .aifc files is output.
        """
        # Make sure the OPdir given is actually a directory
        if OPdir:
            if OPdir[-1] != path.sep:                           # '/' on unix etc, '\' on windows
                OPdir += path.sep                               # in case it has been left off
        if not path.isdir (OPdir):
            print "The directory to write to cannot be found."

        # Construct the filenames
        shortFname = self.IPdatafile.split(path.sep)[-1].split(path.extsep)[0]
        now = marketTimeString(datetime.today())                # 'yyyymmdd.hhmmss'
#        now= string.split(now, '.')[0]                         # drop the mins & secs

        # Make the output file names
```

```
        rtnFname    = OPdir + shortFname+'-ReturnsRaw_'+ now    # for rtns
        decorrFname = OPdir + shortFname+'-RtnsDecorr_'+ now    # for decorrelated rtns
        uniformFname= OPdir + shortFname+'-RtnsUniform_'+ now   # for white random rtns
        normalFname = OPdir + shortFname+'-RtnsNormal_'+ now    # for normal rtns
        CandDFname  = OPdir + shortFname+'-CorrAndDecorr_'+ now # for alternate A/B comparison
        UNDRtnFname = OPdir + shortFname+'-uniNormDecorRtn_'+ now # all 4 in sequence

        ret1dc1Fname = OPdir + shortFname+'-ret+2dcorr1_'+ now   # 1 x ret + 2 x decorr nr2
        ret1dc2Fname = OPdir + shortFname+'-ret+2dcorr2_'+ now   # 1 x ret + 2 x decorr nr2
        ret1dc3Fname = OPdir + shortFname+'-ret+2dcorr3_'+ now   # 1 x ret + 2 x decorr nr3

        # Concatenate the composite buffers (raw and Decorrelated + UNDR
        b0 = np.zeros (int(gapSecs*SR))
        b1 = scaleArray(self.returnsArray[self.returnsIx], miny=-1.0, maxy = 1.0)
        b2 = scaleArray(self.returnsArray[self.decorrReturnsIx], miny=-1.0, maxy = 1.0)
        b3 = scaleArray(self.returnsArray[self.uniformIx], miny=-1.0, maxy = 1.0)
        b4 = scaleArray(self.returnsArray[self.normalIx], miny=-1.0, maxy = 1.0)

        retAndDecorrArray = np.concatenate((b1,b0,b2))
        UNDRtnArray = np.concatenate((b3, b0, b4, b0, b2, b0, b1))

        # de-correlate the decorrelation, and again
        b5 = np.concatenate((b1, b0, np.random.permutation(b2), b0, np.random.permutation(b2)))
        b6 = np.concatenate((np.random.permutation(b2), b0, b1, b0, np.random.permutation(b2)))
        b7 = np.concatenate((np.random.permutation(b2), b0, np.random.permutation(b2), b0, b1,))

        if trace:
            print "Range cross-check:"
            print "     Array            Min    Max"
            print " _____________________________________ "
            print "     ReturnsRaw_     %.2f   %.2f" %  (b1.min(), b1.max())
            print "     RtnsDecorr_     %.2f   %.2f" %  (b2.min(), b2.max())
            print "    RtnsUniform_     %.2f   %.2f" %  (b3.min(), b3.max())
            print "     RtnsNormal_     %.2f   %.2f" %  (b4.min(), b4.max())
            print "  CorrAndDecorr_     %.2f   %.2f" %  (retAndDecorrArray.min(),
                                                         retAndDecorrArray.max())
            print "uniNormDecorRtn_     %.2f   %.2f" %  (UNDRtnArray.min(), UNDRtnArray.max())
            print "   ret+1xdcorr1_     %.2f   %.2f" %  (b5.min(), b5.max())
            print "   ret+1xdcorr2_     %.2f   %.2f" %  (b6.min(), b6.max())
            print "   ret+2xdcorr3_     %.2f   %.2f" %  (b7.min(), b7.max())

        # send multiples of the individual arrays  off to the audio OP routine
        self.arrayToAIFC (b1, rtnFname,     SR=SR, nrReps = 3, gapSecs = 1, trace=trace)
        self.arrayToAIFC (b2, decorrFname,  SR=SR, nrReps = 3, gapSecs = 1, trace=trace)
        self.arrayToAIFC (b3, uniformFname, SR=SR, nrReps = 3, gapSecs = 1, trace=trace)
```

```
        self.arrayToAIFC (b4, normalFname,  SR=SR, nrReps = 3, gapSecs = 1, trace=trace)
        # send combinations of the individual arrays   to the audio OP routine
        self.arrayToAIFC (retAndDecorrArray, CandDFname,    SR=SR, nrReps=3, gapSecs=2, trace=trace)
        self.arrayToAIFC (UNDRtnArray, UNDRtnFname,         SR=SR, nrReps=3, gapSecs=2, trace=trace)

        # send 2x different decorrelates + raw return to the audio OP routine
        self.arrayToAIFC (b5, ret1dc1Fname,  SR=SR, trace=trace)
        self.arrayToAIFC (b6, ret1dc2Fname,  SR=SR, trace=trace)
        self.arrayToAIFC (b7, ret1dc3Fname,  SR=SR, trace=trace)

        print "Finished writing audio files to", OPdir + shortFname
        # send multiples of the individual arrays  off to the audio OP routine
#        self.arrayToWAV (b1, rtnFname,     SR=SR, nrReps = 3, gapSecs = 1, trace=True)
#        self.arrayToWAV (b2, decorrFname,  SR=SR, nrReps = 3, gapSecs = 1, trace=True)
#        self.arrayToWAV (b3, uniformFname, SR=SR, nrReps = 3, gapSecs = 1, trace=True)
#        self.arrayToWAV (b4, normalFname,  SR=SR, nrReps = 3, gapSecs = 1, trace=True)
        # send combinations of the individual arrays   to the audio OP routine
#        self.arrayToWAV (retAndDecorrArray, CandDFname,    SR=SR, nrReps=3, gapSecs=2, trace=True)
#        self.arrayToWAV (UNDRtnArray, UNDRtnFname,         SR=SR, nrReps=3, gapSecs=2, trace=True)
        # send 2x different decorrelates + raw return to the audio OP routine

#--------o---------o---------o- End of Returns Class ------o---------o---------o---------o---------o

#--------o---------o---------o---------o---------o---------o---------o---------o---------o---------o
#--------o---------o---------o---- Play Audio Routines ----o---------o---------o---------o---------o
#--------o---------o---------o---------o---------o---------o---------o---------o---------o---------o

# mono test tone @ 44.1KHz
def sineTone (Fname="sine", freq=1000, dur=1,SR = 44100):
    abscissa = np.linspace(0,1,SR)                          # abscissa  is the x of y = f(x)
    data = np.sin(abscissa *2 * np.pi* freq)                # data      is the y of y = f(x)
    return data
    arrayToAIFC (data,OPdir+str(freq), trace=False)          # if Fname doesn't contain full path,
                                                            # file will be written to current dir

#--------o---------o---------o---------o---------o---------o---------o---------o---------o---------o
#--------o---------o---------o- Plotting Routines ---------o---------o---------o---------o---------o
#--------o---------o---------o---------o---------o---------o---------o---------o---------o---------o
# NB All plotting routines need a call to pylab.show() to display the plot
def plotXAO ():
    pylab.plot(xao.data, 'g.')                  # dots are '.'  ; pixels are ','
    pylab.xlabel('Contiguous Trading Days')
    pylab.ylabel('XAO Closing Value')
    pylab.grid(True)
#    pylab.savefig('XAOpix')                    # use if .png file is required
```

```python

def plotXAOreturns (col=1):
    """
    Plots the returns. col=1 is the returns, col=2 is the decorrelated returns.
    """
    pylab.plot(xao.returnsArray[col],'g-')       # dots are '.'  ; pixels are ','
    pylab.xlabel('Contiguous Trading Days')
    pylab.ylabel('XAO Net Returns')
    pylab.grid(True)
#    pylab.savefig('XAOnetRtns')                 # use if .png file is required
    pylab.savefig('XAOdecorrelatedReturns')      # use for decorrelated signal

def histMeanLine (color='#FF0000', linestyle='-', width = 1, yminmax=(0,100)):
    """
    Draws a vertical line t at or near the mean.
    'Near to' is there for situations where it would improve visibility of dataplots at the mean.
    """
    fudge = xao.returnsStats[2]+ 0.0002          # add a small o'set, when necess.
    meanLineX = [fudge,fudge]
    meanLineY = yminmax
    pylab.plot(meanLineX,meanLineY, color=color, linestyle=linestyle,\
                                            linewidth=1,  label='mean') # ¿ '_nolegend_' ?

def histXAOreturns (col=1, nrbins=10, color='green'):
    pylab.hist(xao.returnsArray[col], nrbins, facecolor=color, label='actual returns',alpha=0.25)
#    histMeanLine(fmat='r-', width=1, yminmax=(0,490))           # alternative to that below
    histMeanLine(color='#ff0000', linestyle='-', width=1, yminmax=(-20,480))
#    pylab.axis([-0.35,0.07, -30,890])                       # alternative to that below
    pylab.axis([-0.35,0.07, -20,500])                        # use this when there's more bins
    pylab.savefig('XAOhist+NormHist')                        # use if .png file is required

def histXAOclippedReturns (col=1, nrbins=10, color='green'):
    pylab.hist(xao.returnsArray[col], nrbins, facecolor=color, \
                label='clipped returns', align='mid', alpha=1)
    #histMeanLine(color='#FF0000', linestyle='-', width=1, yminmax=(-20,480))
    pylab.axis([-0.1,0.07, -30,450])
    pylab.savefig('XAOclipped+NormHist')             # use if .png file is required

#--------o---------o---------o---------o---------o---------o---------o---------o---------o---------o
def histNormal (color='000000', trace = False):
    """
    Plot a normal distribution with mu and sigma sa,e ar original or clipped dataset.
    To place mu at the centre of the abscissa space, use:
    #mu = xao.returnsStats[1][0]+ (0.5* (xao.returnsStats[1][1] - xao.returnsStats[1][0]))
    """
```

```python
    mu=xao.returnsStats[2]
    sigma = np.sqrt(xao.returnsStats[3])                              # sdev-[3] is variance
    normSamps = np.random.normal(mu, sigma, xao.returnsStats[0])      # [0] is nr datapoints
    dta , smallest, binsize, ignore = stats.histogram (normSamps, nrbins)
    histy, binEdges = np.histogram (normSamps, bins=nrbins, normed=0, new=False)
#    pylab.plot(binEdges,histy, 'b-', linewidth=1, color='#888800', antialiased=True,\
#           label='normal distribution,\nwith same St.Dev.', alpha=1)
    pylab.hist(normSamps, nrbins, facecolor=color, align='mid' ,bottom=None, \
                label='simulated normal\ndistrib. with same StD.', alpha=1)
    if trace:
        print "histy:", np.sum(histy) # ,histy
        print "binEdges", np.sum(binEdges) #, binEdges
        print 'size of normSamps:', len(normSamps)
        print ' sum of hist:', histy.sum()
        print "over same abscissa:", np.min(normSamps), np.max(normSamps)
        print "over SND histogram",  np.min(histy), np.max(histy)

def xAxis ():
    ax=pylab.subplot (111)
    majLoc = pylab.MultipleLocator(0.02)
    majFmat = pylab.FormatStrFormatter('%f') # or some other format - this puts the numbers on
    ax.xaxis.set_major_locator(majLoc)
#    ax.xaxis.set_major_formatter(majFmat)

    minLoc = pylab.MultipleLocator(0.01)
    ax.xaxis.set_minor_locator(minLoc) #for the minor ticks, use no labels; default NullFormatter

def yAxis ():
    ay=pylab.subplot (111)
    majLoc = pylab.MultipleLocator(50)
    majFmat = pylab.FormatStrFormatter('%d') # or some other format - this puts the numbers on
    ay.yaxis.set_major_locator(majLoc)
    ay.yaxis.set_major_formatter(majFmat)

    minLoc = pylab.MultipleLocator(10)
    ay.yaxis.set_minor_locator(minLoc) #for the minor ticks, use no labels; default NullFormatter

#    pylab.axhline(y=-50, color='black') # draw a line - why not!

def legend ():
    pylab.legend(loc='center left') #, size='medium')
#    pylab.legend(loc='top left') #, size='medium')

def labels ():
    s = 'Decorrelated Net Returns (%d bins)' % (nrbins)
```

```
    pylab.xlabel(s)
    pylab.ylabel('Frequency of occurrence')

#--------o---------o---------o- End of Plotting Routines --o---------o---------o---------o---------o

#--------o---------o---------o---------o---------o---------o---------o---------o---------o---------o
#--------o---------o---------o- __main__() code -o---------o---------o---------o---------o---------o
#--------o---------o---------o---------o---------o---------o---------o---------o---------o---------o
xao = Returns ("XAO_5725.csv", field ='close') # get original dataset from file etc
#plotXAO ()                                     # plots the original dataset

xao.calculateReturns ()                        # calcs the Prob. Density Fn (pdf)
#plotXAOreturns (col=1)                        # plots the pdf of the correlated returns
xao.clipReturns (1,0)                          # the dataset with the LH datapoint removed
xao.doStats (trace=True)                       # stats have changed, so reset the record!
xao.decorrelateReturns ()                      # decorr rtns[returnsIx] into retns[decorreatedRtnsIx]
xao.makeUniformReturns()
xao.makeNormalReturns()

#xao.outputAudioFiles (OPdir='/Users/drw/Desktop', trace=True) # output all returns to AIF files
xao.outputAudioFiles ('/Users/drw/Desktop/XAO-5725audiofiles/',
                                            SR = 16000,
                                            gapSecs = 1.0,
                                            trace = False)  # output all returns to AIF files

"""
The works and is used - just commented out for the mo. should be part of doit()
# uncomment the commands below to generate what's needed
plotXAOreturns (col=2)                          # plots the pdf of the correlated returns

    # _______________________________- MAKE THE HISTOGRAMS - ________________________________o
                    #calc for space just of most -ve sample:
                    # total s.sppce = 0.333204213427 + 0.0588620748375 = 0.39206628826450002
                    # diff btw least two: 0.33320421 - 0.08807546 = 0.24512875000000001
                    # proport. of total space = 0.24512875/0.39206628826450002= 0.6252227170182727

nrbins=200                                     # nr is somewhat arbitrary,easily changed!
#nrbins = int (200/(1- 0.625))                  # use this to keep the bin size the same for
#histXAOreturns(nrbins, color='#888888')        # compute the returns from the original dataset
#histXAOreturns(col=2,nrbins=nrbins, color='#00ff00')
                                               # the raw and the clipped.
#histXAOclippedReturns(col=1, nrbins=nrbins, color='#00ff00')
#histMeanLine(fmat='g-')
#histNormal(color='#000000')
legend()
```

```
labels()
yAxis()
#xAxis()
#pylab.grid(True)
#pylab.axhline(y=-50, color='black') # draw a line - why not!

#pylab.axis('off')
#pylab.savefig('XAOhist+NormHist')             # use if .png file is required
#pylab.savefig('XAOdecorrelatedReturns')
pylab.show()

"""
# results of the stats analysis = for the record
"""
==================================

Description b4 clip
-----------
 nr samples  5725
 min sample   -0.333204213427
 max sample   0.0588620748375
 arith mean   0.000217488453128
 variance     9.56858813276e-05
 skewness    -7.64911826438
 kurtosis   241.729886112
---
calc for space just of most -ve sample:
total s.sppce = 0.333204213427 + 0.0588620748375 = 0.39206628826450002
diff btw least two: 0.33320421 - 0.08807546 = 0.24512875000000001
proport. of total space = 0.24512875/0.39206628826450002  =   0.6252227170182727
---
Description after clip
-----------
 nr samples  5725
 min sample  -0.088075456212
 max sample   0.0588620748375
 arith mean   0.000260305703297
median       0.000236144211932
mode         0.0
 variance     7.76242316164e-05
 skewness    7.76106727987e-05
 kurtosis   12.2617257637
"""
```

THIS PAGE HAS INTENTIONALLY BEEN LEFT BLANK

A 4.2 TO CLEAN HIGH–FREQUENCY TRADING–ENGINE DATASET (SEE §6.8.1)

THIS PAGE HAS INTENTIONALLY BEEN LEFT BLANK

```python
# -*- coding: utf-8 -*-
# ::::::::::::::::::::::::::::::::::::::::::::: CMdataCleaner:.py :::::::::::::::::::::::::::::::::::::::::::::::::::::::::
# ::::::1:::::::::2:::::::::3:::::::::4:::::::::5:::::::::6:::::::::7:::::::::8:::::::::9:::::::::0:::::::::1:::::::::2
#:set printoptions=header:0,portrait:n,number:y,left:2pc,right:2pc,top:25mm,bottom:25mm
#for options in vim use :help popt¿option
"""
CMDataCleaner.py cleans the original 65 day files of CMdata, saving the cleaned files to a different directory.
"""
import glob, string

def trunc(x,n=0):
        """ truncates x, to n decimal places ... works for +ve and -ve floats and ints """
        if x> 0:
                return  round( x -0.5 * 10**-n,n)
        else:
                return -round( abs(x) -0.5 * 10**-n,n)

def fdata_to_list(fid):
        """Reads file by line into a list."""
        fid.seek(0,0)
        fdata=fid.readlines()
        return fdata

def printdata(fdata):
        """Prints file data list line by line."""
        count = 0
        for line in fdata:
                print count, line
                count = count + 1

IPdir="/Users/drw/Documents/R_DATA/CMdataOrig/"   # IP dir of sourcefile
OPdir="/Users/drw/Documents/R_DATA/CMdataClean/"  # OP dir of cleaned file

orderTypes={
                'ENTER': 11,
                'DELET': 5,
                'AMEND': 15,
                'TRADE':  10,
                'OFFTR': 11,
                'CANTR': 11,
                'FIELD': 2, #actually needs 3 but cludge here to trigger a fix for field 3
                'CONTL': 6,
                'FINIS': 3
                }
#print orderTypes['FINIS'] , orderTypes.has_key('FINIS') # for testing
```

```python

def priceCheck (order, priceField, volField, valueField):
        """ Checks whether priceField * volField = valueField to .... 2 decimal places"""
        try:
                not (round(string.atof(order[priceField][1:]) \
                                * string.atof(order[volField]),2) \
                                == round(string.atof(order[valueField]),2))
        except:
                print "PriceCheckError:", order

global NrErrors ; NrErrors = 0
def fixOrder(order):
        """ corrects known errors in various error types """
        global NrErrors
        NrErrors +=1
        if order[1]=='ENTER':
                priceCheck(order,5,6,7)
        elif order[1]=='DELET': pass
        elif order[1]=='AMEND':
                try:
                        order=order[:6] + order[7:11] + order[14:]      # delete last field
                        #fields are now  <timeStamp>,AMEND,<securityID>,Bid|Ask,<oldOrderID>,
                        #<newOrderID>,<newPrice>,<newDisclosedVolume>,<newDisclosedValue>
                except:
                        print "AMEND conversion error", order
                # X-check that <newPrice>*<newDisclosedVolume>=<newDisclosedValue>
                priceCheck (order,6,7,8)
        elif order[1]=='TRADE':
                try:
                        order=order[:5] + [string.joinfields(order[5:-4],"")[1:]] +order[-4:]
                        #fields are now <timeStamp>,TRADE,<securityID>,<tradeID>,
                        #<Price>,<Volume>,<bidOrderID>,<askOrderID>
                except:
                        print "TRADE conversion error", order
                # X-check that <newPrice> * <newDisclosedVolume> = <newDisclosedValue>
                priceCheck (order,4,5,6)
        elif order[1]=='OFFTR':
                priceCheck (order,5,6,7)
        elif order[1]=='CANTR':
                priceCheck (order,5,6,7)
        elif order[1]=='FIELD':
                try:                                    #NB assumes the GN= a single digit numeral
                        temp=string.splitfields(string.strip(order[2]),' ')
                        order=order[:2]+[temp[0]]+[temp[-1][-1]]
                except:
```

```
                        print "FIELD conversion error:", order
        elif order[1]=='CONTL':
                try:                                    #turn last field into 3 fields
                        temp=string.split(order[-1])
                        order=order[:-1]+[temp[0]]  + [string.join(temp[1:])]
                except:
                        print "CTRL conversion error:", order
        elif order[1]=='FINIS':
                try:
                        order=order[:2]                 # just timeStamp and 'FINIS'
                except:
                        print "FINIS conversion error:", order
        else: print " can't fix order of type", order[1]
        return order

def fixNewline(order):
        """ ensures that the last item in the order list is a \n and that it is the only one """
        if order[-1] != '\n':   # if id the last item in the order list is not already only a \n
                try:
                        if order[-1][-1]=='\n':          #if the last char in the last item is a \n
                                order[-1]=order[-1][:-1] # drop the \n
                        order += ['\n']                  # add a \n to the list
                except:
                        "fixNewline error", order
        return order

def cleanFiles():
        """
        Opens all the files in the IPdir and writes clean files of the same name into the OPdir.
        Reads & writes files line-at-a-time so can handle large files.
        """
        filelist=glob.glob(IPdir+"*.csv")
#       print filelist
        print
        print "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
        filecount=0
        for filename in filelist:
                try:
                        IPfileID=open(filename,'r')
                        shortFname=filename[string.rfind(filename,"/")+1:] # just the fname not the dir
                        print "Processing:", shortFname
                except:
                        "Can't open", filename
                        break
                OPfileID=open(OPdir+shortFname,'w',0)
```

```
                orderCount = errorCount = 0
                while 1:
                        order=IPfileID.readline()
                        if not order:
                                break                       # stop and EOF
                        order = string.splitfields(order,',')   # orders are 1 per line, of diff types
                        if orderTypes.has_key(order[1]):
                                order=fixOrder(order)
                                order=fixNewline(order)
                                OPfileID.write(string.joinfields(order,',')) # order to OPfile as a.csv
                                orderCount += 1
                        else:
                                print " orderType not recognised:", order
                                errorCount +=1
                print " Number of orders processed: ", orderCount,
                print " ...... Number of errors:", errorCount
                print "-------"
                IPfileID.close()
                OPfileID.close()
                filecount = filecount + 1
        print "Number of files processed: " , filecount
        print "FIN"

cleanFiles()    # Just do it!
```

A 4.3 TO CREATE HDF5 DATABASE OF ALL HDTF DATA (SEE §6.8)

THIS PAGE HAS INTENTIONALLY BEEN LEFT BLANK

```python
# -*- coding: utf-8 -*-
# :::::::::::::::::::::::::::::::::::::::::::: CMpyTables.py :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
# ::::::1:::::::::2:::::::::3:::::::::4:::::::::5:::::::::6:::::::::7:::::::::8:::::::::9:::::::::0:::::::::1:::::::::2
#:set printoptions=header:0,portrait:n,number:y,left:2pc,right:2pc,top:25mm,bottom:25mm
#for options in vim use :help popt¿option
#
# pyTables file and group classes for CM65 order data in HDF5 format
# last revision: drw 20070811
"""
Defines a H5file class which is a superstructure around a HDF5 file created with PyTables for the CM65 data.
The pyTable leaf classes for orderbook entries are also defined here.
orderbook structure: two techniques:
 - a 3D structure which represents the state of the orderbook at each point of change
       need to check whether pyTables/numpy can handle 3D array w. 1D blank
       day time (price vol) x n
    or can it handle a Dict?

