Motivations and Methodologies for Automation of the Compositional Process

Marcus Pearce, David Meredith and Geraint Wiggins
Department of Computing, City University, London

Abstract

Our aim in this paper is to clarify the range of motivations that have inspired the development of computer programs for the composition of music. We consider this to be important since different methodologies are appropriate for different motivations and goals. We argue that a widespread failure to specify the motivations and goals involved has lead to a methodological malaise in music related research. A brief consideration of some of the earliest attempts to produce computational systems for the composition of music leads us to identify four activities involving the development of computer programs which compose music each of which is inspired by different practical or theoretical motivations. These activities are algorithmic composition, the design of compositional tools, the computational modelling of musical styles and the computational modelling of music cognition. We consider these four motivations in turn, illustrating the problems that have arisen from failing to distinguish between them. We propose a terminology that clearly differentiates the activities defined by the four motivations and present methodological suggestions for research in each domain. While it is clearly important for researchers to embrace developments in related disciplines, we argue that research in the four domains will continue to stagnate unless the motivations and aims of research projects are clearly stated and appropriate methodologies are adopted for developing and evaluating systems that compose music.

1 Introduction

Over the past 50 years, the development of the digital computer has provided an increasingly powerful means of devising and implementing formal systems for the composition of music. Furthermore, the wide availability of computers has allowed a variety of people, among them musicians, philosophers, psychologists and computer scientists, to develop computer programs for musical composition with very different aims in mind. Although we regard as beneficial the interdisciplinary nature of research which this state of affairs has fostered, a less welcome side effect has been a state of confusion in those cases where the reasons for writing such programs have not been made explicit.

We argue that the fields of research involved have suffered as a result of this confusion. The failure to distinguish between different motivations for the development of computer programs which compose music has made it difficult for researchers to precisely and unambiguously state their aims. As a consequence, researchers often fail to adopt suitable methodologies for the development and evaluation of compositional programs and this, in turn, has compromised the practical or theoretical value of their research.

Our argument is organised as follows. In § 2 we identify four motivations that have inspired researchers to develop computer programs which compose music. The nature of the malaise that we believe afflicts computational music research is described in § 3 where we also discuss some relevant epistemological and methodological issues. In § 4, 5, 6 and 7, we consider each of the four motivations in turn. In each case, we discuss terminological issues, illustrate why it is important to distinguish research with one motivation from the others and present methodological suggestions for future research. § 8 contains a discussion of the nature of a healthy interdisciplinary research strategy. Finally, in § 9 we summarise the results of this enquiry and present our conclusions.

2 A Historical Perspective

The earliest work on the application of computers to composition was generally oriented towards practical uses. Hiller & Isaacson (1959, p. 4), for example, describe the compositional aims of the four programs they developed. The first two experiments were designed "to demonstrate that standard musical techniques could be handled by computer programming", the third to "show that computers might be used by contemporary composers to extend present compositional techniques", while the fourth experiment was intended to "show ... that computers might be used in highly unusual ways to produce radically different species of music". The programs were used to generate a single composition, *The Iliac Suite*, with four movements corresponding to the music generated in each of their four experiments. This work can be seen as part of a unique compositional experiment whose goal was to produce a single composition.

The goal of Xenakis (1971), on the other hand, was primarily to develop a mathematical language for describing and manipulating music. Ames (1987, p. 173) notes that he was one of "the first to conceive of a composing program as a *utility* capable of generating many pieces, rather than a one shot effort geared towards a specific compositional goal". Xenakis was concerned with the use of computers to produce compositional tools which might be used by any composer to generate any number of compositions.

The use of computers for theoretical research (rather than practical applications) in music appears rather later in the literature. For example, Sundberg & Lindblom (1976, p. 100) state that their purpose was "to describe facts in music theory by means of generative rule systems". They describe their use of generative grammars to provide an intensional description of a set of Swedish nursery tunes and folk songs. A second theoretical motivation for developing computer programs which compose music is drawn from cognitive science. For example, Steedman (1984, p. 75) declares that: "...generative rules are only really interesting when they can be used to drive a model of human performance" and Johnson-Laird (1991, p. 291) states his intention to develop a theory of "what the mind has to compute in order to produce an acceptable improvisation".

Thus, we can identify four different motivations that have lead to the development of computer programs which compose music and, correspondingly, four distinct activities:

1. computer programs are written by the composer as an idiosyncratic extension to her own compositional processes;

2. computer programs are written as general tools to aid any composer in the composition of music;

3. theories of a musical style are implemented as computer programs;

4. cognitive theories of the processes supporting compositional expertise are implemented as computer programs.

We do not claim to have provided a fine-grained analysis of all the different motivations involved. However, at a general level of description, we suggest that the four activities described above can account for much of the existing literature presenting programs which compose music.[1]

3 Motivations and Methodologies

3.1 A Methodological Malaise

Bundy (1990) suggests that there exists a methodological malaise in artificial intelligence research and that one important cause is a failure to distinguish different types of AI, each with its own motivations, aims and methodologies. Among the symptoms of this malaise are the widespread use of different criteria for assessing AI research and a lack of continuity between projects.

We shall argue that there exists evidence of a similar stagnation in the body of published work involved in the development of computer programs which compose music. The malaise seems to be a result of an implicit assumption

---

[1]Research projects sometimes use music as an interesting domain for testing general AI techniques (see e.g., Ghahramani & Jordan, 1997). These projects are not included in our survey since they are not primarily musically motivated and do not usually concern the composition of music specifically.

that simply describing a computer program that composes music counts as a useful contribution to research and is characterised by three features:

1. a failure to specify the precise practical or theoretical aims of research;

2. a failure to adopt an appropriate methodology for achieving the stated aims;

3. a failure to adopt a means of evaluation appropriate for judging the degree to which the aims have been satisfied.