The overall structure of the HDF5 datafile
------------------------------------------
root/MessageTexts                       # A Node with MessageText leaves
root/securitiesInfo/SecID12346Leaf      # The SecuritiesInfo node is created both as a h5 file node
                   /SecID22346Leaf      # and as a python a RAM Dict called self.SecurityNodesDict
.....                                   # This may eventually be eliminated.
root/securityID/OnTrades/OnTrade_Leaf1
                        /OnTrade_Leaf2
                        /.....
               /OffTrade/OffTrade_Leaf1
                        /OffTrade_Leaf2
                        /.....
               /CancelledTrades/CanceTrade_Leaf1
                        /CanceTrade_Leaf2
                        /.....
               /EnteredOrder/EnterOrder_Leaf1
                             /EnterOrder_Leaf2
                             /.....
               /DeletedOrder/DeleteOrder_Leaf1
                             /DeleteOrder_Leaf2
                             /.....
               /AmendedOrder/AmendOrder_Leaf1
                             /AmendOrder_Leaf2
                             /.....
               /Messages/MessageReference_Leaf1
                             /MessageReference_Leaf2
                             /.....

```

```python
root/securityID/ (etc)
"""
import string, tables, numpy        # for tables

# ::::::1:::::::::2:::::::::3:::::::::4:::::::::5:::::::::6:::::::::7:::::::::8:::::::::9:::::::::0:::::::::1:::::::::2
#------------- HDF5 file class definition --------------------
class H5file:
    def __init__(self, filename):
        self.fileName = filename    # the name string of the H5 file. It's usual for it to have a .h5 extension.
        self.shortFileName = self.fileName[string.rfind(self.fileName,"/")+1:] # just filename.ext not the directory
        self.fileID = None          # returned by the opening definitions. Used to access the file.
        self.openError = "Can't open ", self.fileName   # for the HDF5 file
        self.mode = None            # the mode of the the file opening. 'w' = write, 'a' = append, 'r' = read
        self.cntrlMessages = {}     # a dictionary whose keys() are CNTRL and FIELD messages found across all dayfiles.
        self._cntrlMessageNr = 0    # the incremental count of a number of unique CNTRL and FIELD flags.
                                    # Internal use only. See getFlagNumber().
        self.securitiesDict = {}    # A dict of tuples for each Node/Group in the root directory.
                                    # each tuple is in the form (CMgroupID, firstDay, lastDay,  ....)
                                    # should this Dict be held in RAM and written to the h5 file before close?

    def wOpen(self):
        """ Opens the HDF5 format file, if closed.
        Makes new one if filename doesn't exist. Returns the file descriptor."""
        if not self.fileID:     # no point trying to open it once it's already open
            try:                # assume it exists and we're appending
                self.fileID = tables.openFile(self.fileName, mode = "a", title = self.fileName)
                self.mode = self.fileID.mode    # do we need to keep a record of this?
            except:
                try:            # it didn't exist, so create it and open in write mode.
                    self.hdf5fileID = tables.openFile(self.fileName, mode = "w", title = self.fileName)
                    node0   = self.fileID.createGroup("/", "MessageTexts", "CM65 data for"+ nameString)

                    self.mode = self.fileID.mode        # do we need to keep a record of this?
                except:
                    print "Problem opening HDF5 file", self.fileName, " for writing."

    def rOpen(self):
        """ Opens the HDF5 format file  for reading, if closed. Returns the file descriptor."""
        if not self.fileID:
            try:
                self.fileID = tables.openFile(self.fileName, mode = "r", title = "CM65 data") # assume it exist
                self.mode = self.fileID.mode            # do we need to keep a record of this?
            except:
                print "Problem opening HDF5 file", self.fileName, " for reading."

```

```
    def close(self):
        if self.fileID:
            try:
                self.fileID.close()     # assumnes all tables have been flushed
                self.fileID = None      # reset
                self.mode = None        # reset NB instance data persists w. some (filename etc) data
            except:
                    "Trouble in closing HDF5 file", self.fileName

# ::::::1::::::::2::::::::3::::::::4::::::::5::::::::6::::::::7::::::::8::::::::9::::::::0::::::::1::::::::2
# This message stuff should probably be methods of the DayFile class,
                                                #?? except for the writing of the Dict to the H5 file.
    def getMessageNumber(self, message):
        """Updates the self.cntrlMessages dictionary. Keys are unique counting numbers. These (_cntrlFlagNr) numbers
        are incremented during the loading routines. NB the number is the message and the Dict key is the number.
        The complete MessagesDict is written to a MessagesNode in the H5 file and the number is included in
        the individual security H5 node."""
        if not message in self._cntrlMessages.keys():   # add a new message
            self._cntrlMessageNr += 1
            self._cntrlMessages[message] = self._cntrlMessageNr
        return self._cntrlMessages[message]      # return the number pointed to by the Dict index (the message)

#    def makeNewRootNodeDict(self):
#         """ Makes a new NodeList of the root directory."""
#         self.rootNodesDict = {}     # zero the list
#         if self.fileID:          # i.e. there is an hdf5 file open
#             for node in self.fileID.root:
#                 self.rootNodes[node._v_name]=()

    def printNodeInfo(self):
        """ Prints the entire node structure of the file."""
        for node in self.fileID.walkNodes():
            print node

    def appendSecurityBranch(self, nameString="s07007", description="Jame Bonds", CMgroup=0):
        """ Appends a new security branch to the root directory if it doesn't exist.
        Adds the sub-branch nodes to the securityID node to accommmodate order book info.
        Updates rootNodes dict."""
        if self.fileID.isopen:
            if not nameString in self.securitiesDict.keys():       # If it's already at self.fileID, leave it alone.
                try:
                    node0   = self.fileID.createGroup("/", nameString)
                    #self.securitiesDict[node0._v_name]=(CMgroup,)
                                        # add it to the self.SercuritiesDict !!!!NB for generating a summary
                    # print "securitiesDict: Node: ", node0._v_name, self.securitiesDict
```

```
                except:
                    print "File:", self.fileName, ": Can't create a node for " , nameString
                    self.printNodeInfo()
                try:    # Append the Orderbook nodes to the securityID branch.
                                                        # "nodeX" assignment to prevent printing. Alternative?
                    node1   = self.fileID.createGroup("/"+nameString, "OnTrade")      # for OnTrade leaves
                    table1  = self.fileID.createTable(node1, "OnTradeTable", OnTrade)
                    node2   = self.fileID.createGroup("/"+nameString, "OffTrade")     # for OffTrade leaves
                    table2  = self.fileID.createTable(node2, "OffTradeTable", OffTrade)
                    node3   = self.fileID.createGroup("/"+nameString, "CancelTrade")  # for CancelTrade leaves
                    table3  = self.fileID.createTable(node3, "CancelTradeTable", CancelTrade)
                    node4   = self.fileID.createGroup("/"+nameString, "Bid")          # for EnterBidOrder leaves
                    table4  = self.fileID.createTable(node4, "BidTable", EnterOrder)
                    node5   = self.fileID.createGroup("/"+nameString, "Ask")          # for EnterAskOrder leaves
                    table5  = self.fileID.createTable(node5, "AskTable", EnterOrder)
                    node6   = self.fileID.createGroup("/"+nameString, "BidDelete")    # for DeleteBidOrder leaves
                    table6  = self.fileID.createTable(node6, "BidDeleteTable", DeleteOrder)
                    node7   = self.fileID.createGroup("/"+nameString, "AskDelete")    # for DeleteAskOrder leaves
                    table7  = self.fileID.createTable(node7, "AskDeleteTable", DeleteOrder)
                    node8   = self.fileID.createGroup("/"+nameString, "BidAmend")     # for AmendBidOrder leaves
                    table8  = self.fileID.createTable(node8, "BidAmendTable", AmendOrder)
                    node9   = self.fileID.createGroup("/"+nameString, "AskAmend")     # for AmendAskOrder leaves
                    table9  = self.fileID.createTable(node9, "AskAmendTable", AmendOrder)
                    node10  = self.fileID.createGroup("/"+nameString, "Message")      # for Message leaves
                    table10 = self.fileID.createTable(node10, "MessageTable", OnTrade)
                except:
                        print "File:", self.fileID.fileName, ": Can't create a sub-node of " , nameString
                        self.printNodeInfo()
            else:
                print nameString, "is already in self.SecurityNodesDict."
        else:
            print self.fileID.fileName, "is not open."    # this might not work

    #def (self,
    def deleteNodes(self, nodeList):
       pass

#-------------end of  HDF5 file class stuff -------------------

""" test.......
H5filename="CM65datafile.h5"
#H5filename="/Users/drw/databases/sixpence/CM65datafile.h5"
h5file=H5file(H5filename)     # instantiate the h5 file object
h5file.wOpen()                # open for writing/appending
h5file.MakeSecuritiesTable()  # for dstoring the list of securities and their summary data.
```

```
print h5file.fileID
h5file.closeHDF5()
"""
# ::::::1:::::::::2:::::::::3:::::::::4:::::::::5:::::::::6:::::::::7:::::::::8:::::::::9:::::::::0:::::::::1:::::::::2
#
#---------- Begin pyTable Leaf classes for pyTable Groups. (Each securityID is a Node in the root ). ---------
#
# ----------------- Trade leaf classes:  -----------------
# an on-market trade order ("TRADE")
class OnTrade(tables.IsDescription):
    day         = tables.UInt8Col()         # unsigned short int for day number
    time        = tables.UInt16Col()        # seconds from midight (=0 and 86400). 10h00 = 36000, 16h00 = 57600
    tradeID     = tables.UInt32Col()        # trade identifier. A unique natural number
    secPrice    = tables.Float32Col()       # price of 1 unit of the security
    secVolume   = tables.UInt16Col()        # (disclosed) volume of the security
    secValue    = tables.Float32Col()       # (disclosed) volume of the security
    flagsMask   = tables.UInt64Col()        # 64 bit mask of the order's flags (see flags.py)
    bidOrderID  = tables.StringCol(16)      # Order ID: A number string prepended by '@'
    askOrderID  = tables.StringCol(16)      # Order ID: A number string prepended by '@'

# an off-market trade order ("OFFTR")   NB no ask broker
class OffTrade(tables.IsDescription):
    day         = tables.UInt8Col()         # unsigned short int for day number
    time        = tables.UInt16Col()        # seconds from midight (=0 and 86400). 10h00 = 36000, 16h00 = 57600
    executeTime = tables.UInt16Col()        # time that trade was effected (as reported by the trading engine)
    tradeID     = tables.UInt32Col()        # order identifier. A unique natural number
    secPrice    = tables.Float32Col()       # price of 1 unit of the security
    secVolume   = tables.UInt16Col()        # (disclosed) volume of the security
    secValue    = tables.Float32Col()       # (disclosed) value of the security
    flagsMask   = tables.UInt64Col()        # 64 bit mask of the order's flags (see flags.py)
    bidBrokerRefs = tables.StringCol(16)    # Bid Broker Trader Refs: A nr string prepended by '@#'
    askBrokerRefs = tables.StringCol(16)    # Ask Broker Trader Refs: 2 nr strings prepended by a '@#' and a '&#'

# a cancel-trade order ("CANTR") NB need to check whether these orders are on trades already in trade register
class CancelTrade(tables.IsDescription):
    day         = tables.UInt8Col()        # unsigned short int for day number
    time        = tables.UInt16Col()       # seconds from midight (=0 and 86400). 10h00 = 36000, 16h00 = 57600
    executeTime = tables.UInt16Col()       # time that trade was effected (as reported by the trading engine)
    tradeID     = tables.UInt32Col()       # order identifier. A unique natural number
    secPrice    = tables.Float32Col()      # price of 1 unit of the security
    secVolume   = tables.UInt16Col()       # (disclosed) volume of the security
    secValue    = tables.Float32Col()      # (disclosed) value of the security
    flagsMask   = tables.UInt64Col()       # 64 bit mask of the order's flags (see flags.py)
    bidBrokerRefs = tables.StringCol(16)    # Bid Broker Trader Refs: A nr string prepended by '@#'
    askBrokerRefs = tables.StringCol(16)    # Ask Broker Trader Refs: 2 nr strings prepended by a '@#' and a '&#'
```

```

# ----------------- Order Book leaf classes: Leaves are used for both Bid and Ask nodes. -----------------

# ::::::1:::::::::2:::::::::3:::::::::4:::::::::5:::::::::6:::::::::7:::::::::8:::::::::9:::::::::0:::::::::1:::::::::2
# an enter-order order ("ENTER")
class EnterOrder(tables.IsDescription):
    day         = tables.UInt8Col()         # unsigned short int for day number
    time        = tables.UInt16Col()        # seconds from midight (=0 and 86400). 10h00 = 36000, 16h00 = 57600
    orderID     = tables.UInt32Col()        # order identifier. A unique natural number
    secPrice    = tables.Float32Col()       # price sought of 1 unit of the security
    secVolume   = tables.UInt16Col()        # (disclosed) volume of the security
    secValue    = tables.Float32Col()       # (disclosed) value of the security
    flagsMask   = tables.UInt64Col()        # 64 bit mask of the order's flags (see flags.py)
    BrokerID    = tables.StringCol(16)       # Broker ID string: A number string prepended by '@'
    TraderID    = tables.StringCol(16)      # Trader ID string. A number string prepended by '&'

# a delete-order order ("DELET")
class DeleteOrder(tables.IsDescription):
    day         = tables.UInt8Col()         # unsigned short int for day number
    time        = tables.UInt16Col()        # seconds from midight (=0 and 86400). 10h00 = 36000, 16h00 = 57600
    orderID     = tables.UInt32Col()        # order identifier. A unique natural number

# an amend order order ("AMEND")
class AmendOrder(tables.IsDescription):
    day         = tables.UInt8Col()         # unsigned short int for day number
    time        = tables.UInt16Col()        # seconds from midight (=0 and 86400). 10h00 = 36000, 16h00 = 57600
#    oldOrderID  = tables.UInt32Col()        # order identifier. A unique natural number
    newOrderID  = tables.UInt32Col()        # old order identifier. A unique natural number
    secPrice    = tables.Float32Col()       # new price sought of 1 unit of the security
    secVolume   = tables.UInt16Col()        # (disclosed) volume of the security
    secValue    = tables.Float32Col()       # (disclosed) value of the security
    flagsMask   = tables.UInt64Col()        # 64 bit mask of the order's flags (see flags.py)
    TraderID    = tables.StringCol(16)       # Trader ID string. A number string prepended by '&'

# Market Engine controls ("CONTL" and "FIELD") The leaf structure for each message in the root Messages node.
class MessageText (tables.IsDescription):
#    day         = tables.UInt8Col()         # unsigned short int for day number
#    time        = tables.UInt16Col()        # seconds from midight (=0 and 86400). 10h00 = 36000, 16h00 = 57600
#    ordertype   = tables.StringCol(8)       # Order type  (either 'CONTL' or 'FIELD')
    number      = tables.UInt8Col()         # A counting number derived from getMessageNumber()
    message     = tables.StringCol(32)      # for FIELD, a group number. For CONTL, a string from the following list:
                        # 'PreOpen system' , 'Suspended system' , 'Opening system' , 'Open system' ,
                        # 'Opening group' , 'ENQUIRE system' , 'ENQUIRE' , 'Adjust-Period system' , 'Suspended group'

# The leaf structure for the Messages node in each security MessageIndex node
```

```
class ControlOrder (tables.IsDescription):
    day         = tables.UInt8Col()         # unsigned short int for day number
    time        = tables.UInt16Col()        # seconds from midight (=0 and 86400). 10h00 = 36000, 16h00 = 57600
    messageNr   = tables.UInt8Col()         # A counting number derived from getMessageNumber()

# __________________________________ FIN: CMpyTables.py _______________
```

A 4.4 TO CONVERT HFTD FROM FLAT-FILE TO HDF5 FORMAT (SEE §6.8)

THIS PAGE HAS INTENTIONALLY BEEN LEFT BLANK

```python
# -*- coding: utf-8 -*-
# ::::::::::::::::::::::::::::::::::::::::::::: CMdataCleaner:.py ::::::::::::::::::::::::::::::::::::::::::::::::::::::::
# ::::::1:::::::::2:::::::::3:::::::::4:::::::::5:::::::::6:::::::::7:::::::::8:::::::::9:::::::::0:::::::::1:::::::::2
#:set printoptions=header:0,portrait:n,number:y,left:2pc,right:2pc,top:25mm,bottom:25mm
#for options in vim use :help popt¿option
"""
CMDataCleaner.py cleans the original 65 day files of CMdata, saving the cleaned files to a different directory.
"""
import glob, string

def trunc(x,n=0):
        """ truncates x, to n decimal places ... works for +ve and -ve floats and ints """
        if x> 0:
                return  round( x -0.5 * 10**-n,n)
        else:
                return -round( abs(x) -0.5 * 10**-n,n)

def fdata_to_list(fid):
        """Reads file by line into a list."""
        fid.seek(0,0)
        fdata=fid.readlines()
        return fdata

def printdata(fdata):
        """Prints file data list line by line."""
        count = 0
        for line in fdata:
                print count, line
                count = count + 1

IPdir="/Users/drw/Documents/R_DATA/CMdataOrig/"   # IP dir of sourcefile
OPdir="/Users/drw/Documents/R_DATA/CMdataClean/"  # OP dir of cleaned file

orderTypes={
                'ENTER': 11,
                'DELET': 5,
                'AMEND': 15,
                'TRADE':  10,
                'OFFTR': 11,
                'CANTR': 11,
                'FIELD': 2, #actually needs 3 but cludge here to trigger a fix for field 3
                'CONTL': 6,
                'FINIS': 3
                }
#print orderTypes['FINIS'] , orderTypes.has_key('FINIS') # for testing
```

```python

def priceCheck (order, priceField, volField, valueField):
        """ Checks whether priceField * volField = valueField to .... 2 decimal places"""
        try:
                not (round(string.atof(order[priceField][1:]) \
                                * string.atof(order[volField]),2) \
                                == round(string.atof(order[valueField]),2))
        except:
                print "PriceCheckError:", order

global NrErrors ; NrErrors = 0
def fixOrder(order):
        """ corrects known errors in various error types """
        global NrErrors
        NrErrors +=1
        if order[1]=='ENTER':
                priceCheck(order,5,6,7)
        elif order[1]=='DELET': pass
        elif order[1]=='AMEND':
                try:
                        order=order[:6] + order[7:11] + order[14:]      # delete last field
                        #fields are now  <timeStamp>,AMEND,<securityID>,Bid|Ask,<oldOrderID>,
                        #<newOrderID>,<newPrice>,<newDisclosedVolume>,<newDisclosedValue>
                except:
                        print "AMEND conversion error", order
                # X-check that <newPrice>*<newDisclosedVolume>=<newDisclosedValue>
                priceCheck (order,6,7,8)
        elif order[1]=='TRADE':
                try:
                        order=order[:5] + [string.joinfields(order[5:-4],"")[1:]] +order[-4:]
                        #fields are now <timeStamp>,TRADE,<securityID>,<tradeID>,
                        #<Price>,<Volume>,<bidOrderID>,<askOrderID>
                except:
                        print "TRADE conversion error", order
                # X-check that <newPrice> * <newDisclosedVolume> = <newDisclosedValue>
                priceCheck (order,4,5,6)
        elif order[1]=='OFFTR':
                priceCheck (order,5,6,7)
        elif order[1]=='CANTR':
                priceCheck (order,5,6,7)
        elif order[1]=='FIELD':
                try:                                    #NB assumes the GN= a single digit numeral
                        temp=string.splitfields(string.strip(order[2]),' ')
                        order=order[:2]+[temp[0]]+[temp[-1][-1]]
                except:
```

```
			print "FIELD conversion error:", order
	elif order[1]=='CONTL':
		try:				#turn last field into 3 fields
			temp=string.split(order[-1])
			order=order[:-1]+[temp[0]]  + [string.join(temp[1:])]
		except:
			print "CTRL conversion error:", order
	elif order[1]=='FINIS':
		try:
			order=order[:2]			# just timeStamp and 'FINIS'
		except:
			print "FINIS conversion error:", order
	else: print " can't fix order of type", order[1]
	return order

def fixNewline(order):
	""" ensures that the last item in the order list is a \n and that it is the only one """
	if order[-1] != '\n':	# if id the last item in the order list is not already only a \n
		try:
			if order[-1][-1]=='\n':		#if the last char in the last item is a \n
				order[-1]=order[-1][:-1] # drop the \n
			order += ['\n']			# add a \n to the list
		except:
			"fixNewline error", order
	return order

def cleanFiles():
	"""
	Opens all the files in the IPdir and writes clean files of the same name into the OPdir.
	Reads & writes files line-at-a-time so can handle large files.
	"""
	filelist=glob.glob(IPdir+"*.csv")
#	print filelist
	print
	print "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
	filecount=0
	for filename in filelist:
		try:
			IPfileID=open(filename,'r')
			shortFname=filename[string.rfind(filename,"/")+1:] # just the fname not the dir
			print "Processing:", shortFname
		except:
			"Can't open", filename
			break
		OPfileID=open(OPdir+shortFname,'w',0)
```

```
		orderCount = errorCount = 0
		while 1:
			order=IPfileID.readline()
			if not order:
				break				# stop and EOF
			order = string.splitfields(order,',')	# orders are 1 per line, of diff types
			if orderTypes.has_key(order[1]):
				order=fixOrder(order)
				order=fixNewline(order)
				OPfileID.write(string.joinfields(order,',')) # order to OPfile as a.csv
				orderCount += 1
			else:
				print " orderType not recognised:", order
				errorCount +=1
		print " Number of orders processed: ", orderCount,
		print " ...... Number of errors:", errorCount
		print "-------"
		IPfileID.close()
		OPfileID.close()
		filecount = filecount + 1
	print "Number of files processed: " , filecount
	print "FIN"

cleanFiles()	# Just do it!
```

A 4.5 TO EXTRACT HFTD TRADES DATA FOR ANALYSIS & SYNTHESIS (SEE §6.8.2.1)

THIS PAGE HAS INTENTIONALLY BEEN LEFT BLANK

```python
# -*- coding: utf-8 -*-
# ::::::::::::::::::::::::::::::::::::::::::::::: Start of CM_IDTradeAnalysis.py :::::::::::::::::::::::::::::::::::::::::::::
# ::::::1:::::::::2:::::::::3:::::::::4:::::::::5:::::::::6:::::::::7:::::::::8:::::::::9:::::::::0:::::::::1:::::::::2
# :set printoptions=header:0,portrait:n,number:y,left:2pc,right:2pc,top:25mm,bottom:25mm  #for options in vim use :help popt-option
"""
=======> CMdataAnalysis.py <=======
This code analyses the (cleaned) data in the 65 CMdatafiles, extracting various statistics.
NB ID = Itraday     ED = Extraday , except in the case of secID which is a security indentfier.
Main functions are analyseTrades(),  printAnalysisFile()
last update 20070811
"""
# ::::::::::::::::::::::::::::::::::::::::::::::::::::::::: begin code here  :::::::::::::::::::::::::::::::::::::::::::::
# ::::::1:::::::::2:::::::::3:::::::::4:::::::::5:::::::::6:::::::::7:::::::::8:::::::::9:::::::::0:::::::::1:::::::::2
from drwGeneric import *
import datetime, glob, string

def datetime2list(dttm):
        """
        Converts an object in the date, time or datetime format to a list of ints.
        Time format is hh:mm:ss.microsecs ; date format is yyyy-mm-dd
        Datetime is yyyy-mm-dd hh:mm:ss.microsecs
        By a quirk of fate, this function can also be used to convert a timestamp string "hh:mm:ss[.microsec]"
        into a list of ints which can then, by indirection, be used to create a datetime instance.
        Eg. t= datetime.datetime(*datetime2List("23:45:33.16789"))
        """
        dttm=str(dttm)
        for sep in ['-','.',' ']:
                dttm=dttm.replace(sep,':')      # replace all ws and other seps with ':'
        return map(int,dttm.split(':'))         # convert the split chars to ints b4 returning


# :::::::::::::::::::::::::::::::::::::::::::::::: analyse ID Trade orders :::::::::::::::::::::::::::::::::::::::::::::
# ::::::1:::::::::2:::::::::3:::::::::4:::::::::5:::::::::6:::::::::7:::::::::8:::::::::9:::::::::0:::::::::1:::::::::2
#IPdir="/Users/drw/Documents/R_DATA/CMdataClean/"
#OPdir="/Users/drw/Documents/R_DATA/CMdataAnalysis/"
IPdir="/Users/drw/Documents/R_DATA/CMdataCleaned/dates/"
OPdir="/Users/drw/Documents/R_DATA/CMdataAnalysis/"
oneD=datetime.timedelta(1)
oneH=datetime.timedelta(0,60*60)
oneM=datetime.timedelta(0,60)
oneS=datetime.timedelta(0,1)
securitiesDict={}       # for holding a daily data summary for each security. This data is written to analysis files
                        # key is securityIDs into most of <TRADE> data
densityProfile = {}     # a dictionary of timestamp keys into a list of [secID,tradeID]s
```

```python
#DstatsList = []        # for holding daily stats, mostly for ascertaining ID synth and timing variable quotents
#TstatsList = []        # for holding total stats, mostly for ascertaining ED synth and timing variable quotents

def analyseTrades():
        """
        Opens all the files in the IPdir and compiles general statistics on the ID trade data of the Securities.
        Focus is on     trade information not the order-book/market depth data.
        Reads & writes files line-at-a-time so can handle large files.
        Fills two distionaries: securitiesDict and densityProfile
        """
        global securitiesDict
        filelist=glob.glob(IPdir+"*.csv")
#       print filelist
#       filelist=filelist[:1]           # determine how many, for testing purposes
        print
        print "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
        print "%i files to process ..." % len(filelist)

        filecount=0
        for filename in filelist:
                print filename
                DtotNrTrades      = 0           # Day total number of trades
                DminTradePrice  = 0             # Day minimum trade price
                DmaxTradePrice = 0              # Day maximum trade price
                DminTradeVal    = [0,0]         # [Day minimum trade value, timeStamp]
                DmaxTradeVal   = [0,0]          # [Day maximum trade value, timeStamp]
#               tempData =[]                    # test
                try:
                        IPfileID=open(filename,'r')
                        shortFname=filename[string.rfind(filename,"/")+1:] # just the filename not the directory
                        print "Processing:", shortFname
                        securitiesDict={}
                except:
                        "Can't open", filename
                        break
                while 1:
                        order=IPfileID.readline()
                        if not order:
                                break                           # stop and EOF
                        order = string.splitfields(order,',')           # orders are 1 per line, of different types
                        if order [1] == 'TRADE':
                                #               0       1       2       3           4      5       6        ....
                                #fields are<timeStamp>,TRADE,<ticker>,<tradeID>,<price>,<volume>,<value>  <...>
                                # convert price and volume strings to numerics
# ::::::1:::::::::2:::::::::3:::::::::4:::::::::5:::::::::6:::::::::7:::::::::8:::::::::9:::::::::0:::::::::1:::::::::2
```

```
                                order[4] = round(float(order[4][1:]),3) # 3 d.p. round of price
                                order[5] = long(order[5])               # volume as long int
                                order[6] = round(float(order[6]),3)     # 3 d.p.round.Unlike price it doesn't have a '$'
                                tradeTime=datetime.time(*datetime2list(order[0])) # convert 'hh:mm:ss' to a datetime val
                                                                        # not an error! see comment in datetime2list()
                                if securitiesDict.has_key(order[2]):    # is already in the list, so update info
                                        tempData =  securitiesDict[order[2]] # get existing record from dict
                                        tempData[1]=tradeTime           # the last trade timestamp, in datetime format
                                        if  order[4] > tempData[3]:     # if new H for this SecID,
                                                tempData[3] =  order[4] # update H for this SecID
                                        elif order[4] < tempData[4]:    # if new low for this SecID,
                                                tempData[4] = order[4]  # update L 4 this SecID NBcan't be newH & newL!
                                        tempData[5]=order[4]            # update close/last price for this SecID
                                        tempData[6]+=order[5]           # increment vol for this SecID
                                        tempData[7]+=1                  # increment Nr of Trades for this SecID
                                        try:
                                            tempData[8]+= order[6]      # increment cumVol for this SecID
                                        except:
                                            print "tempData: ",  tempData
                                            print "Order    : " , order
                                            print float(tempData[8]) + order [6]
                                else:
                                        # add the security and trade info to dictionary
                                        #       0        1      2     3     4     5     6        7          8
                                        #firstTrade lastTrade   O     H     L     C     cumVol  nrTrades    value
                                        tempData= [tradeTime]*2+[float(order[4]) ]*4+[order[5],1,float(order[6])]
#                                       print tempData
#                               print tempData
                                securitiesDict[order[2]]=tempData
                                DtotNrTrades += 1
                                updateDensityProfile(order)
                IPfileID.close()
                writeIDanalysis(shortFname) # write the analysis data to file

                # now collate some _daily_ stats:
                # fname, DtotNrTrades, DminTradePrice, DmaxTradePrice, [DminTradeVal,when], [DmaxTradeVal,when]
                print " Number of securites:", len(securitiesDict)
                print " Number of intraday trades: ", DtotNrTrades
                filecount += 1
        print " --------------------------------\nNumber of files processed:", filecount
        print "FIN"

# :::::::1:::::::::2:::::::::3:::::::::4:::::::::5:::::::::6:::::::::7:::::::::8:::::::::9:::::::::0:::::::::1:::::::::2
def writeIDanalysis(shortFname):
        OPfileID=open(OPdir+"IDTradeAnal"+shortFname,'w',0)
```

```
        OPfileID.writelines("<secID>,<Otrade>,<Ctrade>,<OpenP>,<HighP>,< LowP>,<ClosP>,<vol>,<NrTds>,<·value>\n")
        for ticker in securitiesDict:
                x= str(securitiesDict[ticker][-1]) # fix value to ensure it has 2 decimal places (boring!)
                securitiesDict[ticker][-1] = x + ((string.rfind(x,'.')+2) - (len(str(x))-1))*"0"
                OPfileID.writelines(ticker+","+list2string(securitiesDict[ticker],',')+'\n')
        OPfileID.close()

def  updateDensityProfile(order):
        """
        Adds [ticker,trade id] ( i.e. order[2:4] ) to the densityProfile dict at tradetime ( order[0] ).
        Creates a new entry if none exists.
        NB tradetime/timestamp is a string in the form "hh:mm:ss", not a datetime value.
        Because densityProfile is a dict,  a print will not necessarily be in time-order.
        See printTimeProfile()
        Adds [bidorderID,askorderID] for further analysis later
        """
        if densityProfile.has_key(order[0]):
                densityProfile[order[0]] += [order[2:4]+order[-3:-1]]
        else:                   # add  [secID, tradeID, bidOrderID, askOrderID] to densityProfile dictionary
                densityProfile[order[0]]   = [order[2:4]+order[-3:-1]]

def  printDensityProfile(fname=0):
        """
         NB timestamp is a string in the form "hh:mm:ss", not a datetime value.
        Because densityProfile is a dict,  the print is not in time-order
        """
        count = 0
        maxdensity=0
        if fname==0:
                for timestamp in densityProfile:
                        den=len (densityProfile[timestamp])
                        if den>maxdensity: maxdensity = den
        #               print count, ":",timestamp,"density=",den, densityProfile[timestamp]
                        print count,  ","  ,  den,   "," ,  timestamp
                        count +=1
        else:
                try:
                        fname = OPdir + fname
                        fid=open(fname, 'w',0)
                        for timestamp in densityProfile:
                                den=len (densityProfile[timestamp])
                                value = 0
                                for trade in densityProfile[timestamp]:
                                        print trade
                                        value += int([trade][1])
```

```
                temp = timestamp,  ',', count, ',', den, ',', value, ',', '\n'
                print temp
                fid.writelines(temp)
            count += 1
        except: "Can't write to ", fname
        fid.close()
    print "max density:", maxdensity

# :::::::1:::::::::2:::::::::3:::::::::4:::::::::5:::::::::6:::::::::7:::::::::8:::::::::9:::::::::0:::::::::1:::::::::2
def printAnalysisFile(shortFname, colWidth=0):
    """
    Pretty-prints an analysis file generated by analyseTrades()
    """
    fid = open(OPdir+"IDTradeAnal"+shortFname+".csv",'r')
    x= fdata_to_list(fid)
    printdata( fdata_to_list(fid),colWidth) # print items Radjusted in a colWidth chars field

# now we need to create a seqence of trading times
def totalValue(densityList):
    """
    Returns the total value of all trades at a timestamp.
    densityList is a timestamp entry in the densityProfile dict,
    ie. a list of ['secID', 'value'] lists.
    'value' is a char form of a Value int.
    """
    value=long(0)                       # not necessary to cast, but good to make the point
    for item in densityList:
        value += long(item[1])
    return value

def chartimeToSecs(char):
    """
    converts a time in the form "hh:mm:ss" to seconds
    """
    res = datetime2list(char)
    return 3600 * res[0] + 60 * res[1] + res[2]
#print  chartimeToSecs("16:01:05")

def totalTimestampValue(adict):                 # , eventRate, sampleRate):
    """
    Generates a time-ordered list of total Value for each timestamp in adict in the form [secTime,totalVal]
    adjusts timeoffst to 0 - need to rework to make it either offset or fixed
    adict is densityProfile{}
    """
    timeStamps = adict.keys()
    timeStamps.sort()                   # cronological order. It works even though they are chars!
    timeOffset=chartimeToSecs(timeStamps[0]) # get the first timeStamp to use as offset
    synthList = []                          # eventually, a list of [timeOffset, ValToBeMapped]
    for timeStamp in timeStamps:
#       print chartimeToSecs(key), timeOffset   # add all values at timeStamp together
        synthList += [[chartimeToSecs(timeStamp)-timeOffset, totalValue(adict[timeStamp])]]
    return synthList

def valueToExpFreq(minFreq, value, valueRange, radix, nrFurcations):
    """ converts a Value to an exonential of 1/pitchFrequency """
    exponent=nrFurcations*(1-(float(value)/valueRange))
    return minFreq * pow(radix,exponent)
#print valueToFreq(100, 6000, 6000, 2, 4)

# :::::::1:::::::::2:::::::::3:::::::::4:::::::::5:::::::::6:::::::::7:::::::::8:::::::::9:::::::::0:::::::::1:::::::::2
def sonifyValues(adict):
#   adict is densityProfile{}
    valuesNM=totalTimestampValue(adict) # calculate density profile of entire dictionary
    mnmx= colMinMax(valuesNM, 1)    # find min and max totalVal for all trades at any timestamp in the dict
    range = mnmx[1]-mnmx[0]         # range of totalValues in all timestamps in adict # print minV, maxV
    for item in valuesNM:
#       exponent=furcations*(1-(float(item[1])/range))
#       item[1] = minF * pow(radix,exponent)            # item=[timestamp, value]
        item[1]=valueToExpFreq(80,item[1],range,2,4)
    # This is a simple example of how a sine synth is done.
    ENV01=((0,0),(50,1),(99,0))   # tuple of (x,y)=(%duration,amplitude) pairs
    env01= makeEnvArray(ENV01)  # an envelope array
    #print env01
#   SR=44100        # sample rate. for CD writes, set SR=44100 and setnchannels(2)
    OPdirectory="/Volumes/DRWSONUSB2/R_CMdata/CMaudio/"
#   OPdirectory="/Users/ceuser/audio/"
    OPfilename="testCM.aiff"
    fname=OPdirectory+OPfilename
    OPfileID = aifc.open(fname, 'w')
    OPfileID.setnchannels(1)      # the number of channels (1=mono)
    OPfileID.setsampwidth(2)      # the sample width in bytes (2=16bit format)
    OPfileID.setframerate(SR)     # the frame rate. For mono, FR =SR
    for item in valuesNM[:200]:
#       print item
        sinAIF(item[1], 0.125, env01, OPfileID) # an S(1) fn
    OPfileID.close()
# :::::::1:::::::::2:::::::::3:::::::::4:::::::::5:::::::::6:::::::::7:::::::::8:::::::::9:::::::::0:::::::::1:::::::::2

def cSoundScore(adict,fund,radix,furcats, OPfilename):
    """ produces a Csound score for instr 1 (see EO this F for declaration)"""
```

```
        timeStamps = adict.keys()               # get the keys (they're in string form)
        timeStamps.sort()                       # crono order. This works even though they are in string form
        timeOffset=chartimeToSecs(timeStamps[0]) # the 0th item will be earliest; value is in secs after midnight
        valuesNM=totalTimestampValue(adict)     # NM = density profile of adict as a list of [timestamp, value]'s
        mnmx= colMinMax(valuesNM, 1)    # find min and max totalVal for all trades at all timestamp in adict
        range = mnmx[1]-mnmx[0]         # range of totalValues in all timestamps in adict # print minV, maxV
#       Fmin=100
#       Fmax=5100
#       cLin= (Fmax-Fmin)/float(range)          # constant for linear mapping to frequency

#       ___________open .sco file for csound score________________
        OPdirectory="/Volumes/DRWSONUSB2/R_CMdata/CMcsound/"
#       OPdirectory="/Users/ceuser/audio/"
        fname=OPdirectory+OPfilename
        OPfileID = open(fname, 'w')
        dur = 1.0
        amp =1.0
        tempoTable='t   0       3600    ; 60 x time compression: 1 sec = 1 minute'
        waveform1='f    1       0       16384   10      1          ; sinw'
        waveform2='f    2       0       16384   9.5     1       0 ; 1/2 sine for envelope'
        OPfileID.write(tempoTable +'\n')
        OPfileID.write(waveform1 +'\n')
        OPfileID.write(waveform2 +'\n')
# ------------begin: for sonifying individual trades -------------------------------------------------------
#       for stamp in timeStamps:
#               stampSec=chartimeToSecs(stamp)# print stamp, adict[stamp]
#               for trade in adict[stamp]:#     print stampSec-timeOffset, cLin/float(trade[1]) #
#                       OPfileID.write('i\t1\t' +str(stampSec-timeOffset) + '\t' + str(dur) \
#                                        + '\t' + str(amp)  + '\t' \
#                                       + str(valueToExpFreq(fund,trade[1],range,radix,furcats)) + '\n')
# ------------end:  for sonifying individual trades -------------------------------------------------------
# ------------begin: for sonifying value of all trades at  timestamp ---------------------------------------
        for item in valuesNM:
                freq = valueToExpFreq(fund,item[1],range,radix,furcats) # item=[timestamp, value]
#               print freq
                OPfileID.write('i\t1\t' +str(item[0])+'\t'+ str(dur)+'\t'+str(amp)+'\t'+str(freq) + '\n')
# ------------end: for sonifying value of all trades at  timestamp -----------------------------------------
        OPfileID.write('e\n')
        OPfileID.close()
        print "FIN: wrote cSound score", OPfilename
# ::::::1:::::::::2:::::::::3:::::::::4:::::::::5:::::::::6:::::::::7:::::::::8:::::::::9:::::::::0:::::::::1:::::::::2

# ::::::::::::::::::::::  Major executables here ::::::::::::::::::::::::::::::::::::::::::::::::
print ":::::::::::::::::::::::"
SR=44100
```

```
analyseTrades()                         # main from raw daily files into securitiesDict{} & densityProfile{}

#printAnalysisFile("001.csv",11)        # or use, say, printAnalysisFile("001.csv")
#printDensityProfile("001denPlot.csv")
#printDensityProfile()
#print dictToSynthList(densityProfile)
#
#sonifyValues(densityProfile)           # use this for single total value at timestamp son.
#cSoundScore(densityProfile, 2,2,12, "001_020212.sco")
#
#cSoundScore(densityProfile, 64,2,7, "001_totVal.sco")
#secIDs=securitiesDict.keys()
#secIDs.sort()
#for id in secIDs:
#       print securitiesDict[id]
courseOfSalesDict = {}     # a dictionary of timestamp keys into a list of [secID,tradeID]s

"""
def courseOfSales():
        # Generates a Course of Sales dictionary for data in densityProfile{}
        timestamps=densityProfile.keys()
        timestamps.sort()
        for stamp in timestamps:
                # register the first trade price
                # check whether each trade is in the [price, volume] dictionary
                for each <TRADE> :
                if first trade of <securityUD>
                        Record price, nrUnits
                else (<TRADE> has already been recorded):
                        if price is same as previous, add nrUnits
                        else price is different, so sonify previous (value + gliss:
up if price rose
down if price dropped

"""
"""
working code to sort out the multiple trade at same bid/askIDs
timestamp=densityProfile.keys()
timestamp.sort()
for stamp in timestamp:
        bidIDlist=askIDlist=[]
#       if len(densityProfile[stamp])>1:  # there is more than one trade at that time
#               # NB no need to check for same securityID because different securityID's
                # will automatically have different bidID & askID
        for order in densityProfile[stamp]:
```

```
#               print order[2], order[3]
                bidIDlist+=[order[2]]
                askIDlist+=[order[3]]
        print "bids:", bidIDlist
        print ":::::::::"
        print askIDlist
        print "%%%%%%%%%%%"

        for bid in bidIDlist:
                if bidIDlist.count(bid)>1:
                        print "multiple trades with same bidID", bid
                else:
                        print "-----------------------only 1 trade for  bidID", bid

for key in densityProfile.keys():
        print key , densityProfile[key]
"""
# ::::::::::::::::::::::::::::::::::::::::::::::: End of CM_IDTradeAnalysis.py ::::::::::::::::::::::::::::::::::::::::::
# :::::::1:::::::::2:::::::::3:::::::::4:::::::::5:::::::::6:::::::::7:::::::::8:::::::::9:::::::::0:::::::::1:::::::::2
"""
; csound instrument for playing score produced by cSoundScore(adict)
sr = 44100
kr = 4410
ksmps = 10
nchnls = 1

instr 1
        ifreq = 2048
        p5=p5/4
        kmaxamp = 120 - (110*p5/ifreq)  ; p5 is frequency
        p3 = (p3*0.9) + (0.1*ifreq/p5)                  ; p3 is dur
        idur = 1/p3                             ; oscil freq = 1/dur
        kenv    linseg 0, 0.01,1, P3-.01, 0
;       kenv    oscil 1, idur, 2                ; use f2 (1st 1/2 sine
        asig    oscili kenv, p5, 1      ; use f1 (sin)
        out asig * kmaxamp
endin

; SIMPLE csound instrument for playing score produced by cSoundScore(adict)
sr = 44100
kr = 4410
ksmps = 10
nchnls = 1

instr 1
```

```
        kmaxamp = 100 - (100*p3/4096)  ; 4096 is max freq for 12 octaves on fund of 1 Hz
        ifreq = p5 ; * 440
        kenv    oscil 1, 1/p3, 2        ; use f2 (1/2 sine)
        asig    oscili kenv, ifreq, 1 ; use f1 (sin)
        out asig * kmaxamp
endin
;
; csound instrument for playing score produced by cSoundScore(adict)
sr = 44100
kr = 4410
ksmps = 10
nchnls = 2

instr 1
        kmaxamp = 100
        ifreq = p5 ; * 440
        iamp = 1
        iamp1= 0.68
        p3=p3*1.2

        idur1 = p3 * 0.38       ;2/5 dur
        idur2 = p3 * 0.5        ;1/2 dur
        idur3 = 1-idur1 ;3/5 dur

        kenv    linseg 0, p3*.25, 1, p3*.75, 0
        asig    oscili kenv, ifreq, 1 ; use f1 (sin)
        outs1 asig * kmaxamp
        outs2 asig * (1-kmaxamp)
endin
=========
"""
```

```python
# -*- coding: utf-8 -*-
# :::::::::::::::::::::::::::::::::::::::::::: CM65summaries.py :::::::::::::::::::::::::::::::::::::::::::::::::::::::
# ::::::1:::::::::2:::::::::3:::::::::4:::::::::5:::::::::6:::::::::7:::::::::8:::::::::9:::::::::0:::::::::1:::::::::2
# :set printoptions=header:0,portrait:n,number:y,left:2pc,right:2pc,top:25mm,bottom:25mm
#for options in vim use :help popt-option

"""
Summaries of the 65 dayfiles are stored in a pickle DB for faster referencing of HDF5 file
"""
import glob, string, cPickle

class MarketDay:
    """ Reads summary data for specified day from .csv file. Optional auto print of summary."""
    def __init__(self, day=0, fprint=False):
        self.day            = day
        self.IPdir          = "/Users/drw/Documents/R_DATA/CMdataAnalysis/"
        self.OPdir          = "/Users/drw/Documents/R_DATA/CMdataAnalysis/"
        self.dayFID         = None
        self.fileList       = glob.glob(self.IPdir+"*.csv")
        self.dayFileName    = None
#        self.dayList        = []+ daylist # this permits a single number 'n' in input rather than '[n]'
        self.dayData        = None  #self.fileToList([]
        self.closeValues    = []    # a list of (key, value) tuples for a day. See self.getValues()
        try:
            if self.day > 0:         # do it automatically
                self.dayToFilename()
                self.fileToList()
                if fprint:
                    self.printSummary()
        except:
            print "EXCEPTION:: No summary data available for day", self.day, "in", self.IPdir

    def stringToNumeric(self, val):
        """ Converts a string to a numeric: an integer if point is present, else a real
        If string cannot be converted to a numberic, input is returned unchanged.
        CHECK : what happens for longs? """
        try:
            res=string.atoi(val)
        except:
            try:
                res=string.atof(val)
            except:
                res=val
        return res
```

```python
# ::::::1:::::::::2:::::::::3:::::::::4:::::::::5:::::::::6:::::::::7:::::::::8:::::::::9:::::::::0:::::::::1:::::::::2
    def dayToFilename(self):
        dayString="IDTradeAnal"+("00"+str(self.day))[-3:]+".csv" # a 3 char numeric string in the form "001" to "065"
        self.dayFileName=self.IPdir+dayString

    def fileToList(self):
        """ Converts a numeric string to a number. Returns contents of file fname in a list """
        self.dayFID=open(self.dayFileName,'r')      # file = IPdirectory+IPfilename ie full path
        self.dayFID.seek(0,0)
        self.dayData = self.dayFID.readlines()  # Reads file into a list of lines
        self.dayFID.close()

    def stringToList(self, stringy):
        """ Converts a string to a list of numerics if possible, else strings."""
        res=[]
        tmp=string.split (stringy,",")
        for i in range (len(tmp)):
            tmp[i] = self.stringToNumeric(tmp[i]) # convert all numberic string to numerics
        res += tmp
        return res

    def printSummary(self, fieldsize=0):
        """ Prints each line of a file in a list R adjusted in a field fieldsize chars wide.
        fdata is a list of strings, each representing a line. NB list of strings, not ints etc.
        If fieldsize=0, the data is scanned to determine the maximum fieldsize necessary or each field
        and that fieldsze is used to  R adjust each chars wide column. """
        count = 0
        if fieldsize > 0:
            for line in self.dayData:
                print  ' '* (fieldsize-len(str(count))),  count,":",
                for y in string.split(line,','):
                    print  ' '* (fieldsize-len(y)), y,  # "\t",
                count += 1
        else:   # fieldsize=0 so calculate them variably
            fieldsizeList={}
            for line in self.dayData:                      # scan the lines
                lineaslist=string.split(line,',')
                for col in range(0,len(lineaslist)):     # find and store the max fieldsize
                    try:
                        if fieldsizeList[col] < len(lineaslist[col]):
                            fieldsizeList[col] =  len(lineaslist[col])
                    except:     # gets here when a column's fieldsize hasn't been stored b4
                        fieldsizeList[col] =  len(lineaslist[col])
            # make up the format string
            fmt = ""
```

```
            for col in fieldsizeList: fmt += ' %' + str(fieldsizeList[col]) + 's'
            # now do the print
            pad = len(str(len(self.dayData)))           # the char size of the max line len
            for line in self.dayData:
                print  ' '* (pad-len(str(count))),  count,":",
                print fmt  % tuple(string.split(line,',')),
                count += 1

#d1=DaySummary(1 fprint=True)

class MarketSummary(MarketDay):
    def __init__(self, dazeList=[]):
        MarketDay.__init__(self)
        self.dayList =[]
        if dazeList:
            self.dayList = dazeList
#        self.day = self.dayList[0]
        self.securitiesDict     = {}    # key is securityID, data is list of day summaries
        self.summaryDict        = {}    # summaries of the trading data of a securities in self.securitiesDict
        if not self.dayList:
            self.dayList = [ i for i in xrange(1,66)]   # a bit crude. better to get them from the filelist
            print "Making a dictionary of all securities in all files....."
        self.doDays()
        self.summariseSecurityData()

    def getDayData(self, dayNr):
        # Gets one day's data from file and adds it to self.securitiesDict as a list.
        # Each security is thus a list of lists.
        try:
            self.day = dayNr
            if self.day > 0:                                   # do it automatically
                self.dayToFilename()
                self.fileToList()                              # data is now in self.dayData
            else:
                print "EXCEPTION:: No summary data available for day", self.day, "in", self.IPdir
                return
            newDayData = []
            for line in self.dayData:
                newline=self.stringToList(line)
                newDayData.append([newline[0]]+newline[3:])
            del newDayData[0]                                  # toss the headings away
            for sec in newDayData:
                if sec[0] in self.securitiesDict.keys():
                    self.securitiesDict[sec[0]].append(sec[1:])
                else:
```

```
                    self.securitiesDict[sec[0]] = [sec[1:]]
#               if sec[0] == '-':            # NB This was a test to discover that on day65 one secID ='-'
#                   print "Day:", self.day, ":", line  # drw changed it to (unique) '99999' (in the analysis file only)
        except:
            print "EXCEPTION:: Can't add", dayNr, "to the dictionary."

# :::::::1:::::::::2:::::::::3:::::::::4:::::::::5:::::::::6:::::::::7:::::::::8:::::::::9:::::::::0:::::::::1:::::::::2
    def doDays(self):
        for day in self.dayList:
            self.getDayData(day)

    def summariseSecurityData(self):
        # Summarises the day data for each security trading in self.dayList
        print "Summarising securitiesDict data...."
        if not self.securitiesDict:     # if the securitiesDict has not been generated
            for d in self.dayList:      # generate it
                self.getDayData(d)      # adds a list for each day to the self.securitiesDict
        secs = self.securitiesDict.keys()
#       # Make self.summaryDict                         0  1  2  3   4     5       6       7
#                                                       O, H, L, C, Vol, #Trades, $Value, NrDays
        for sec in self.securitiesDict.keys():
            secSummary = [0]*8
            for d in xrange(len(self.securitiesDict[sec])):
                secSummary[7]+=1                                # increment Nr of days summarised
                secSummary[3] = self.securitiesDict[sec][d][3]      # new Close price
                secSummary[4] += self.securitiesDict[sec][d][4]     # increment trading Volume
                secSummary[5] += self.securitiesDict[sec][d][5]     # increment #trades
                secSummary[6] += self.securitiesDict[sec][d][6]     # increment $Value
                if secSummary[0] == 0:                              # if this is the first day
                    secSummary[0] = self.securitiesDict[sec][d][0]  # Open price
                    secSummary[2] = self.securitiesDict[sec][d][0]  # initial Low price
                if self.securitiesDict[sec][d][1] > secSummary[1]:
                    secSummary[1] = self.securitiesDict[sec][d][1]  # new High price
                if self.securitiesDict[sec][d][2] < secSummary[2]:
                    secSummary[2] = self.securitiesDict[sec][d][2]  # new Low price
            self.summaryDict[sec]=secSummary                        # add it to the summary dictionary

    def sortByValue(self, dict):
        val = [(dict.itervalues(), values[6]) for values in dict.itervalues()]
        return val

    def dictColSort(self, dictionary, col=0, descend=False ):
        """ returns a list of (value, key) tuples of sorted col column of dictionary values, ascending by default
        # example:  # s          = MarketSummary()
                    # valHL      = s.dictColSort(s.summaryDict, 6, descend=True)
```

```
                    # top100keys = valHL[:100][1]    """
        res=[( v[col], k) for k, v in dictionary.iteritems()]   # NB (value, key) not (key, value)
        res.sort(reverse=descend)
        return res

# ::::::::1:::::::::2:::::::::3:::::::::4:::::::::5:::::::::6:::::::::7:::::::::8:::::::::9:::::::::0:::::::::1:::::::::2
def doit():
    s = MarketSummary()
    kees=s.summaryDict.keys()
    OPfileID = open(s.OPdir+"summaryDict_all.pickle", 'w')
    cPickle.dump (s.summaryDict, OPfileID)                      # pickles a 65 day summary of all securities
    OPfileID.close()
    keysSortedByValue = s.dictColSort(s.summaryDict, col=6, descend=True)
    keysSortedByPrice = s.dictColSort(s.summaryDict, col=3, descend=True)   # uses Closing Price
    keysSortedByVolume = s.dictColSort(s.summaryDict, col=5, descend=True)
    OPfileID = open(s.OPdir+"summary65sortedLists.pickle", 'w')
    cPickle.dump (keysSortedByValue, OPfileID)  # pickles list of (value, secID) tupes  (sorted by value over 65 days)
    cPickle.dump (keysSortedByPrice, OPfileID)  # pickles list of (price, secID) tupes  (sorted by price over 65 days)
    cPickle.dump (keysSortedByVolume, OPfileID) # pickles list of (volume, secID) tupes (sorted by vol over 65 days)
    OPfileID.close()

def getTopPriceSecs(n=1):
    IPdir       = "/Users/drw/Documents/R_DATA/CMdataAnalysis/"
    filename    = IPdir+"summary65sortedLists.pickle"
    IPfileID = open(filename, 'r')
    # how do we get the 2nd item without reading the first? IPfileID.next() doesn't work.
    listy=cPickle.load(IPfileID)     # how do we get the 2nd pickled item without reading the first?
#    listy=cPickle.load(IPfileID)    # IPfileID.next() doesn't work.
#    listy=cPickle.load(IPfileID)    # IPfileID.next() doesn't work.
    lst = []
    for i in xrange(n):
        lst.append(listy[i][1])     # the 2nd of the tuple is the secID
    IPfileID.close()
    return lst
#------------------EOF - FIN --------- pickled onions 4T!
```

```python
# -*- coding: utf-8 -*-
# :::::::::::::::::::::::::::::::::::::::::::: getH5security.py :::::::::::::::::::::::::::::::::::::::::::::::::::::::::
# ::::::1::::::::2::::::::3::::::::4::::::::5::::::::6::::::::7::::::::8::::::::9::::::::0::::::::1::::::::2
#:set printoptions=header:0,portrait:n,number:y,left:2pc,right:2pc,top:25mm,bottom:25mm
#for options in vim use :help popt¿option

# class definitions for fetching, analysing and plotting of a given securityID from an HDF5 file.
# NB python days are zero-based. h5 days are 1-based. All days are by default python days. h5 days are called 'h5days'
# last edit: drw 20071011

from CMpyTables import *    # HDF5 classes for the CM data
from CMflags import *       # used to compress the <flags> field in order book entries into a single 64bit mask
import glob, string, time, datetime, os, tables, cPickle
import numpy
import pylab                # matplotloib routines for plotting


class h5security (H5file):              # inherits the h5file class (defined in CMpyTables.py)
    def __init__(self, filename):
        self.secID      = ''        # the Id (name) of the security data to be instanced
        self.fileName   = filename  # the name string of the H5 file. It's usual for it to have a .h5 extension.
        self.h5fileError    = "Unable to access HDF5 file "
        self.shortFileName  = self.fileName[string.rfind(self.fileName,"/")+1:] # just filename, not the path
        self.fileID = None          # returned by the opening definitions. Used to access the file.
        self.mode = None            # the mode of the the file opening. 'w' = write, 'a' = append, 'r' = read
                                    # Internal use only. See getFlagNumber().
        self.securitiesDict = {}    # A dict of tuples for each Node/Group in the root directory.
                                    # each tuple is in the form (CMgroupID, firstDay, lastDay,  ....)
                                    # should this Dict be held in RAM and written to the h5 file before close?
        self.h5data      = []       # into which h5 file data is extracted, one list per order
        self.h5dataBuff1 = []       # into which h5 file data can be transferred for temporary or analytic purposes
        self.h5dataBuff2 = []       # into which h5 file data can be transferred for temporary or analytic purposes
        self.dayRange    = (1,66)   # range of days requested/required      - default values are auto assigned
        self.hourRange   = (0,24)   # range of hours requested/required     - ditto
        self.minuteRange = (0,60)   # range of minutes requested/required   - ditto
        self.secondRange = (0,60)   # range of seconds requested/required   - ditto
        self.timePeriod  = (0,0)    # (min,max) of time period required, in seconds from midnight
        self.nrDays      = 0        # set to self.dayRange[1] - self.dayRange[0].
                                    # Indexes of the 1st of 3 dimensions of the self.bins array

        self.dayDuration =  0       # Nr of secs of a day's data to be analysied.
                                    # Set to self.timePeriod[1] - self.timePeriod[0]. Used for squeezing data to bins
        self.nrBins      = 100      # the number of bins into which each day data is to be sorted   - ditto
        self.bins        = None     # a numpy array of data bins
        self.breakpoints = None     # placeholder for a numpy1Darray of self.nrBins break points (in secs)
```

```python
                                    # into which orders are binned.
        self.nrVariates  = 1        # the nr of independ. vars to be extracted from h5data or +ed for interum results
#        self.barData     = []      # a list of day lists of summary data for bar/histogram plotting
        self._getAllSecurityNodes() # open the bugger!

# ::::::1::::::::2::::::::3::::::::4::::::::5::::::::6::::::::7Y::::::::8::::::::9::::::::0::::::::1::::::::2
    def _getAllSecurityNodes(self):
        """makes a dictionary of all the securities in the h5 file.
        Nominally a private method/function because our focus is on single securties."""
        try:
            self.rOpen()            # open the h5 file for reading.  inherited from h5file class.
        except: print self.h5fileError
        for sec in self.fileID.getNode("/"):
            self.securitiesDict[sec._v_name] = []           # make the string of the secID a key in the securitiesDict
#            print "sub groups:-----------------:"
            for subgp in self.fileID.getNode("/"+sec._v_name):
                self.securitiesDict[sec._v_name] += [subgp]     # add to the list of group name strings
        print self.securitiesDict.keys()

    def _securityExistsQ(self):
        res=False
        if self.secID in self.securitiesDict.keys():
            return True
        else:
            print self.secID, "not available. Securitites available are:"
            print self.securitiesDict.keys()
            return False

    def setTimePeriods (self, days = (1,66), hours = (0,24), minutes = (0,60), seconds = (0,60)):
        """Time data is the time period from which data is to be pulled from the h5 file for this security.
        Transfers the time data to class variables; fixing common range error if in evidence."""
        self.timePeriod = ( hours[0]*3600+minutes[0]*60+seconds[0],\
                            hours[1]*3600+minutes[1]*60+seconds[1]) # (min_sec, max_sec)
        if  days[0] < 1: days[0] = 1                    # just in case. The h5 day is 1-based.
        if days[1] == days[0]: days[1] = days[0]+1      # ditto ASSIGNMENT DON'T WORK - days is a () i.e. immutable
        self.dayRange = days
        if hours[1] == hours [0]: hours = (hours[0],hours[0]+1) ; self.hourRange = hours
        if minutes[1] == minutes [0]: minutes = (minutes[0],minutes[0]+1) ; self.minuteRange = minutes
        if seconds[1] == seconds [0]: seconds = (seconds[0], seconds[0]+1) ; self.secondRange = seconds
        self.nrDays = self.dayRange[1] - self.dayRange[0]           # for indexing 1st of 3 Ds of the array
        self.dayDuration =  self.timePeriod[1] - self.timePeriod[0] # nr of secs of a day's data to be analysed.

    def getOnTradeData (self, secID):
        """Pulls the relevant leaf data from the h5file secID branch & stores it in self.bins according to time data.
        Assumes self.bins has been appropriately made to accomodate the data. See self.makeBins()
```

```
        If multiple days, assumes all days are for the same time period."""
        self.secID = secID
        if self._securityExistsQ():                             # fetch requested trade data from h5 file
            nodey = "/" + self.secID + "/OnTrade/OnTradeTable"
            print "getOnTrades: ", nodey
            tab = self.fileID.getNode(nodey)
            print "getOnTrades: tradeTable columns:" , tab.colnames
            print "getOnTrades: MaxNrrows available:",tab.nrows,"\tdays:",self.dayRange ,"\ttime range:",self.timePeriod

            for row in tab.iterrows():
                if int(row['day']) in range(*self.dayRange):
                    if int(row['time']) in range (*self.timePeriod):                                # PULL THE h5 DATA
                        rowsie = [ row['day'], row['time'], row['secPrice'], row['secVolume'] ]
#                        print rowsie
                        self.h5data += [rowsie]                                         # NB the 0th item is for self.dayRange[0]
#                    else:
#                        break   # assume we can do this ... because a day's timestamps stored incrementally
#                else:
#                    break   # assume we can do this ... because day numbers are stored incrementally?
        else:
            print "getOnTrades: Can't get OnTrades for", self.secID
            print "getOnTrades: Securities in open file:", self.securitiesDict.keys()

# ::::::1:::::::::2:::::::::3:::::::::4:::::::::5:::::::::6:::::::::7:::::::::8:::::::::9:::::::::0:::::::::1:::::::::2
    def getAskData (self, secID) :
        """ Pulls the relevent (leaf) data from the h5file. A leaf is the data table at the end of a node.
        If multiple days, assumes all days are for the same time period."""
        self.secID = secID
        if self._securityExistsQ():                             # fetch requested trade data from h5 file
            nodey = "/" + self.secID + "/Ask/AskTable"
            print "getAskOrders: ", nodey
            tab = self.fileID.getNode(nodey)
            print "getAskOrders: AskTable columns:" , tab.colnames
            print "getAskOrders: NrRowsOfData:",tab.nrows,"\tdays:",self.dayRange ,"\ttime range:",self.timePeriod
            for row in tab.iterrows():
                if int(row['day']) in range(self.dayRange[0]+1,  self.dayRange[1]+1):       # because day0 == h5day1
                    if int(row['time']) in range (*self.timePeriod):                                # PULL THE h5 DATA
                        self.h5data += [[ row['day'],row['time'],row['secPrice'],row['secVolume'], row['flagsMask'] ]]
        else:
            print "getAskOrders: Can't get Ask Orders for", self.secID
            print "getAskOrders: Securities in open file:", self.securitiesDict.keys()

    def getBidData (self, secID):
        """ Pulls the relevent (leaf) data from the h5file. A leaf is the data table at the end of a node.
        If multiple days, assumes all days are for the same time period."""
```

```
        self.secID = secID
        if self._securityExistsQ():                             # fetch requested trade data from h5 file
            nodey = "/" + self.secID + "/Bid/BidTable"
            print "getBidOrders: ", nodey
            tab = self.fileID.getNode(nodey)
            print "getBidOrders: BidTable columns:" , tab.colnames
            print "getBidOrders: Nr rows of data:",tab.nrows,"\tdays range:",self.dayRange,"\ttime range:",self.timePeriod
            for row in tab.iterrows():
                if int(row['day']) in range(self.dayRange[0]+1,  self.dayRange[1]+1):    # because day0 == h5day1
                    if int(row['time']) in range (*self.timePeriod):                             # PULL THE h5 DATA
                        self.h5data += [[ row['day'], row['time'], row['secPrice'], row['secVolume'], row['flagsMask'] ]]
        else:
            print "getBidOrders: Can't get Bid Orders for", self.secID
            print "getBidOrders: Securities in open file:", self.securitiesDict.keys()

# ::::::1:::::::::2:::::::::3:::::::::4:::::::::5:::::::::6:::::::::7:::::::::8:::::::::9:::::::::0:::::::::1:::::::::2
    def makeBins(self, nrBins=100, nrVariates= 2):
        """Makes an list lists: self.nrDays X nrBins X nrVariates+1.
        self.bins is in the form [day [breakpoint [variate1, variate2, ...]]] """
        self.nrBins     = nrBins        # the number of bins (rows) into which each day data is to be sorted
        self.nrVariates = nrVariates    # the number of variable data (columns) to be analysed, or needed for analysis
        self.breakpoints = numpy.arange(nrBins, dtype=numpy.int16)  # a 1D-array of self.nrBins break points (in secs)
                                                                    # into which orders are binned.
        self.breakpoints = numpy.multiply(self.breakpoints, self.dayDuration/self.nrBins)
                # each elt in self.breakpoints in now a rel start time of the bin NB this arg has to be a list or tuple
        self.breakpoints = numpy.add(self.breakpoints, [self.timePeriod[0]])
                                         # each elt in self.breakpoints in now an absolute start time of the bin
        self.bins = []                   # a list of day lists of summary data for bar/histogram plotting
        for day in range(*self.dayRange):      #  make a list of day lists of breakpoints. The day Nr is not important.
            self.bins.append([nrVariates*[0,] for i in xrange(len(self.breakpoints))])

    def dataToBins (self):
        """Massages the H5 data in self.hData into nrBin time clumps, aggregating values in the variates.
         Transfers the timer data to class variables; fixing common range error if in evidence.
        The type and amount of data is set in getH5securityData()."""
        if len(self.h5data)==0:
            print "dataToBins: self.h5data array is empty."
            print "Fill using self.getOnTradeData() etc before calling self.dataToBins()"
            return 0
        for order in self.h5data:                               # find the right bin Nr based on breakpoint time
            binNr = 0                                           # temp var to hold which bin the order is in
            for i in range (len(self.breakpoints)):             # find the appropriate time bin for this order
                if order[1] >= self.breakpoints[binNr+1]:       # if order time >= bin breakpoint time
                    binNr += 1                                  # increment the binNr
            try:
```

```
                            # array index is -vely offset from the min day number
                dayIndex=order[0]-self.dayRange[0]-1        # min day index is 0 despite which actual day it is.
                self.bins[dayIndex][binNr][-2] += order[-1] * order[-2]    # ++ the value (price X vol)
                self.bins[dayIndex][binNr][-1] += 1  # ++ the number of trades in this bin. res index is 1 < dayNr
            except:
                print "order:", order, "binNr:", binNr, order[0]
                return 0

# :::::::1:::::::::2:::::::::3:::::::::4:::::::::5:::::::::6:::::::::7:::::::::8:::::::::9:::::::::0:::::::::1:::::::::2
    def _plotLayoutCodes(self, nrCols=5):
        """ returns a list of location codes for nrplots in nrCols columns."""
        nrPlots=len(range(*self.dayRange))
        codes=[]           # a list of tuples, 1 for each plot e.g. (5,2,3) is the 3rd plot in a 5x2 subplot drawing
        nrRows= divmod(nrPlots+nrCols-1, nrCols)[0]   # the old rounding trick!
        stem=(nrRows,nrCols)
        for plotNr in range(nrPlots):
            codes.append(stem+(plotNr+1,))
        return codes

    def getMaxArrayVal(self, array, index):
        """Returns the maximum value of the array."""
        maxie = 0
        for day in array:
            for bin in day:
                if bin[index] > maxie:
                    maxie = bin[index]
        return maxie

    def drawBinBars(self, index=-2):
        """Draws bar charts of value data in self.bins, which is a list of day lists, each of nrBins divisions"""
        codes=self._plotLayoutCodes(5)     # in weekly (5-day) columns
#        print "plot layout codes:", codes
        maxVal = self.getMaxArrayVal(self.bins, 0)
        print "drawValueBars: max array value = ", maxVal
        for day in range(*self.dayRange):
            try:
                print "drawValueBars: processing data for h5day", day
                pylab.subplot(*codes[day-self.dayRange[0]])
                                        # eg: int('5'+'10')+1 = 510+2 = 512 ie 2nd plot in 5 x 1 array of plots

                for i in range(self.nrBins):
                    pylab.bar (i, self.bins[day-self.dayRange[0]][i][index], color='#22cc55')   #[-2]= $value
                pylab.axis([-10, 100, 0, maxVal])               # NB this must come after the previous line
                titl = 'Day '+repr(day)                         # display h5day
                pylab.title (titl, fontsize='small')
                if (day-self.dayRange[0]) % 5 == 0:
                    pylab.ylabel ('$value', fontsize='small')
#                pylab.xticks(pylab.arange(1,100,1), xt )
#                pylab.xlabel(titl, fontsize='small')
#                pylab.xlabel(' 100 periods', fontsize='small')
                pylab.grid (True)
            except:
                print "drawBinBars: Problem making plot data for a h5day", day+1
                return 0
#        pylab.title('A tale of 2 subplots')
        heading = "On-Market trades for security " + self.secID
        pylab.title(heading, bbox={'facecolor':'0.8', 'pad':5})
#        pylab.plot("Value of Trades in equal time daily periods")
        pylab.show()

fname = "/Users/drw/Documents/R_DATA/CM65top20.h5"
s40 = h5security(fname)
s40.setTimePeriods (days = (62,65), hours = (10,16), minutes = (0,29), seconds = (0,60)) #
s40.getOnTradeData('s40') # raw data is stored in the self.h5data list
s40.makeBins(nrBins=100, nrVariates= 2)
s40.dataToBins()
s40.drawBinBars(-2)
#for item in s40.bins:   print item

print "closing h5file" ; s40.close()    # inherited from h5file class defn

#s40.h5dataBuff1 = s40.h5data[:] # copy the data over
#print "s40.h5dataBuff1", len(s40.h5dataBuff1)
#s40.h5data=[]
#s40.dataToBins(nrBins=100, nrVariates=3)     # v1 = self.nrDays, v2 = self.nrBins, v3 = nrTrades
#s40.drawValueBars((0,10))

"""
seccy = h5security(fname)                             # instantiate for securityID nr 20
seccy.setTimePeriods (days = (0,65), hours = (10,16), minutes = (0,29), seconds = (0,60)) #
for sec in seccy.securitiesDict.keys():
    acount, bcount = 0, 0
    seccy.getBidOrders(sec) # get raw data from the self.h5data list
    for b in seccy.h5data:
        if flagInMaskQ('t', b[4]):
            print "B::", b
            bcount += 1
#        printFlagsInMask(a[4])

    seccy.h5data=[]
```

```
    seccy.getAskOrders(sec)
    for a in seccy.h5data:
        if flagInMaskQ('t', a[4]):
            print "A::", a
            acount += 1
#        printFlagsInMask(a[4])
seccy.close()
"""
"""
acount = 0
for a in s20.h5data:
    if flagInMaskQ('t', a[4]):
#        print "A::", a
#        printFlagsInMask(a[4])
        acount += 1
print "-----------------------------"
bcount = 0
for b in s20.h5dataBuff1:
    if flagInMaskQ('t', b[4]):
#        print "B::", b
#        printFlagsInMask(b[4])
        bcount += 1
print "NrAsks::", acount,  "NrBids::", bcount,

    def _securityExistsQ(self):
        res=False
        if self.secID not in self.securitiesDict.keys():
            if len (self.securitiesDict.keys()) > 0:
                print secID, "not in h5 file."
#                return 0
        else:
            try:
                print " Loading security Names from ", self.shortFileName
                self._getAllSecurityNodes()
                res = True
            except self.h5fileError, self.fileName:
                pass
#                    return 0
        return res
"""
```

GLOSSARY

> For every complex problem there is always a simple solution–and it's always wrong. H.L. Mencken

a posteriori. A way of gaining *knowledge* by appealing to some particular *experience(s)*. This method is used to establish *empirical* and *hypothetical* truths. (Cf. *a priori*.) [IK]

a priori. A way of gaining *knowledge* without appealing to any particular *experience(s)*. This method is used to establish *transcendental* and *logical* truths. (Cf. *a posteriori*.) [IK]

aesthetic. Having to do with sense-perception. In Kant's first *Critique* this word refers to *space and time* as the necessary conditions for sense-perception. The first half of the third *Critique* examines the *subjective* purposiveness in our perception of beautiful or sublime *objects* in order to construct a *system* of aesthetic *judgment*. (Cf. *teleological*.) [IK]

analysis. Division of a *representation* into two opposing representations, with a view towards *clarifying* the original representation. Philosophy as *metaphysics* employs analysis more than *synthesis*. (Cf. *synthesis*.) [IK]

analytic. A statement or an item of *knowledge* which is true solely because of its conformity to some *logical* laws. (Cf. *synthetic*.) [IK]

appearance. An *object* of *experience*, when viewed from the *transcendental perspective*. Though often used as a synonym for *phenomenon*, it technically refers to an object considered to be conditioned by *space and time*, but not by the *categories*. (Cf. *thing in itself*.) [IK]

apperception. A term used by Leibniz for conscious perception, while he calls 'unconscious perceptions', 'minute perceptions' or just 'perceptions.' Leibnitz, G.W. 1704. New Essays on Human Understanding. Edited and Translated P. Remnant and J. Bennett. Cambridge: Cambridge University Press, 1999).

architectonic. The *logical* structure given by *reason* (especially through the use of twofold and threefold divisions), which the philosopher should use as a plan to organize the contents of any *system*. [IK]

ASX Australian Securities Exchange. http://www.asx.com.au/

audiation. n. The mental review of sonic experiences with an auditory display [AD pp 188]. (C.f. ideation. n. the power of the mind for forming ideas [Ch]). A coined term.

audient adj. listening. Paying attention. n. a hearer [Ch].

audification. n. The direct playback of data samples [AD pp xxvii]. The direct conversion of data to sound [AD pp 190]. (C.f. sonification.) A coined term.

audile adj. Pertaining to hearing. n. one inclined to think in terms of sound. [Ch].(C.f. visile.)

audio. n. Reproduction of recorded or broadcast sounds (also adj) [Ch].

audiolisation. n. see Auralisation. =A coined term.

audition. n. The sense, or an act, of hearing [Ch].

auditive. adj. Of, or related to, hearing [Ch].

auditory icon. n. A mapping of computer events and attributes to the events and attributes that normally make sounds...In general, the result is to relate interface sounds to their referents in the same way that natural sounds are related to their sources and, thus, to allow people to use their existing everyday listening skills in listening to computers [Gaver, AD pp 420].

auditory. adj. Relating to the sense of hearing.

auditory information design. The design of sounds to support an information processing activity. [SB: 30]

aural. adj. Pertaining to the ear. adv. aurally [Ch].

auralisation. n. The auditory representation or "imaging" of data [AD pp xxvii]. The representation of program data using sound...an auralisation is based on the actual execution data of the program [Jackson, AD pp 292]. (C.f. sonification.) A coined term.

autonomy. An action which is determined by the *subject's* own free choice (see *will*). In the second *Critique*, moral action is defined as being autonomous. (Cf. *heteronomy*.) [IK]

autophenomenology. see hetrophenomenology.

categorical imperative. A command which expresses a general, unavoid-able requirement of the *moral law*. Its three forms express the requirements of universalisability, respect and autonomy. Together they establish that an action is properly called 'morally good' only if (1) we can will all persons to do it, (2) it enables us to treat other persons as ends and not merely as the means to our own selfish ends, and (3) it allows us to see other persons as mutual law-makers in an ideal 'realm of ends'. [IK]

categories. The most general *concepts*, in terms of which every *object* must be viewed in order for it to become an object of *empirical knowledge*. The four main categories (quantity, quality, relation and modality) each have three sub-categories, forming a typical example of a twelvefold, *architectonic* pattern. (Cf. *space and time*.) [IK]

concept. The active species of *representation*, by means of which our *understanding* enables us to think. By requiring perceptions to conform to the *categories*, concepts serve as 'rules' allowing us to perceive general relations between representations. (Cf. *intuition*.) [IK]

conscience. The *faculty* of the human *subject* which enforces the *moral law* in a particular way for each individual by providing an awareness of what is right and wrong in each situation.

constitutive. Playing a fundamental role in making up some type of *knowledge*. (Cf. *regulative*.) [IK]

Copernican revolution. In astronomy, the theory that the earth revolves around the sun; in philosophy, the (analogous) theory that the *subject* of *knowledge* does not remain at rest, but revolves around (i.e., actively determines certain aspects of) the *object*. Thus, the *formal* characteristics of the *empirical* world (i.e., *space and time* and the *categories*) are there only because the *subject's* mind puts them there, *transcendentally*.

CPR. Kant's monograph, *The Critique of Pure Reason*. Within this glossary this work is also referred to simply as the *Critique*.

critical. Kant's lifelong approach to philosophy which distinguishes be-tween different *perspectives* and then uses such distinctions to settle otherwise unresolvable disputes. The Critical approach is not primarily negative, but is an attempt to *adjudicate* quarrels by showing the ways in which both sides have a measure of validity, once their perspective is properly understood. Kant's *system* of Critical philosophy emphasizes the importance of examining the structure and limitations of *reason* itself.

critique. To use the method of *synthesis* together with a *critical* approach to doing philosophy. This term appears in the titles of the three main books in Kant's Critical philosophy, which adopt the *theoretical*, *practical* and *judicial standpoints*, respectively. The purpose of Critical philosophy is to prepare a secure foundation for *metaphysics*. (Cf. *metaphysics*.) [IK] See also *CPR*.

data-controlled sound [AD pp xxvii]: Processes that disrupt the relationships of successive samples in favour of simplifying and enhancing features of the data, such as multiplying the data by a cosine wave, would be classified as sonification [AD pp 190]. A coined term.

disposition. The tendency a person has at a given point in time to act in one way or another (i.e., to obey the *moral law* or to disobey it). (Cf. *predisposition*.) [IK]

domain expert. Someone with knowledge in a particular field of endeavour. In sonification, a domain expert is the person with knowledge of the field from which the data is collected and for whom the sonification is attempting to assist. [DW]

earcon. n. Tone or sequence of tones as a basis for building messages [Blattner, AD pp 450]. a nonverbal audio message used in the user-computer interface to provide information to the user about some computer object, operation, or interaction. the aural counterpart of an icon [Blattner et al. Earcons and Icons: Their Structure and Common Design Principles, 1989]. (C.f. auditory icon.) A coined term..

empirical. One of Kant's four main *perspectives*, aiming to establish a kind of *knowledge* which is both *synthetic* and *a posteriori*. Most of the knowledge we gain through ordinary *experience*, or through science, is empirical. 'This swn is black" is a typical empirical statement. (Cf. *transcendental*). [IK]

empiricism., A practical rather than abstract epistemology that asserts that knowledge arises from experience rather than revelation. It emphasizes the role of experience and evidence, especially sensory perception, in the formation of ideas, while discounting the notion of innate ideas. In the philosophy of science, empiricism emphasizes those aspects of scientific knowledge that are closely related to evidence, especially as discovered in experiments. It is a fundamental part of the scientific method that all hypotheses and theories must be tested against observations of the natural world, rather than resting solely on a priori reasoning, intuition, or revelation. Hence, science is considered to be methodologically empirical in nature. [WP][DW]

experience. The combination of an *intuition* with a *concept* in the form of a *judgment*. 'Experience' in this 'mediate' sense is a synonym for '*empirical knowledge*'. The phrase 'possible experience' refers to a *representation* that is presented to our *sensibility* through *intuition*, but is not yet known, because it has not been presented to our *understanding* through *concepts*. 'Experience' in this sense is 'immediate' and contrasts with 'knowledge'. [IK]

faculty. A fundamental power of human *subjects* to do something or perform some *rational* function. [IK]

faith. A rational attitude towards a potential *object* of *knowledge* which arises when we are *subjectively* certain it is true even though we are unable to gain *theo-retical* or *objective* certainty. By contrast, knowledge implies objective and subjective certainty, while opinion is the state of having neither objective nor subjective certainty. Kant encouraged a more humble approach to philosophy by claiming to deny knowledge in order to make room for faith-i.e., by distinguishing between what we can know *empirically* and what is *transcendent*, which we can approach only by means of *faith*. [IK]

formal. The active or *subjective* aspect of something-that is, the aspect which is based on the *rational* activity of the *subject*. (Cf. *material*.) [IK]

heteronomy. An action which is determined by some outside influence (i.e., some force other than the freedom given by *practical reason*, such as *inclina-tion*) impelling the *subject* to act in a certain way. Such action is nonmoral (i.e., neither moral nor immoral). (Cf. *autonomy*.) [IK]

HCI Human Computer Interaction.

heterophenomenology. "phenomenology of another not oneself". A term coined by Daniel Dennett to describe an explicitly third-person, scientific approach to the study of consciousness. It consists of applying the scientific method with an anthropological bent, combining the subject's self-reports with all other available evidence to determine their mental state. The goal is to discover how the subject sees the world and themselves, without taking the accuracy of the view for granted. [WP] (Cf. phenomenology)

hypothetical. One of Kant's four main *perspectives*, aiming to establish a kind of *knowledge* which is both *analytic* and *a posteriori* (though Kant himself wrongly identified it as *synthetic* and *a priori*). Most metaphysical knowledge is properly viewed from this perspective, instead of from the *speculative* perspective of traditional *metaphysics*. (Cf. *logical*). [IK]

ideas. The species of *representation* which gives rise to metaphysical beliefs. Ideas are special *concepts* that arise out of our *knowledge* of the *empirical* world, yet seem to point beyond nature to some *transcendent* realm. The three most important metaphysical ideas are God, freedom and immortality. [IK]

imagination. The *faculty* responsible for forming *concepts* out of the 'manifold of *intuition*' and for synthesizing intuitions with concepts to form *objects* which are ready to be *judged*. [IK]

inclination. The *faculty* or *object* which motivates a person to act in a *heteronomous* way. Following inclinations is neither morally good nor morally bad, except when doing so directly prevents a person from acting according to *duty*-i.e., only when choosing to obey an inclination results in disobedience to the *moral law*. [IK]

information. A recognised relation between datum. a difference which makes a difference. [GB 1972]. The answer to a question. [JB 1981]

intelligible. Presented to the *subject* without any *material* being provided by *sensibility*. It is more or less equivalent to the terms *supersensible* and *transcendent*. (Cf. *sensible*.) [IK]

intuition. The passive species of *representation*, by means of which our *sen-sibility* enables to have sensations. By requiring *appearances* to be given in *space and time*, intuitions allow us to perceive particular relations between representations, thereby limiting *empirical knowledge* to the *sensible* realm. (Cf. *concept*.) [IK]

judgment. In the first *Critique*, the use of the *understanding* by which an *object* is determined to be *empirically* real, through a *synthesis* of *intuitions* and *concepts*. The third *Critique* examines the form of our feelings of pleasure and displeasure in order to construct a *system* based on the *faculty* of judgment (= the *judicial standpoint*) in its *aesthetic* and *teleological* manifestations. (Cf. *reason*.) [IK] judicial. One of Kant's three main *standpoints*, relating primarily to *experience*-i.e., to what we feel, as opposed to what we know or desire to do. Judicial *reason* is virtually synonymous with '*Critique*' itself, and is concerned with questions about the most profound ways in which we *experience* the world. Finding the source of two examples of such experiences is the task of the third *Critique*. (Cf. *theoretical* and *practical*.) [IK]

knowledge. The final goal of the *understanding* in combining *intuitions* and *concepts*. If they are *pure*, the knowledge will be *transcendental*; if they are impure, the knowledge will be *empirical*. In a looser sense, 'knowledge' also refers to that which arises out adopting any legitimate *perspective*.

logical. One of Kant's four main *perspectives*, aiming to establish a kind of *knowledge* which is both *analytic* and *a priori*. Hence it is concerned with nothing but the relationships between *concepts*. The law of noncontradiction (A is not -A) is the fundamental law of traditional, Aristotelian logic. (If we call this 'analytic' logic, then '*synthetic*' logic would be based on the oppo-site law of 'contradiction' [A is -A].) 'All bachelors are unmarried' is a typical logical statement. (Cf. *hypothetical*.) [IK]

material. The passive or *objective* aspect of something-that is, the aspect which is based on the *experience* a *subject* has, or on the *objects* given in such an experience. (Cf. *formal*.) [IK]

maxim. The *material* rule or principle used to guide a person in a particular situation about what to do (e.g., 'I should never tell a lie'). It thus provides a kind of bridge between a persons inner *disposition* and outer actions.

metaphysics. The highest form of philosophy, which attempts to gain knowledge of the *ideas*. Because the traditional, *speculative perspective* fails to succeed in this task, Kant suggests a new, *hypothetical* perspective for metaphysics. Metaphysics can succeed only when it is preceded by *Critique*. (Cf. *Critique*.) [IK]

noemata. The phenomena at which noesis it is directed. (Husserl) [WP]

noematic. The object or content (noema) which appears in the noetic acts (respectively the believed, wanted, hated and loved ...). (Husserl) [WP]

noesis. An act of consciousness. (Husserl) [WP]. The psychological result of perception and learning and reasoning [syn: cognition, knowledge]

noetic. The intentional act of consciousness (believing, willing, etc.) (Husserl) [WP]

noumenon. The name given to a thing when it is viewed as a *transcendent object*. The term 'negative noumenon' refers only to the recognition of some-thing which is *not* an object of *sensible intuition*, while 'positive noumenon' refers to the (quite mistaken) attempt to know such a thing as an *empirical* object. These two terms are sometimes used loosely as synonyms for '*transcendental object*' and '*thing in itself*', respectively. (Cf. *phenomenon*.) [IK]

object. A general term for any 'thing' which is conditioned by the *subject's representation*, and so is capable of being known. The *thing in itself* is a thing that cannot become an object. (Cf. *subject*; see *thing in itself*.) [IK] (Lat. *objectus*, pp. of *objicere*, "to throw over against") In the widest sense, object is that towards which consciousness is directed, whether cognitively or conatively The cognitive or epistemological object of mind is anything perceived, imagined, conceived or thought about. See Eptstemological Object. The conative object is anything desired, avoided or willed. [L.W.]

objective idealism. The view that the mind possesses objects, norms, or meanings of universal validity. The opposite of subjectivism, psychologism, solipsism, individualism (q.v.) [W.L.]. A name for that philosophy which is based on the theory that both the subject and the object of knowledge are equally real and equally manifestations of the absolute or ideal. Earlier employed to describe Schelling's philosophy. Used independently by Charles S. Peirce (1839-1914) and A. N. Whitehead (1861-) to describe their varieties of realism. Subjective idealism supposes the world to consist of exemplifications of universals which have their being in the mind. Objective idealism supposes the world to consist of exemplifications of universals that have their being independent of the mind. [J.K.F.]

objective reference. The self-transcendence of an immediately given content whereby it is directed toward an object. See Object. [L.W.]

objective relativism. The epistemological theory which ascribes real objectivity to all perspectives and appearances of an object of perception. (See A. E. Murphy, "Objective Relativism in Dewey and Whitehead," Philosophical Review, Vol. XXXVI, 1927.) [L.W.]

objective. Related more to the *object* or *representation* out of which *knowledge* is constructed than to the *subject* possessing the knowledge. Considered *transcendentally*, objective knowledge is less certain than *subjective* knowledge; considered *empirically*, objective knowledge is more certain. (Cf. *subjective*.) [IK]

objectivism, epistemological. A doctrine that maintains that everything apprehended is independent of the apprehender. (Montague.) [H.H.]

perceptions, minute. A term used by Leibniz for ‘unconscious perceptions’. (Cf. *Apperception*.

perspective. A way of thinking about or considering something; or a set of assumptions from which an *object* can be viewed. Knowing which perspective is assumed is important because the same question can have different an-swers if different perspectives are assumed. Kant himself does not use this word, but he uses a number of other expressions (such as *standpoint*, way of thinking, employment of *understanding*, etc.) in precisely this way. The main *Critical* perspectives are the *transcendental*, *empirical*, *logical* and *hypothetical*. [IK]

phenomenology. The description and classification of phenomenon. [AOED]. (Cf. also autophenomenology, hetrophenomenology)

phenomenon. the *object* of *knowledge*, viewed *empirically*, in its fully knowable state (i.e., conditioned by *space and time* and the *categories*). (Cf. *noumenon*.) [IK] A fact or occurrence that appears or is perceived, especially one of which the cause is in question.

phenomenonalism. The doctrine that human knowledge is confined to the appearances presented to the senses. [AOED]

practical. One of Kant's three main *standpoints*, relating primarily to action -i.e., to what we desire to do as opposed to what we know or feel. Practical *reason* is a synonym for *will*; and these two terms are concerned with questions of morality. Finding the sources of such action is the task of the second *Critique*. (Cf. *theoretical* and *judicial*.) [IK]

predisposition. The natural tendency a person has, apart from (or before having) any *experience*, to be morally good or evil. (Cf. *disposition*.) [IK]

pure. Not mixed with anything *sensible*. Although its proper opposite is 'impure', Kant normally opposes 'pure' to '*empirical*'.

rational. Brounded in the *faculty* of *reason* rather than in *sensibility*. (See also *intelligible*.) [IK] reality. If regarded from the *empirical perspective*, this refers to the ordinary world of nature; if regarded from the *transcendental* perspective, it refers to the *transcendent* realm of the *noumenon*. [IK]

reason. In Kant's first *Critique*, the highest *faculty* of the human *subject*, to which all other faculties are subordinated. It abstracts completely from the conditions of *sensibility*. The second *Critique* examines the form of our de-sires in order to construct a *system* based on the faculty of reason (= the *practical standpoint*). Reason's primary function is *practical*; its *theoretical* function, though often believed to be more important, should be viewed as having a secondary importance. (Cf. *judgment*.) [IK]

regulative. Providing important guidelines for how *knowledge* should be used, yet not itself playing any fundamental role in making up that knowledge. (Cf. *constitutive*.) [IK]

representation. The most general word for an *object* at any stage in its determination by the *subject*, or for the *subjective* act of forming the object at that level. The main types of representations are *intuitions*, *concepts* and *ideas*. In the first Critique, the understanding is the dominant faculty in processing representations, while in the third Critique the faculty of imagination is dominant. Sometimes translated as 'presentation'. [IK]

res cogitans. Thinking thing [Latin *res*, thing + *cogitan*, to think]. Descartes's term for thinking substance, in contrast to *res extensa*. [c.f.]

res extensa. Extended thing [Latin *res*, thing + *extensa* extended, external] Descartes's term for extended or corporeal substance, the physical world. Descartes presents a subjectivist point of view : it is the subject that determines the world around him, and not the other way around. subject ("res cogitans") -> world ("res extensa")

schematism. The function of the *faculty* of *imagination*, through which *concepts* and *intuitions* are combined, or *synthesized*, according to a rule (called a schema). In the first *Critique*, this function is presented as one of the steps required in order for the *understanding* to produce *empirical knowledge*. [IK]

sensibility. The *faculty* concerned with passively receiving *objects*. This is accomplished primarily in the form of physical and mental sensations (via 'outer sense' and 'inner sense', respectively). However, such sensations are possible only if the objects are intuited, and *intuition* depends on *space and time* existing in their *pure* form as well. (Cf. *understanding*.) [IK]

sensible. Presented to the *subject* by means of *sensibility*. (Cf. *intelligible*.) [IK]

soniculation. Sonic articulation. A term to identify a type of sonification (C.f.) in which principal purpose is to articulate the information being sonified as clearly as possible, rather than for the she beauty of the sound or other artistic expressive purposes. A term coined in this thesis largely to cover the somewhat ambiguous term scientific sonification,which seems to have fallen out of favour. [DW]

sonification. N. A mapping of numerically represented relations in some domain under study to relations in an acoustic domain for the purposes of interpreting, understanding, or communicating relations in the domain under study [Scaletti, AD pp 224]. A mapping of numerically represented relations in some domain under study to relations in an acoustic domain for the purposes of interpreting, understanding, or communicating relations in the domain under study. [CS 1994]. A mapping of information to perceptual relations in the acoustic domain to meet the information requirements of an information processing activity. [SB 1996]. The use of non-speech audio to convey information. More specifically, sonification is the transformation of data relations into perceived relations in an acoustic signal for the purposes of facilitating communication or interpretation. [GK 1999]. The transformation of information or data relations into acoustic signals for perception and interpretation by listeners. [DW 2008].

space and time. Considered from the *empirical perspective*, they form the context in which *objects* interact outside of us; considered from the *transcendental* perspective, they are *pure*, so they exist inside of us as conditions of *knowledge*. (Cf. *categories*.) [IK]

speculative. The illusory *perspective* which wrongly uses *reason* in a hope-less attempt to gain *knowledge* about something *transcendent*. Sometimes used loosely as a synonym of *theoretical*. [IK]

standpoint. The special type of *perspective* which determines the point from which a whole *system* of perspectives is viewed. The main *Critical* standpoints are the *theoretical*, *practical* and *judicial*. [IK]

subject. A general term for any *rational* person who is capable of having *knowledge*. (Cf. *object*; see also *representation*.) [IK]

subjective. Related more to the *subject* than to the *object* or *representation* out of which *knowledge* is constructed. Considered *transcendentally*, subjec-tive knowledge is more certain that *objective* knowledge; considered *empiri-cally*, subjective knowledge is less certain. (Cf. *objective*.) [IK]

summum bonum. Latin for 'highest good'. This is the ultimate goal of the moral *system* presented in the second *Critique*; it involves the ideal distribu-tion of happiness in exact proportion to each person's virtue. In order to con-ceive of its possibility, we must postulate the existence of God and human immortality, thus giving these *ideas practical reality*. [IK]

supersensible. see *intelligible* and *transcendent*. [IK]

synaesthesia. n. sensation Produced at a point different from the point of stimulation; a sensation of another kind suggested by one experience (e.g. in colour-hearing). adv. synaesthetic [Ch]. n. the substitution of one sensory modality for another [AD pp 11].

synthesis. integration Of two opposing *representations* into one new representation, with a view towards constructing a new level of the *object's reality*. Philosophy as *Critique* employs synthesis more than *analysis*. On the operation of synthesis in the first *Critique*, see *imagination*. (Cf. *analysis*.) [IK]

synthetic. A statement or item of *knowledge* which is known to be true because of its connection with some *intuition*. (Cf. *analytic*.) [IK]

system. A set of basic facts or arguments (called 'elements') arranged according to the order of their *logical* relationships, as determined by the *architectonic* patterns of *reason*. Kant's *Critical* philosophy is a System made up of three subordinate systems, each defined by a distinct *standpoint*, and each made up of the same four *perspectives*. [IK]

tabula rasa. Blank slate. {Latin].

teleological. Having to do with purposes or ends. The second half of the third *Critique* examines the *objective* purposiveness in our perception of natural organisms in order to construct a *system* of teleological *judgment*. [IK]

theoretical. One of Kant's three main *standpoints*, relating primarily to cognition-i.e., to what we know as opposed to what we feel or desire to do. Theoretical *reason* is concerned with questions about our *knowledge* of the ordinary world (the world science seeks to understand). Finding the source of such knowledge is the task of the first *Critique*, which would best be entitled the *Critique of Pure 'Theoretical' Reason*. (Cf. *practical* and *judicial*; see *speculative*.) [IK]

thing in itself. An *object* considered *transcendentally* apart from all the conditions under which a *subject* can gain *knowledge* of it. Hence the thing in itself is, by definition, unknowable. Sometimes used loosely as a synonym of *noumenon*. (Cf. *appearance*.) [IK]

time. See *space and time*. [IK]

transcendent. The realm of thought which lies beyond the boundary of pos-sible *knowledge*, because it consists of *objects* which cannot be presented to us in *intuition*-i.e., objects which we can never *experience* with our senses (sometimes called *noumena*). The closest we can get to gaining knowledge of the transcendent realm is to think about it by means of *ideas*. (The opposite of 'transcendent' is 'immanent'.) [IK]

transcendental object. An *object* considered *transcendentally* insofar as it has been presented to a *subject*, but is not yet *represented* in any determined way-i.e., not yet influenced by *space and time* or by the *categories*. Also called an 'object in general'.

transcendental. One of Kant's four main *perspectives*, aiming to establish a kind of *knowledge* which is both *synthetic* and *a priori*. It is a special type of philosophical knowledge, concerned with the necessary conditions for the possibility of *experience*. However, Kant believes all knowing *subjects* assume certain transcendental truths, whether or not they are aware of it. Transcendental knowledge defines the boundary between *empirical* knowledge and *speculation* about the *transcendent* realm. 'Every event has a cause' is a typical transcendental statement. (Cf. *empirical*.) [IK]

understanding. On the first *Critique*, the *faculty* concerned with actively producing *knowledge* by means of *concepts*. This is quite similar to what is normally called the mind. It gives rise to the *logical perspective*, which enables us to compare concepts with each other, and to the *empirical* perspective (where it is also called *judgment*), which enables us to combine concepts with *intuitions* in order to produce empirical knowledge. The first *Critique* examines the form of our cognitions in order to construct a *system* based on the *faculty* of understanding (= the *theoretical standpoint*). (Cf. *sensibility*.) [IK]

visile. adj. Of or pertaining to sight. learning by means of visual images and recalling such images readily. n. one whose imagery naturally takes a visual form [Ch]. C.f. audile.

will. The manifestation of *reason* in its *practical* form (see *practical*). The two German words, 'Willkür' and 'Wille' can both be translated in English as 'will'. Willkür refers to the *faculty* of choice, which for Kant is just one (*empirical*) function of the more fundamental faculty of practical reason (= *Wille*). [IK]

'world-as-it-is'. The real empirically experienced worldThe world from which Husserl's phenomenal world is bracketed off by a process known as *epoche.*

XAO The All Ordinaries Index of the ASX (c.f.). A broad market indicator.

Glossary references

[ACOD] Australian Concise Oxford Dictionary. 2nd Edition. 1992.

[AD] Auditory Display, ed. Gregory Kramer, Proc. Vol XVIII Santa Fe Institute Studies in the Sciences of Complexity; Addison-Wesley Reading MA, 1994. Definitions from Auditory Display, unless prefixed by an author's name are provided by Gregory Kramer.

[Ch] Chambers English Dictionary, mostly by way of Vicker's website.

[CS] Scaletti, C. 1994.

[DW] Worrall, D.R. 2008.

[GK] Kramer, G. et al. 1999.

[IK] As used by Immanuel Kant. Compiled by Stephen Palmquest. Used with permission. Available separately and in full at http://www.hkbu.edu.hk/~ppp/ksp1/KSPglos.html

[JB] Bertin, J. 1981.

[SB] Barrass, S. 1994.

BIBLIOGRAPHY

ACOD. 1992. The Australian Concise Oxford Dictionary. 2nd Ed. South Melbourne: Oxford University Press.

Alten, S.R. 2002. *Audio in Media*. 6th Edition. Belmont California: Wadsworth Thomson Learning.

Anderson, J., P. Sanderson and M. Norris. 2002. The role of auditory attention and auditory perception in the design of real-time sonification of anaesthesia variables. In *Proceedings of HF2002 Human Factors conference.* November 25-27, Melbourne, Australia.

Aquila, R.E. 1983. *Representational Mind*. Bloomington: Indiana University Press.

Aristotle, 350BC. *De Anima* (On the soul). J.A. Smith (trans.). Accessed 11 July 2008 at http://psychclassics.yorku.ca/Aristotle/De-anima/de-anima2.htm

Attali, J. 1985. *Noise. The Political Economy of Music*. B. Massumi (trans.). Minneapolis, MN: University of Minnesota Press.

Bachelier, L. 1900/1967. Théorie de la speculation. In *Annales de l'Ecole Normale Superieure 17* Pages 21-86. A.J. Boness (trans.). In *The Random Character of Stock Market Prices*. P.H. Cootner (ed.). Boston, MA: MIT Press. Pages 17-78.

Baier, G., T. Hermann, O. M. Lara and M. Müller. 2005. Using sonification to detect weak cross-correlations in coupled excitable systems. In *Proceedings of The Eleventh Meeting of the International Conference on Auditory Display*, Limerick, Ireland, July 6-9.

Balaban, M., K. Ebciöglu, and O. E. Laske. (eds.).1992. *Understanding music with AI: perspectives on music cognition*. Cambridge, Mass: AAAI Press.

Ballas, J.A. 1994. Delivery of information through sound. In G. Kramer (ed.). 1994. *Auditory display: Sonification, Audification, and Auditory Interfaces*. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings, Volume XVIII. Reading, MA: Addison Wesley Publishing Company. Pages 79-94.

Barrass, S. 1996a. TaDa! Demonstrations of auditory information design. In *Proceedings of the Third International Conference on Auditory Display ICAD'96*, Xerox PARC, Palo Alto, California.

Barrass, S. 1996b. EarBenders: Using stories about listening to design auditory interfaces. In *Proceedings of the First Asia-Pacific Conference on Human Computer Interaction APCHI'96,* Information Technology Institute, Singapore.

Barrass, S. 1996c. Sculpting a sound space with information properties. In *Organised Sound, Volume* 1, Number 2. Cambridge UK: Cambridge University Press. Pages 125-136.

Barrass, S. 1997. *Auditory Information Design*. Ph.D. Thesis, Australian National University. Accessed 13 May 2008 at http://thesis.anu.edu.au/public/adt-ANU20010702.150218.

Barrass, S. 2003. Sonification from a design perspective", Invited Keynote. In *Proceedings of the Ninth International Conference on Auditory Display.* Boston, USA. 6-9 July.

Barrass, S. 2004. *Listening to the Mind Listening Call for Submissions*. www.icad.org/websiteV2.0/Conferences/ICAD2004/concert_call.htm

Barrass, S. and Nesbitt, K. 2002. Evaluation of a multimodal sonification and visualisation of depth of market stock data. In *Proceedings of the Eighth International Conference on Auditory Display*, Kyoto, Japan, July 2-5.

Barrass, S., M. Whitelaw and G. Potard. 2005. Listening to the mind listening. In *Media International Australia Incorporating Culture and Policy*. Number 118. February. Pages 60-67.

Bateson, G. 1972. *Steps to an Ecology of Mind*. New York: Ballantine Books. Accessed 8 March 2010 at http://www.rawpaint.com/library/bateson/formsubstancedifference.html

Bauer, R.J. and J.R. Dahliquist. 1999. *Technical market indicators. Analysis & performance*. New York: John Wiley & Sons, Inc.

Begault, D.R. 1994. *3-D Sound for Virtual Reality and Multimedia*. Boston MA: AP Professional.

Behnke, H., F. Bachmann, K. Fladt and W. Süss (eds.). 1983. *Fundamentals of mathematics*, Volume 1. Translated from the 1962 second German edition. Cambridge, MA: The MIT Press.

Ben-Tal, O., J. Berger, B. Cook, M.Daniels, G. Scavone and P. Cook. 2002. SONART: The sonification application research toolbox. In *Proceedings of the Eighth International Conference on Auditory Display*, Kyoto, Japan, July 2-5.

Berg, B.A. 2004. Introduction to Markov chain Monte Carlo simulations and their statistical analysis. In Kendall, W.S. F. Liang, and J-S.Wang. 2005. *Markov Chain Monte Carlo innovations and applications*. Lecture Notes Series, Institute for Mathematical Sciences, National University of Singapore - Volume 7, World Scientific Publishing Co., Singapore.

Bertin, J. 1981. *Graphics and Graphic Information Processing*. Walter de Gruyter.

Best, V., E.J. Ozmeral, N. Kopco and B.G. Shinn-Cunningham. 2008. Object continuity enhances selective auditory attention. In *Proceedings of the National Academy of Sciences of the United States of America*, Volume 105, Issue 35. Pages 13174-13178.

Biology Online. 2001. A part of the Scientific American Partner Network. Accessed 29 February 2008 at http://www.biology-online.org/dictionary/.

Blattner M., D. Sumikawa and R. Greenberg. 1989. Earcons and Icons: Their structure and common design principles. In *Human Computer Interaction*, Volume 4, Number 1. London: Lawrence Erlbaum Associates. Pages 11-44.

Blattner, M.M. A.L. Papp and E.P. Glinert. 1994. Sonic Enhancement of Two-Dimensional Graphics Displays. In G. Kramer (ed.). 1994. *Auditory display: Sonification, Audification, and Auditory Interfaces*. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings, Volume XVIII. Reading, MA: Addison Wesley Publishing Company. Pages 447-470.

Bly, S. 1994. Multivariate Data Mappings. In G. Kramer (ed.). 1994. *Auditory display: Sonification, Audification, and Auditory Interfaces*. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings, Volume XVIII. Reading, MA: Addison Wesley Publishing Company. Pages 405-416.

Bolstad, W.M. 2004. *Introduction to Bayesian statistics*. John Wiley & Sons, Hoboken, New Jersey.

BonJour, L. 2007. Epistemological Problems of Perception. In *The Stanford Encyclopedia of Philosophy (Summer 2007 Edition)*, Zalta, E.N. (ed.). Accessed 19 September 2008 at http://plato.stanford.edu/archives/sum2007/entries/perception-episprob/.

BonJour, L. and E. Sosa. 2003. *Epistemic Justification: Internalism vs. Externalism, Foundations vs. Virtues*. UK: Blackwell Publishing.

Bovermann, T., T. Hermann, and H. Ritter. 2006. Tangible Data Scanning Sonification Model. In *Proceedings of the Twelfth International Conference on Auditory Display*, London, UK June 20-23.

Boyd, E. 1905. History of auditing in R. Brown (ed.). *A History of Accounting and Accountants*. Edinburgh: T.L & E.C. Jack.

Branigan, E. 1989. Sound and Epistemology in Film. In *The Journal of Aesthetics and Art Criticism*, Volume 47, Number 4 (Autumn). Pages 311-324.

Bregman, A. 1994. *Auditory Scene Analysis: The Perceptual Organization of Sound*. Cambridge, MA: The MIT Press.

Brentano, F. 1891/1995. *Descriptive psychology*. (From Brentano's lectures of 1890-1891). London: Routledge.

Bresin, R. and S. Dahl. 2003. Experiments on gestures: walking, running, and hitting. In Rocchesso, D and F. Fontana (eds.). *The Sounding Object*. Accessed on 25 October 2008 at http://www.soundobject.org.

Brewster, S. and R. Murray. 2000. Presenting dynamic information on mobile computers. In *Personal Technologies*, Volume 4, Number 2. Pages 209-212.

Brewster, S.A., P.C. Wright and A.D.N. Edwards 1994. A detailed investigation into the effectiveness of earcons. In G. Kramer (ed.). 1994. *Auditory display: Sonification, Audification, and Auditory Interfaces*. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings, Volume XVIII. Reading, MA: Addison Wesley Publishing Company. Pages 471-498.

Brewster, S.A., P.C. Wright and A.D.N. Edwards. 1993. *Parallel earcons: reducing the length of audio messages*. Accessed 5 June 2008 at http://citeseer.ist.psu.edu/154731.html

Bühlmann, P. 2000. Model selection for variable length Markov chains and tuning the context algorithm. *Annals of the Institute of Statistical Mathematics, Volume* 52. Pages 287-315.

Burk, P. and R. Bencina. 2001. PortAudio–An open source cross platform audio API. In *Proceedings of the ICMC*, La Habana, Cuba.

Burnet, J. 1904/1981. *Greek Philosophy. From Thales to Plato.* London: MacMillan Press.

Burtt, E.A. 1946. *Right Thinking: A Study of Its Principles and Methods*. New York: Harpers.

Bussemakers, M.P. and A. de Haan. 2000. When it sounds like a duck and it looks like a dog… auditory icons vs. earcons in multimedia environments. In *Proceedings of the Tenth International Conference on Auditory Display*. Georgia Institute of Technology. Atlanta, USA. Pages 184-189.

Buxton, W. 2003. Performance by design: The role of design in software product development. *Proceedings of the Second International Conference on Usage-Centered Design*. Portsmouth, NH, 26-29 October. Pages 1-15.

Cabrera, D. S. Ferguson, and E. Schubert. 2007. Psysound3: Software for acoustical and psychoacoustical analysis of sound recordings, in *Proceedings of the Thirteenth International Conference on Auditory Display*, Montreal, Canada.

Cage, J. 1961/1967. *Silence.* Cambridge, MA: The MIT Press. 1st edition: Wesleyan University Press.

Camurri, A. and G. Volpe (eds.). 2004. Gesture-based communication in human-computer interaction. Selected revised papers of the 5th

International Gesture Workshop (GW2003). Lecture Notes in Artificial Intelligence, LNAI. Berlin: Springer-Verlag.

Camurri, A. G. Volpe, G. De Poli, and M. Leman. 2005. Communicating expressiveness and affect in multimodal interactive systems. *IEEE Multimedia*, Volume 12, Number 1. Pages 43-53.

Casati, R. and J. Dokic. 2008. Sounds. *The Stanford Encyclopedia of Philosophy.* Fall 2008 Edition, E. N. Zalta (ed.). Accessed 25 September 2008 at http://plato.stanford.edu/archives/fall2008/entries/sounds/.

Cassirer, E. 1944. The concept of group and the history of perception. *Philosophy and phenomenological research*, Volume 5, Number 1. Pages 1-35.

Chafe, C. and R. Leistikow. 2001. Levels of Temporal Resolution in Sonification of Network Performance. In *Proceedings of Eighth International Conference on Auditory Display*, Espoo, Finland, July 29-August 1.

Chalmers, D.J. 1995. Facing up to the problem of consciousness. *Journal of Consciousness Studies*, Volume 2 Number 3. Pages 200-219.

Chalmers, D.J. 1996. *The Conscious Mind: In search of a fundamental theory.* New York: Oxford University Press.

Chandler, P. and J. Sweller. 1991. Cognitive load theory and the format of instruction. In *Cognition and Instruction*, Volume 8, Number 4. Pages 293-332.

ChemiCool. 2008. A part of the Scientific American Partner Network. Accessed 3 April 2008 at http://www.chemicool.com/dictionary.html.

Childs, E. 2005. Auditory graphs of real-time data. In *Proceedings of The Eleventh Meeting of the International Conference on Auditory Display*, Limerick, Ireland, July 6-9.

Chion, M. 1994. Audi–Vision: Sound on screen. Ed. and trans. by C. Gorbman New York: Columbia University Press.

Churchland, P. M. 1981. Eliminative Materialism and the Propositional Attitudes. In *Journal of Philosophy*, Volume 78. Pages 67-90.

Ciardi, F.C. 2004. 'sMAX'. A multimodal toolkit for stock market data sonification. In *Proceedings of the Tenth Meeting of the International Conference on Auditory Display*, Sydney, Australia, July 6-9.

Clynes, M 1997. *Sentics: the touch of emotions.* New York: Anchor Press.

Coguen, J.A. 2002. *Consciousness and the decline of cognitivism*. Presented at the UCSD Distributed Collective Practices symposium, 6-9 February. Accessed 28 February 2009 at http://www.cs.ucsd.edu/~goguen/pps/dcp.pdf

Cohen, J. 1994. Monitoring background activities. In G. Kramer (ed.). 1994. *Auditory display: Sonification, Audification, and Auditory Interfaces*. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings, Volume XVIII. Reading, MA: Addison Wesley Publishing Company. Pages 495-532.

Cook, P.R. (ed.). 1999. *Music, cognition, and computerized sound: An introduction to psychoacoustics*. Cambridge, MA: The MIT Press.

Cooke, D. 1959. The Language of Music. New York: Oxford University Press.

Cootner, P.H. 1962. Stock prices: Random versus systematic changes. In *Industrial Management Review 3*, Page 2445.

Cope, D. 1991. *Computers and Musical Style*. Madison, WI: A-R Editions.

Coronel, P.R.C. 2000. *Database Systems. Design, implementation and management.* 4th Edition. Cambridge, MA: Course Technology (Thomson Learning).

Coulouris, G., G. Dollimore and T. Kindberg. 2005. *Distributed systems: concepts and design*. Boston, MA: Addison-Wesley.

Cumming. N. 2000. The sonic self: musical subjectivity and signification. In *Advances in semiotics*. Bloomington, Ind: Indiana University Press.
D'haes, W. 2005. *Automatic estimation of control parameters for musical synthesis algorithms*. PhD thesis, University of Antwerp, 2004. Accessed on 11 November 2008 at http://visielab.ua.ac.be/staff/dhaes/thesis.html
Damasio, A. 1995. *Descartes' Error: Emotion, Reason, and the Human Brain*. NY: Penguin.
Damasio, A. 203. Looking for Spinoza: Joy, Sorrow, and the Feeling Brain. Orlando Fl. Harcourt.
Date, C.J. 2004. *An Introduction to Database Systems*. 8th edition. Addison-Wesley.
de Campo, A. 2007. Toward a data sonification design map. In *Proceedings of the 13th International Conference on Auditory Display*, Montréal, Canada, June 26 - 29. Pages 342-347.
de Campo, A., C. Frauenberger, and R. Höldrich. 2004. Designing a generalized sonification environment. In *Proceedings of the Tenth Meeting of the International Conference on Auditory Display*, Sydney, Australia, July 6-9. Also accessed 7 August 2007 at http://sonenvir.at/
Dean, R.T. 2005. NoiseSpeech, a Noise of Living Bodies: Towards Attali's 'Composition'. In *Journal of New Media and Culture*, Volume 3, Number 1. Accessed 12 June 2008 at http://www.ibiblio.org/nmediac/winter2004/NoiseSpc.htm
Dean, R.T. and F. Bailes. 2006. NoiseSpeech. In *Performance Research,* Volume 11, Number 3. Pages 85-86.
Dean, R.T. and F. Bailes. 2009.When is noise speech? A survey in sonic ambiguity. *Computer Music Journal*, Volume 33, Number 1. Spring. Cambridge, MA: The MIT Press. Pages 1–11,
Dean, R.T., D.R. Worrall and G. White. 2004. *Mind your Body*. A collaborative sonification. First performed in The Studio, Sydney Opera House, July 8 2004 as part of the Tenth International Conference on Auditory Display.
Deatherage, B.H. 1972. Auditory and other sensory forms of information presentation. In Van Cott H.P. and Kinkade R.G. (eds.). *Human Engineering Guide to Equipment Design (Revised Edition)* McGraw-Hill.
Dennett, D. 1987. *The Intentional Stance*. Cambridge, MA: MIT Press.
Dennis, D. 1997. Rene Descartes' Curve-Drawing Devices: Experiments in the Relations Between Mechanical Motion and Symbolic Language. In *Mathematics Magazine,* Volume 70, Number 3. Pages 163-174.
Descartes, R. 1641. Meditations. J. Veitch (trans.). 1901. Accessed 13 July 2008 at http://www.wright.edu/cola/descartes/meditation2.html.
Dewey, J. 1938. *Logic: The Theory of Inquiry*. New York: Holt.
Dombois, F. 2001. Using Audification in Planetary Seismology. In *Proceedings of Eighth International Conference on Auditory Display*, Espoo, Finland, July 29-August 1.
Dombois, F. 2002a. Auditory Seismology on free Oscillations, Focal Mechanisms, Explosions and Synthetic Seismograms. In *Proceedings of the Eighth International Conference on Auditory Display*, Kyoto, Japan, July 2-5.
Dombois, F. 2002b. Underground Sounds. An approach to earthquake prediction by auditory seismology. In *Geophysical Research Abstracts,* Volume 4. EGS02-A-02741.
Dretske, F. 2000. *Perception, Knowledge and Belief*. Cambridge: Cambridge University Press.

Dretske, F.I. 1981. *Knowledge and the flow of information*. Oxford: Blackwell.
Dreyfus, H. 1972. *What computers can't do*. NY: Harper & Row.
Edwards, R.D. and R. Magee 1948/2001. *Technical analysis of stock trends*. Stock trend service. 8th edition. Springfield, MA: Amacom Books.
Einstein, A. 1905. On the Motion—Required by the Molecular Kinetic Theory of Heat—of Small Particles Suspended in a Stationary Liquid. *Annalen der Physik*, Volume 17. Pages 549–560.
Einstein, A. 1954. *Ideas and Opinions*. New York: Bonanza Books.
Engle, R. 1982. Autoregressive conditional heteroskedasticity with estimates of the variance of United Kingdom inflation. In *Econometrica*, Volume 50. Pages 987-1008.
Fabbri, R. and M.F. Chiozo 2008. 'figusdevpack': A Python-based framework for audio and multimedia newbies. *Proceeding of the 2008 The Linux Audio Conference*. Cologne. 28 February - 2 March 2008.
Fama, E.F. 1965. The behavior of stock-market prices. *Journal of Business (Chicago)*, Volume 38. Pages 34-105.
Farmer, J.D., P.Patelli, and I.I. Zovko. 2005. The predictive power of zero intelligence in financial markets. In *Proceedings of the National Academy of Sciences of the United States of America*, Volume 102, Number 6, Pages 2254-2259. Accessed 23 October 2007 at http://www.pnas.org/content/102/6/2254.full.pdf+html. Includes links to extensive range of supplementary material.
Flowers, J. H., D.C. Buhman and K.D. Turnage. 1997. Cross-modal equivalence of visual and auditory scatterplots for exploring bivariate data samples. In *Human Factors*, Volume 39. Pages 341-351.
Flowers, J.H. 2005. Thirteen years of reflection on auditory graphing: Promises, pitfalls, and potential new directions. In *Proceedings of the First Symposium on Auditory Graphs*, Limerick, Ireland, July 10.
Fodor, J.A. 1987. *Psychosemantics*. Cambridge, Mass.: The MIT Press/
Frantti, G.E. and L.A. Leverault. 1965. Auditory discrimination of seismic signals from earthquakes and explosions. In *Bulletin of the seismic society of America*, Volume 55. Pages 1-25.
Frauenberger, C., A. de Campo and G. Eckel. 2007. Analysing time series data. In *Proceedings of the Thirteenth International Conference on Auditory Display*, Montréal, Canada, June 26 - 29.
Friedes D. 1974. Human information processing and sensory modality: Cross-modal functions. Information complexity, memory and deficit. In *Psychological Bulletin* Number 81, Volume 5. Pages 284-310.
Fromm, E. 1947. *Man for Himself. An Inquiry into the Psychology of Ethics*. New York: Rinehart.
Frysinger, S. P. 1990. Applied research in auditory data representation. *Extracting Meaning from Complex Data: Processing, Display, Interaction*, Santa Clara, CA: SPIE-The International Society for Optical Engineering.
Frysinger, S.P. 2005. A brief history of auditory data representation to the 1980s. In *Proceedings of the First Symposium on Auditory Graphs*, Limerick, Ireland, July 10.
Galambos, J.A., R. P. Abelson and J.B. Black. (eds.). 1986. *Knowledge Structures*. New Jersey: Lawrence Erlbaum Associates.
Gann, W.D. 1942. *How to make profits in commodities*. Revised 1951. Washington Pomeroy: Library of Gann Publishing.
Gaver, W.W. 1994. Using and Creating Auditory Icons. In G. Kramer (ed.). 1994. *Auditory display: Sonification, Audification, and Auditory Interfaces*.

Santa Fe Institute Studies in the Sciences of Complexity, Proceedings, Volume XVIII. Reading, MA: Addison Wesley Publishing Company. Pages 417-446.

Gaver, W.W. and R.B. Smith. 1990. Auditory Icons in Large-Scale Collaborative Environments. In *Proceedings of Human-Computer Interaction: Interact '90*, Cambridge, UK. Pages 735-740.

Gibson, J.J. 1966. *These senses considered as perceptual systems*. Boston, MA: Houghton Mifflin Company.

Gibson, J.J. 1979. *The ecological approach to visual perception*. Boston, MA: Houghton Mifflin Company.

Goehr, L. 1994. London: The Imaginary Museum of Musical Works. Oxford, UK: Oxford University Press.

Goldman, A. 2008. Reliabilism. In *The Stanford Encyclopedia of Philosophy* (Fall 2008 Edition), (ed.). Edward N. Zalta. Accessed 30 September 2008 at http://plato.stanford.edu/archives/fall2008/entries/reliabilism/

Gosh, P. 2008. Team records 'music' from stars. *BBC news* website. Accessed 24 October 2008 at http://news.bbc.co.uk/2/hi/science/nature/7687286.stm. Related (anonymously authored) sites are: http://www.ournightsky.com/stellarseismology.html and http://www.noao.edu/education/ighelio/solar_music.html.

Graham, B. and D. Dodd. 1934. *Security analysis: Principles and technique* 4th Edition. 1962. New York: McGraw-Hill Book Company.

Gregory, R.L. 1981. *Mind in science*. Cambridge University Press, UK.

Hankinson, J. C. K. and A.D.N. Edwards. 2000. *Musical phrase-structured audio communication*. In *Proceedings of the Tenth International Conference on Auditory Display*. Georgia Institute of Technology. Atlanta, USA. Pages 200-205. Also available as a multimedia presentation at http://www-users.cs.york.ac.uk/~alistair/research/dphil/jckh/mpsac/.

Harrar, L. and T. Stockman. 2007. Designing Auditory Graph Overviews: An examination of discrete vs. continuous sound and the influence of presentation speed. In *Proceedings of the Thirteenth International Conference on Auditory Display*, Montréal, Canada, June 26-29.

Hatfield, G. 2002. Perception as unconscious inference. In *Perception and the Physical World: Psychological and Philosophical Issue in Perception*. D. Heyer and R. Mausfeld (eds.). West Sussex, UK: John Wiley & Sons, Ltd. Pages 115-143.

Hayward C. 1994. Listening to the Earth sing. In G. Kramer (ed.). 1994. *Auditory display: Sonification, Audification, and Auditory Interfaces*. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings, Volume XVIII. Reading, MA: Addison Wesley Publishing Company. Pages 369-404.

Heidegger, M. 1954/1977. The Question Concerning Technology. In W. Lovitt. *The Question Concerning Technology and Other Essays*, Harper Torchbooks.

Heidegger, M. 1927. *Being and Time*. (Trans. J. Macquarrie, 1962 from *Sein und Zeit*). Oxford: Blackwell.

Helmholtz, H. 1878. The facts of perception. In R. Kahl (ed.). *Selected Writings of Hermann Helmholtz*, Wesleyan University Press.

Hergenhahn, B.R. 1992. *An introduction to the History of Psychology*. Belmont, CA: Wadsworth Publishing Company.

Hermann, T. 2002. *Sonification for Exploratory Data Analysis*, Ph.D. thesis,

Bielefeld University, Bielefeld, Germany.
Hermann, T. and H. Ritter. 1999. Listen to your data: Model-based sonification for data analysis. In *Advances in intelligent computing and multimedia systems*, G. E. Lasker (ed.). Baden-Baden, Germany, Int. Institute for Advanced Studies in System research and cybernetics. Pages 189-194.
Hermann, T. M., H. Hansen and H. Ritter. 2001. Sonification of Markov Chain Monte Carlo simulations, Sonification of Markov Chain Monte Carlo simulations. In *Proceedings of the Seventh International Conference on Auditory Display*, Helsinki University of Technology. Pages 208-216.
Hermann, T., G. Baier , U. Stephani , H. Ritter. 2006. Vocal Sonification of pathologic EEG features. In *Proceedings of the Twelfth International Conference on Auditory Display*, London, UK June 20 – 23.
Hermann, T.P. Meinicke, and H. Ritter 2000. Principal curve sonification. In *Proceedings of the Tenth International Conference on Auditory Display.* Georgia Institute of Technology. Atlanta, USA. Pages 81-86.
Heymann, M. and M. Hansen. 2002. A new set of sound commands for R; sonification of the HMC algorithm. In *ASA Proceedings, Statistical Computing Section*. Alexandria, VA: American Statistical Association. Pages1439-1443.
Hoffmeyer, J. and Emmeche, C. 1991. Code-duality and the semiotics of nature. In I. M. Anderson and F. Merrell (eds.). *On semiotic modeling*. Berlin and New York: Mouton de Gruyter. Pages 117-166.
Huemer, W. 2008. Franz Brentano. *The Stanford Encyclopedia of Philosophy* (Fall 2008 Edition), E. N. Zalta (ed.). Accessed 3 February 2008 at http://plato.stanford.edu/archives/fall2008/entries/brentano/.
Hume, D. 1777/1902. *An Enquiry Concerning Human Understanding*. L.A. Selby-Bigge (ed.).Posthumous Second Edition. A Project Gutenberg Ebook. Accessed 30 June 2008 at http://www.gutenberg.org/etext/9662.
Hunter, G. 1971. Metalogic. *An introduction to the metatheory of standard first order logic*. University of California Press.
Husserl, E. 1927/1971. Phenomenology. Article for *Encyclopaedia Britannica*. Revised and translated by R. E. Palmer in *Husserl's Shorter Works* Pages 21-35 in Journal of the British Society for Phenomenology 2. Pages 77-90. Accessed 2 February 2008 at http://www.hfu.edu.tw/~huangkm/phenom/husserl-britanica.htm.
Husserl, E. 1960. *Cartesian Meditations*: An introduction to Phenomenology. D. Cairns (trans.). The Netherlands: Kluwer Academic. Accessed 2 February 2008 at http://www.archive.org/details/cartesianmeditat017661mbp
Hutchins, E. 1995. *Cognition in the Wild*. Cambridge, MA: The MIT Press.
Innis, R.E. 1994. *Consciousness and the Play of Signs*. Bloomington and Indianapolis IL: University of Illinois Press.
Iverson, K.E. 1980. Notation as a tool for thought. The 1979 ACM Turing Award Lecture. In *Communications of the ACM*, Volume 23, Number 8. Accessed 29 May 2008 at http://portal.acm.org/citation.cfm?id=1234322#.
Janata, P and Childs, E. 2004. 'Marketbuzz': Sonification of real-time financial data. In *Proceedings of the Tenth Meeting of the International Conference on Auditory Display*, Sydney, Australia, July 6-9.
Kant, I. 1783. *Prolegomena to any future metaphysics*. English translation by J.

Fieser, based on P. Carus' 1902 translation. Accessed 7 May 2008 at http://philosophy.eserver.org/kant-prolegomena.txt.
Kant, I. 1787/1929. *Critique of pure reason*. 2nd Edition. N.K. Smith trans. of Kritik der reinen Vernunft. London: Macmillan. Accessed 11 May 2007 at http://arts.cuhk.edu.hk/Philosophy/Kant/cpr/.
Kant, I. 1787/1929b. *Critique of pure reason*. 2nd Edition. J.M.D. Meiklejohn (trans.). A Project Gutenberg Etext. Accessed 11 May 2007 at http://www.gutenberg.org/dirs/etext03/cprrn10.txt.
Kant, I. 1790 Critique of Judgement 1st Ed. trans. J.C. Meredith. Accessed 7 May 2008 at http://philosophy.eserver.org/kant/critique-of-judgment.txt
Kaplan, B. 1993. Sonification in AVS. In *AVS '93, Walt Disney World, Lake Buena Vista, FL*, May 24-26.
Khinchin, A.I. 1957. *Mathematical foundations of information theory*. New York: Dover Publications. Translation from the original 1953 Russian edition.
Khun, T.S. 1962. *The structure of scientific revolutions*. Chicago: University of Chicago Press.
Kirschner, P.A., Sweller, J., and Clark, R.E. 2006. Why minimal guidance during instruction does not work: an analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist*, Volume 41, Number 2. Pages 75-86.
Kline, M. 1972. Mathematical thought from ancient to modern times. Oxford: OUP.
Knuth, D.E. 1981. *The Art of computer programming, Volume 2 Seminumerical Algorithms*.2nd Edition. Reading, MA: Addison Wesley Publishing.
Kohonen, T. 1995. *Self organizing maps. Springer series in information sciences.* Number 30. Berlin: Springer.
Kramer, G and S. Ellison. 1991. Audification: The use of sound to display multivariate data. In *Proceedings of the International Computer Music Conference*. San Francisco, C.A. ICMA. Pages 214-221.
Kramer, G. 1994. Some organizing principles for representing data with sound. In G. Kramer (ed.). 1994. *Auditory display: Sonification, Audification, and Auditory Interfaces*. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings, Volume XVIII. Reading, MA: Addison Wesley Publishing Company. Pages 185-221.
Kramer, G. 1994b. An Introduction to Auditory Display. In G. Kramer (ed.). 1994. *Auditory display: Sonification, Audification, and Auditory Interfaces*. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings, Volume XVIII. Reading, MA: Addison Wesley Publishing Company. Pages 1-77.
Kramer, G. 2008. Personal correspondence with the author.
Kramer, G., B. Walker, B. T. Bonebright, P. Cook, J. Flowers, P.N. Miner and J. Neuhoff. 1999. (eds.) *Sonification report: Status of the field and research agenda* Technical report, International Community for Auditory Display. Accessed 29 February 2008 at http://www.icad.org/websiteV2.0/References/nsf.html
Krishnan S., R.M. Rangayyan, B.G. Douglas and C.B. Frank. 2001.Auditory display of knee-joint vibration signals, Journal of the Acoustical Society of America, Volume 110, Number 6, December 2001. Pages 3292-3304.
Küppers, B. O. 1990. Information and the origin of life. Cambridge: MIT Press.
Ladd, C.O., P.M. Plotsky and M. Davis. 2000. Startle response. Page 486 in

Encyclopedia of stress, Volume 3. G. Fink (ed.). Academic Press. San Diego, California.
Lagerlund, H. 2008. Mental Representation in Medieval Philosophy. In *The Stanford Encyclopedia of Philosophy* (Fall 2008 Edition), E.N. Zalta (ed.), Accessed 30 September 2008. at http://plato.stanford.edu/archives/fall2008/entries/representation-medieval/
Lamont, C. 1949/1997. *The Philosophy of humanism*. 8th Edition. Amherst, New York: Humanist Press.
Landy, L., 2007. *Understanding the art of sound organization*. Cambridge, MA: MIT Press.
Lazzarini, V. 2000. The sound object library. In *Organized Sound, Volume* 5, Number 1. Cambridge, UK: Cambridge University Press. Pages 35-49.
Lefèvre, E. 1923/1994. *Reminiscences of a Stock Operator*. New Jersey: John Wiley & Sons.
Lehar, S. 2000. The function of conscious experience: An analogical paradigm of perception and behavior. Accessed 15 March 2008 at http://cns-alumni.bu.edu/~slehar/webstuff/consc1/consc1.html
Lemmon, E. J. 1965. *Beginning logic*. London: Thomas Nelson and Sons.
Levy, G. An Introduction to GARCH Models in Finance.
Lipton, P. 2004. *Inference to the Best Explanation*. 2nd edition, Routledge.
Locke, J. 1690/1975. *An essay concerning humane understanding*. 2nd edition. Editor P. H. Nidditch, Oxford. Original MS accessed 17 April 2008 at http://www.gutenberg.org/etext/10615/ (Book I) and http://www.gutenberg.org/etext/10616/ (Book II).
Lombardo, T.J. 1987. *The reciprocity of perceiver and environment. The evolution of James J. Gibson's ecological psychology*. Hillsdale, N. J.: Lawrence Erlbaum.
Lucas, P.A. 2004. An Evaluation of the communicative ability of auditory icons and earcons. In G. Kramer (ed.). 1994. *Auditory display: Sonification, Audification, and Auditory Interfaces*. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings, Volume XVIII. Reading, MA: Addison Wesley Publishing Company. Pages 79-94.
Luke, R., A. Clement, R. Terada, D. Bortolussi, C. Booth, D. Brooks and D. Christ. 2004. The Promise and Perils of a Participatory Approach to Developing an Open Source Community Learning Network. In *Proceedings Participatory Design Conference*. Toronto, Canada.
Lutz, M. 1996. *Programming Python*. O'Reilly & Associates. Sebastopol CA.
Mach, E. 1886/1897. *Analysis of Sensations.*
Mandelbrot, B.B. 1997. *Fractals and scaling in finance: Discontinuity, concentration and risk*. New York: Springer-Verlag.
Mandelbrot, B.B. and R.L. Hudson. 2004. *The (mis)behaviour of markets.* New York: Basic Books.
Manzotti, R. 2005. An outline of an alternative view of conscious perception. Delivered at *Toward a Science of Consciousness*. Copenhagen. August 17-20.
Martin, 2007. Review of D.W. Smith. 2006. *Husserl.* Oxford, UK: Routledge. Accessed on 14 March 2008 at http://ndpr.nd.edu/review.cfm?id=10923.
Mathews, M. V. 1969. *The Technology of Computer Music*. Cambridge: Cambridge, MA: The MIT Press.
Mathews, M.V. 1999. What is loudness? In P.R. Cook, P.R. (ed.). *Music, cognition, and computerized sound: An introduction to psychoacoustics*.

Cambridge, MA: The MIT Press. Pages 71–78.
Maturana H.R. and F. J. Varela. 1987. *The tree of knowledge: the biological roots of human understanding*. Boston: New Science Library.
Mauney, B.S. and B. N. Walker. 2004. Creating functional and livable soundscapes for peripheral monitoring of dynamic data. In *Proceedings of the Tenth Meeting of the International Conference on Auditory Display*, Sydney, Australia, July 6-9.
Mayer-Kress, G., R. Bargar, and I. Choi. 1994. Musical structures in data from chaotic attractors. In G. Kramer (ed.). 1994. *Auditory display: Sonification, Audification, and Auditory Interfaces*. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings, Volume XVIII. Reading, MA: Addison Wesley Publishing Company. Pages 341–367.
McCabe, B.P.M, Martin G.M. and Freeland, R.K. 2004. *Testing for dependence in non-Gaussian time series data*. Accessed 13 July 2007 at http://www.buseco.monash.edu.au/depts/ebs/pubs/wpapers/
McCormick, E. J. and N. Sanders. 1984. *Human Factors in Engineering and Design*, International Student Edition, McGraw Hill.
McGookin, D.K. 2004. *Understanding and improving the identification of concurrently presented earcons*, PhD thesis, University of Glasgow, UK. [cited on 20 June 2008]. Accessed 29 February 2008 at http://theses.gla.ac.uk/14/01/mcgookin.pdf
McNeil, D. and P. Freiberger. 1993. *Fuzzy logic*. Melbourne: Bookman Press.
Merleau-Ponty, M. 1964. *The Primary of Perception and Other Essays on Phenomenological Psychology, the Philosophy of Art, History and Politics.* J. M. Edie. (ed.). Evanston: Northwestern University Press.
Mezrich, J.J., S.P. Frysinger, and R. Slivjanovski. 1984. Dynamic representation of multivariate time-series data. In *Journal of the American Statistical Association*, Volume 79. Pages 34-40.
MIDI http://www.midi.org/.
Miele, J.A. 2003. Smith-Kettlewell display tools: a sonification toolkit for Matlab. In *Proceedings of the Ninth International Conference on Auditory Display*. Boston, USA. 6-9 July.
Mill, J. S. 1865/2005. *An Examination of Sir William Hamilton's Philosophy.* London: Longman Green and Co. Elibron Classics.
Miller, G. A. 1956. The magical number seven plus or minus two : Some limits on our capacity for processing information. *Psychological Review*, Volume 63. Pages 81-97.
Moles, A. 1966. *Information Theory and Esthetic Perception*. Trans. J.E.Cohen. Urbana and London: University of Illinois Press.
Moore, F. R. 1990. *Elements of Computer Music*. Prentice Hall.
Münsterberg, H. 1899. Psychology and History President's Address, American Psychological Association, New York Meeting, December,. First published in *Psychological Review*, Volume 6, 1-31. Accessed 23 May 2008 at http://psychclassics.yorku.ca/Munster/history.htm.
Murphy, J. 1999. *Technical Analysis of the Financial Markets*. New York: New York Institute of Finance.
Mynatt, E. D. 1994. Auditory Presentation of Graphical User Interfaces. In *Auditory Display: Sonification, Audification, and Auditory Interfac*e, Volume XVIII, Santa Fe Institute, Studies in the Sciences of Complexity Proceedings, G. Kramer, Ed. Reading, MA: Addison-Wesley. Pages 533-556.
N. Nettheim, N. 1996. How musical rhythm reveals human attitudes: Gustav

Becking's theory. *International Review of the Aesthetics and Sociology of Music*, Volume 27 Number 2. Pages 101– 122.
Nees, M. A. and B.N. Walker. 2007. Listener, task, and auditory graph: Toward a conceptual model of auditory graph comprehension. In *Proceedings of the Thirteenth International Conference on Auditory Display*, Montréal, Canada, June 26 - 29. Pages 266–273.
Nees, M. A. and B.N. Walker. 2008. Data density and trend reversals in auditory graphs: Effects on point-estimation and trend-identification tasks. In *ACM Transactions on Applied Perception*, Volume 5, Number 3, Article 13.
Nesbitt, K. V. and S. Barrass. 2002. Evaluation of a multimodal sonification and visualisation of depth of market stock data. In *Proceedings of the Eighth International Conference on Auditory Display*, Kyoto, Japan, July 2-5.
Neuhoff , J.G. , G. Kramer and J. Wayand. 2002. Sonification and the interaction of perceptual dimensions: Can the data get lost in the map? In *Journal of Experimental Psychology: Applied*, Volume 8, Number 1. Pages 17-25.
Nigam, N.C. 1983. *Introduction to Random Vibrations*. Cambridge, MA: The MIT Press.
NIST/SEMATECH, 2005 *e-Handbook of Statistical Methods*, http://www.itl.nist.gov/div898/handbook/ 20051130.
OLED. Online Etymology Dictionary. 2001. Ed. Douglas Harper. http://www.etymonline.com/
OLMD. 1997. *Online Medical Dictionary*. http://cancerweb.ncl.ac.uk/omd/index.html. Newcastle: Department of Medical Oncology, University of Newcastle upon Tyne.
Olsen, R. 2002. *High Frequency Data-an essential resource*. In IFC Bulletin 13, December.
Olson, H.F. 1967. *Music, physics and engineering*. Dover Publications.
Palladino, D.K. and B.N. Walker. 2007. Learning rates for auditory menus enhanced with spearcons versus earcons. In *Proceedings of the Thirteenth International Conference on Auditory Display*, Montréal, Canada, June 26-29, 2007.
Papert, S. 1980. *Mindstorm: Children, Computers, and Powerful Ideas*. New York: Basic Books.
Partridge, D. 1991. *A New Guide to Artificial Intelligence*. Bristol UK: Intellect Books.
Patterson, R.D. 1982. *Guidelines for auditory warning systems on civil aircraft*. Paper No. 82017. London: Civil Aviation Authority.
Patterson, R.D. 1989. Guidelines for the design of auditory warning sounds. *Proceeding of the Institute of Acoustics*, Spring Conference, Volume 11, Number 5. Pages 17-24.
Pauletto, S. and A. Hunt. 2004. A Toolkit for interactive sonification. In *Proceedings of the Tenth Meeting of the International Conference on Auditory Display*, Sydney, Australia, July 6-9.
Pauletto, S. and A. Hunt. 2005. A comparison of audio and visual analysis of complex time-series data sets. *Proceedings of The Eleventh Meeting of the International Conference on Auditory Display*, Limerick, Ireland, July 6-9.
Pearl, J. 2000. *Causality: Models, Reasoning, and Inference*. Cambridge MA: Cambridge University Press.
Pearson, M. 1996. TAO: a physical modelling system and related issues. In *Organised Sound*, Volume 1: Cambridge University Press. Pages 43-50.

Peinke, J., M. Siefert, S. Barth, C. Renner, F. Reiss, M. Wâcher and R. Friedrich 2004. Fat tail statistics and beyond. In *Advances in Solid State Physics*, Volume 4. Pages 363-374.
Peirce, C.S. 1868. Some Consequences of Four Incapacities. In *Journal of Speculative Philosophy*, Volume 2. Pages 140-157. Accessed on 3 March 2008 at http://www.peirce.org/writings/p27.html.
Peirce, C.S. c.1897. An untitled manuscript. In *Collected Papers of Charles Sanders Peirce*, eds. Charles Hartshorne and Paul Weiss, Volume 1 Para 171. Cambridge, Mass.: Harvard University Press, 1931-1958. Accessed 23 May 2008 at http://www.helsinki.fi/science/commens/terms/fallibilism.html.
Peters, E. E. 1991. *Chaos and order in the capital markets*. New York: John Wiley & Sons, Inc.
Piaget, J. and Garcia, R. 1991. *Towards a logic of meanings*. P.M. Davidson and J. Easley (eds.). New Jersey: Lawrence Erlbaum Associates. Accessed 7 May 2008. at http://www.questia.com/read/59090284?title=Toward%20a%20Logic%20of%20Meanings.
Pilkington, R.M. 1992. *Intelligent help. Communicating with knowledge-based systems*. New York: Oxford University Press.
Poincaré, H. 1913. *The Foundations of Science*. English translation by G. B. Halsted. New York: The Science Press. Originally published in 1905 as *La Science et l'Hypothèse.* Accessed 10 May 2008 at http://www.archive.org/details/scienceandhypoth00poinuoft.
Polansky, L. and E. Childs. 2002. *Manifestation and Sonification. The Science and Art of Sonification, Tufte's Visualization, and the "slippery slope" to Algorithmic Composition*. An Informal Response to Ed Childs' Short Paper on Tufte and Sonification; with additional commentary by Ed. Accessed on 12 January 2006 at http://eamusic.dartmouth.edu/~larry/sonification.html
Polanyi, M. 1966. *The tacit dimension*. Garden City, N.J.: Doubleday.
Polanyi, M. 1975.(with H. Prosch) *Meaning*. Chicago: University of Chicago Press.
Pollack, I. and L. Ficks. 1954. Information of elementary multidimensional auditory display. In *Journal of the Acoustic Society of America*, Volume 26. Pages 155–158.
Poole, D. 2000. Learning, Bayesian probability, graphical models, and abduction. In P.A. Flach and A. C. Kakas (eds.). *Abduction and Induction: Essays on Their Relation and Integration*. Norwell, MA: Kluwer Academic Publishers. See http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.45.4813
Popper, K.R. 1959/1972. *The logic of scientific discovery*. Revised Edition. London: Hutchenson.
Ramsey, F.P. 1931. Knowledge. in *The foundations of mathematics and other essays*. R. B. Braithwaite (ed.). New York: Harcourt Brace.
Read, J.R. (2006). *Learning theories*. Accessed 11 May 2008 at http://acell.chem.usyd.edu.au/Learning-Theories.cfm
Reicher, M. 2006. Nonexistent Objects. *The Stanford Encyclopedia of Philosophy* Edward N. Zalta (ed.). Accessed 2 February 2008 at http://plato.stanford.edu/archives/fall2008/entries/nonexistent-objects/.
Repp, B.H. 1993. Music as motion: a synopsis of Alexander Truslit's (1938)

Gestaltung und Bewegung in der Music. *Psychology of Music*, Volume 12, Number 1 Pages 48–72.
Roads, C. 1996. (ed.). *The computer music tutorial*. Cambridge MA: MIT Press.
Roads, C. 2004. *Microsound.* Cambridge, MA: The MIT Press.
Roberts, L.A., and C.A. Sikora. 1997. Optimising feedback signals for multimedia devices: Earcons vs. auditory icons vs. speech. In *Proceedings of the International Ergonomics Association Congress*, Tampere, Finland. Pages 224-226.
Roederer, J.G. 1993. *Introduction to the Physics and Psychophysics of Music.* New York: Springer Verlag. 4th Edition, published in 1998 as *Physics and Psychophysics of Music*. New York:Springer-Verlag.
Roederer, J.G. 2005. *Information and its Role in Nature*. Heidelberg: Springer-Verlag.
Roederer, J.G. 2008. *Information, Physics, Life and Brain*. A summary of Roederer (2005) provided by the author October. Accessed 19 January 2009 at http://www.gi.alaska.edu/~Roederer/pdf/book_summary.pdf
Rogers, J.L, W.A. Nicewander and L. Toothmaker 1984. Linearly independent, orthogonal, and uncorrelated variables. *The American Statistician*, May 1984, Volume 38, Number. 2. Pages 133-134.
Runes, D.D. 1942. (ed.). *Dictionary of philosophy*. New York: Philosophical Library.
Runyon, R.P., A. Hauber, D.J. Pittenger and K.A. Coleman. 1996. In *Fundamentals of behavioral statistics*. Eighth Edition. Boston, MA: McGraw-Hill.
Saue, S. and Fjeld, O.K. 1997. A platform for audiovisual seismic interpretation. I In *Proceedings of the Fourth Meeting of the International Conference on Auditory Display*, Palo Alto, California. November 2-5.
Scaletti, C. 1994. Sound synthesis algorithms for auditory data representation. In G. Kramer (ed.). 1994. *Auditory display: Sonification, Audification, and Auditory Interfaces*. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings, Volume XVIII. Reading, MA: Addison Wesley Publishing Company. Pages 223-251.
Scavone, G.P. and P. Cook 2005. RtMidi, RtAudio and a synthesis toolkit (STK) update. In *Proceedings of the 2005 International Computer Music Conference,* Barcelona, Spain.
Schaeffer, P. 1966. *Traité des objets musicaux: Essai interdisciplines*. Paris: Editions du Seuil.
Schmidt, P. 1997. The Serpent Website. Accessed on 1 April 2007 at http://www.serpentwebsite.com/.
Schopenhauer, A. 1844. *The world as will and representation.* R.B. Haldane and J. Kemp (trans.). London: Kegan Paul, Trench, Trubner & Co.
Segall, M., D. Campbell and M. J. Herskovits. 1966. *The Influence of Culture on Visual Perception*. New York: The Bobbs-Merrill Company.
Sekuler, R and R. Blake. 1985. *Perception*. New York: Alfred A. Knopf, Inc.
Setzer, V.W. 2006. Data, information, knowledge and competence. In *Proceeding of 3rd CONTECSI International Conference on Information Systems and Technology Management*. San Paulo: University of Sao Paulo, Brazil, 31 May 31 -2 June. Accessed 27 April 2008 at http://www.ime.usp.br/~vwsetzer/data-info.html.
Shannon, C. and Weaver, W. 1949. *The Mathematical Theory of Communication.* Urbana, Ill: The University of Illinois Press.
Shepard, R. 1999. Pitch perception and measurement. In P. R. Cook (ed.).

Music, cognition, and computerized sound: An introduction to psychoacoustics. Cambridge, MA: The MIT Press. Pages 149–165.
Sikora, C.A., L.A. Roberts, and L. Murray. 1995. Musical vs. real world feedback signals. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, New York: Association for Computing Machinery. Pages 220-221.
Simmel, G. 1900/1979. *The Philosophy of Money*. D. Frisby (ed.). Translated from *Philosophie der Geldes*, Leipzig: Duncker & Humblot, 1900, 2nd Edition 1907 by T. Bottomore and D. Frisby. New York: Routledge.
Skiena, S.S. 2004. *Financial time series data.* Accessed 20 July 2007 at http://www.cs.sunysb.edu/~skiena/691/lectures/lecture6/lecture6.html.
Slevc, L.R, J.C. Rosenberg and A.D. Patel. 2008. Language, music, and modularity: Evidence for shared processing of linguistic and musical syntax. In *Proceedings of the 10th International Conference on Music Perception & Cognition (ICMPC10)*. Sapporo, Japan. M. Adachi et al., (eds.). August.
Smith S. Pickett R.M. and Williams M.G. 1994. Environments for Exploring Auditory Representations of Multidimensional Data. In G. Kramer (ed.). 1994. *Auditory display: Sonification, Audification, and Auditory Interfaces*. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings, Volume XVIII. Reading, MA: Addison Wesley Publishing Company. Pages 167-184.
Smith, D.R and B.N. Walker. 2005. Effects of auditory context cues and training on performance of a point estimation sonification task. In *Applied Cognitive Psychology*, Volume 19, Number 8. Pages 1065–1087.
Smith, D.W. 2007. *Husserl*. London-New York: Routledge.
Smith., D.W. 2006. *Husserl.* Oxford, UK:Routledge.
Sornette, D. 2003. *Why Stock Markets Crash. Critical events in complex financial systems*. Princeton, N.J: Princeton University Press.
Speeth, S.D. 1961. Seismometer sounds. In *Journal of the Acoustic Society of America*, Volume 33. Pages 909-916.
Stamper, R. 1996. Signs, Information, Norms and Systems. In Holmqvist et al (eds.). *Signs of Work: Semiosis and Information Processing in Organisations*. New York: Walter de Gruyter.
Stevens, S.S. 1946. On the theory of scales and measurement. *Science*, Volume 103. Pages 677-680.
Stockhausen, K. 1961. Two Lectures: I: Electronic and Instrumental Music. In *Die Reihe*. H. Eimert and K. Stockhausen (eds.). (English ed.). Volume 5. Pages 59-67. Bryn Mawr, Penn: Theodore Presser.
Stockman, T., G. Hind and C. Frauenberger. 2005. Interactive Sonification Spreadsheets. In *Proceedings of The Eleventh Meeting of the International Conference on Auditory Display*, Limerick, Ireland, July 6-9.
Stockman, T., L.V. Nickerson and G. Hind. 2005. Auditory graphs: A summary of current experience and towards a research agenda. In *Proceedings of the First Symposium on Auditory Graphs*, Limerick, Ireland, July 10.
Stuart, R. 1996. *The Design of Virtual Environments*. New York: McGraw-Hill.
Sturm, B.L. 2000. Sonification of particle systems via de Broglie's Hypothesis. In *Proceedings of the Tenth International Conference on Auditory Display.* Georgia Institute of Technology. Atlanta, USA.
Sweller, J. 1999. *Instructional design in technical areas.* Camberwell, Australia:

Australian Council for Educational Research.
Sweller, J.1988.Cognitive load during problem solving: Effects on learning. *Cognitive Science*, Volume 12. Pages 257–285.
Taylor, A.E. 1911. The Words eido. In *Varia Socratica*, First Series. Oxford: James Parker.
Taylor, T. D. 1997. *Global pop: World music, world markets*. New York: Routledge.
Todd, P.M and G. Loy. 1991. (eds.). *Music and connectionism*. Cambridge, MA: MIT Press.
Tranas, R. The Passion of the Western Mind. 1991. London: Random House.
Truax, B. (ed.). 1978. *Handbook of Acoustic Ecology*. Vancouver: A.R.C. Publications.
Truax, B. 1988. Real-time granular synthesis with a digital signal processing computer. In *Computer Music Journal*, Volume 12, Number 2. Pages 14-26.
Varela, F., E. Thompson and E. Rosch. 1991. *The Embodied Mind.* Cambridge, MA: The MIT Press.
Vickers, P. 2004. External Auditory Representations of Programs: Past, Present, and Future - An Aesthetic Perspective. In *Proceedings of the Tenth Meeting of the International Conference on Auditory Display*, Sydney, Australia, July 6-9.
Vickers, P. 2005. Whither and wherefore the auditory graph? Abstractions & aesthetics in auditory and sonified graphs. In *Proceedings of the First Symposium on Auditory Graphs*, Limerick, Ireland, July 10.
Vincente, K.J. 2002. Ecological interface design. In *Human Factors* Volume 44, Number 1, Spring. Pages 62–78.
Voss, R.F. 1975. *1/f noise: diffusive systems and music*. Unpublished doctoral dissertation, University of California, Berkeley.
Voss, R.F. 1978. Linearity of 1/f Noise Mechanisms. In *Physical Review 40*. Letters. Pages 913 -916.
Voss, R.F. 1988. Fractals in nature: from characterization to simulation. In H-O Peitgen and D. Saupe, D. (eds.). *The Science of Fractal Images.* New York: Springer-Verlag. Pages 21-70.
Voss, R.F. and J. Clarke. 1975. '1/f noise' in music and speech. In *Nature* 258 November. Pages 317-318.
Voss, R.F. and J. Clarke. 1978. "1/*f* noise" in music: Music from 1/*f* noise. In *Journal of the Acoustical Society of America*, Volume 63, Number 1. Pages 258-263.
Walker, B. N. and M.A. Nees. 2005. An agenda for research and development of multimodal graphs. In *Proceedings of The Eleventh Meeting of the International Conference on Auditory Display*, Limerick, Ireland, July 6-9. Pages 428–432.
Walker, B.N. and J.T. Cothran. 2003. Sonification Sandbox: A Graphical Toolkit for Auditory Graphs. In *Proceedings of the Ninth International Conference on Auditory Display.* Boston, USA. 6-9 July.
Walker, B.N., A. Nance and J. Lindsay. 2006. Spearcons: speech-based earcons improve navigation performance in auditory menus. In *Proceedings of the Twelfth International Conference on Auditory Display*, London, UK, June 20-23.
Warin, C. 2002. *Sound As Means for Data Representation for Data Mining,* Master Degree Thesis, Facultés Universitaires Notre-Dame de la Paix, Namur.
Watson, M.O. and P. Sanderson. 2007. Designing for attention with sound: Challenges and extensions to ecological interface design. In *Human*

Factors, Volume 49, Number 2. Pages 331–346.

Watter, A., G.van Rossen and J. Ahlstrom. 1996. *Internet programming with Python*. New York: M&T Books.

Wenzel, E.M. 1994. Spatial sound and sonification. In G. Kramer (ed.). 1994. *Auditory display: Sonification, Audification, and Auditory Interfaces*. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings, Volume XVIII. Reading, MA: Addison Wesley Publishing Company. Pages 127-150.

Werker, J. and A. Vouloumanos. 2001. Speech and language processing in infancy: A neurocognitive approach, in *Handbook of developmental cognitive neuroscienc*e Nelson, C.A., M. Luciana (Eds). Cambridge, MA: The MIT Press.

Whitehead, A.N. and B. Russell. 1927. *Principia mathematica*, 3 Volumes. Second edition. Cambridge University Press. Pages 1925-1927.

Whitelaw, M. 2004. Hearing pure data: Aesthetics and ideals of data-sound. In *Unsorted: Thoughts on the Information Arts: An A to Z for sonic ac*ts X. Amsterdam: Sonic Acts/De Balie. Pages 45-54.

Williams, S.M. 1994. Perceptual principles in sound grouping. In G. Kramer (ed.). 1994. *Auditory display: Sonification, Audification, and Auditory Interfaces*. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings, Volume XVIII. Reading, MA: Addison Wesley Publishing Company. Pages 95–125.

Winckel F. 1967. *Music, Sound and Sensation*. New York: Dover.

Wood, G. 2002. *Living dolls: A magical history of the quest for mechanical life*. London: Faber and Faber.

Wordnet 1998. *A lexical database for the English language*. Version 1.6. Princeton, NJ: Cognitive science laboratory Princeton University. Version 3.0 available online at http://wordnet.princeton.edu/perl/webwn. Accessed 13 May 2008.

Worrall, D.R. 1996. Studies in metamusical methods for sound and image composition. *Organised Sound* Volume 1, Number 3. Cambridge UK: Cambridge University Press. Pages 20-26.

Worrall, D.R. 1999. Composition as revelation. In *Proceeding of First Iteration, A Conference on Generative Computational Processes in the Electronic Arts*, Monash University, Melbourne December 1-3. Available at http://avatar.com.au/worrall/papers/ "Composition%20as%20revelation.pdf

Worrall, D.R. 2000. *The physic and psychophysics of music and sound.* Course notes, available at http://www.avatar.com.au/courses/PPofM/.

Worrall, D.R. 2004. Audification experiments using XAO returns. In *Auditory display as a tools for exploring emergent forms in exchange-trading data.* Research report to the Sonic Communication Research Group, University of Canberra, 28 October. Available at http://www.avatar.com.au/sonify/research/sonsem1/ 4audExperi.html

Worrall, D.R. 2008. Overcoming software inertia in data sonification research using the SoniPy framework. In *Proceedings of The Inaugural International Conference on Music Communication Science* 5-7 December 2007, University of NSW, Sydney, Australia.

Worrall, D.R. 2009. (upcoming). The use of sonic articulation in identifying correlation in capital market trading data. *Proceedings of the Fifteenth International Conference on Auditory Display*, Copenhagen, Denmark, May

18-22.
Worrall, D.R. M. Bylstra, S. Barrass and R.T. Dean. 2007. SoniPy: The design of an extendable software framework for sonification research and auditory display. In *Proceedings of the 13th International Conference on Auditory Display*, Montréal, Canada, June 26-29.
Wright, M., A. Freed, and A. Momeni. 2003. Open sound control: State of the art 2003. In *Proceedings of the 2003 Conference on New Interfaces for Musical Expression.* Singapore: National University of Singapore.
Xenakis, I. 1971. *Formalized Music: Thought and Mathematics in Music*. Indiana: Indiana University Press.
Xenakis, I. 1985. *Arts/Sciences: Alloys*. New York: Pendragon Press.
Yehoshua Bar-Hillel,Y. 1955. An examination of information theory. In *Philosophy of Science*, Volume 22, Number 2, April. Chicago: The University of Chicago Press. Pages 86-105.
Zatorre, R.J., P. Belin, V. B. Penhune. 2002. Structure and function of auditory cortex: music and speech. In *Trends in Cognitive Sciences*, Volume 6, Issue 1, Elsevier Science Ltd. Pages 37-46.