These problems often seem to stem from a failure to distinguish between the different motivations for writing computer programs that compose music and to adopt one of these in a research project. This has the effect of making it impossible to clearly state the research aims. If the aims of the research are not clearly stated, then it is impossible to employ a suitable methodology for achieving those aims and evaluating the degree to which the resulting system achieves them. Finally, if the degree to which the system satisfies its goals is not evaluated in a satisfactory manner then the research has little practical or theoretical significance. Furthermore, it becomes difficult for the reader to judge the value of the research, in terms of what it demonstrates and why this is important. We submit that these effects have engendered a stagnant academic climate characterised by *degenerating* research programmes (in the sense of Lakatos, 1970).

We believe, with Bundy (1990), that one useful remedy for this situation is to be clear about the aims and motivations involved in a project and to adopt an appropriate methodology and appropriate criteria for evaluation. In § 4, 5, 6 and 7, we provide examples of the problems created by failing to distinguish between the different motivations for developing computer programs which compose music and present methodological suggestions for future research. Although we present criticism of existing work in order to illustrate our arguments, it is to be understood that this criticism is presented with positive and constructive intentions. Before we start, however, we must state our position regarding a number of epistemological and methodological issues.

3.2 Three Types of AI

Noting that it is possible to differentiate natural science (the study and understanding of natural phenomena) from engineering science (the study and understanding of practical techniques), Bundy (1990, p. 216) suggests that there are three kinds of AI:

1. *basic AI*: an engineering science whose aim is to "explore computational techniques which have the potential for simulating intelligent behaviour";

2. *cognitive science* or *computational psychology*: a natural science whose aim is "to model human or animal intelligence using AI techniques";

3. *applied AI*: epistemologically speaking a branch of engineering "where we use existing AI for commercial techniques, military or industrial products, i.e., to build products".

Since research in the different disciplines is guided by different motivations and aims, this taxonomy implies different "criteria for assessing research in each kind of AI. It suggests how to identify what constitutes an advance in the subject and it suggests what kind of methodology AI researchers might adopt" (Bundy, 1990, p. 219). In accordance with this analysis, Wiggins & Smaill (2000) note that the motivations for applying AI techniques to the musical domain can be divided into those concerned with understanding human musical abilities and those concerned with designing useful tools for musicians and composers.[2]

3.3 Speculative and Empirical Disciplines

Speculative disciplines are characterised by the use of deduction from definitions of concepts, self-evident principles and generally accepted propositions: "Their ultimate criterion of validity is whether they leave the reader with a feeling of conviction" (Berlyne, 1971, p. 2). Such fields as the aesthetics of music, music history and music criticism fall into this category. *Empirical* disciplines, on the other hand, derive their methodologies from science. It is important to be clear about the meaning of the term *science* since:

---

[2]Most work in artificial intelligence may be classified as applied AI.

"A great deal of confusion has arisen from failure to realise that words like the French *science* and the German *Wissenschaft* (with their equivalents in other European languages) do not mean what the English word 'science' means. A more accurate translation for them would be 'scholarship'" (Berlyne, 1971, p. 3).

In light of this, it will be helpful to state our position on the epistemological status of scientific knowledge.

In an attempt to distinguish (systems of) propositions in the scientific disciplines from those of non-scientific fields, Karl Popper developed an approach known as *methodological falsificationism*. Scientific statements must be embedded in a framework that will potentially allow them to be refuted:

"…statements, or systems of statements, convey information about the empirical world only if they are capable of clashing with experience; or, more precisely, only if they can be *systematically tested*, that is to say, if they can be subjected ... to tests which *might* result in their refutation" (Popper, 1959, p. 313–314)

In logical terms, this thesis stems from the fact that while an existential statement (e.g., 'the book in front of me is rectangular') can be deduced from a universal statement (e.g., 'all books are rectangular'), the reverse is not true. It is impossible to verify a universal statement by looking for instances which confirm that statement (e.g., by looking for rectangular books). We may only evaluate a universal statement by looking for empirical data supporting an existential statement that falsifies that statement (e.g., by looking for non-rectangular books). The demarcation criterion demands that a scientific theory must be stated clearly and precisely enough for it to be possible to decide whether or not any existential statement conflicts with the theory.

In methodological terms, falsificationism suggests that science does not consist of a search for truth but involves the construction of explanatory hypotheses and the design of experiments which may refute those hypotheses. A theory that goes unrefuted in the face of empirical testing is said to have been *corroborated*. Popper was aware that "scientific discovery is impossible without a faith in ideas which are of a purely speculative kind" (Popper, 1959, p. 25). However, he argued that the experiments designed to refute a scientific hypothesis must be empirical in nature in order for them to be intersubjectively tested. Therefore, the demarcation between scientific and non-scientific theories relies not on degree of formality or precision nor on weight of positive evidence but simply on whether empirical experiments which may refute those theories are proposed along with the hypotheses.

3.4 Cognitive Science

Cognitive science is a natural science whose goal is to understand how the mind works and the approach taken to reaching this goal may be characterised by several methodological heuristics. First, a functional level of description is considered sufficient for the development of theories of cognition (Chalmers, 1994; Johnson-Laird, 1983; Pylyshyn, 1989). Second, experimental studies of human behaviour are used to constrain hypotheses that are made about cognitive processes (Johnson-Laird, 1983; Newell & Simon, 1976). Third, theories are defined as *effective procedures* and implemented as computer programs to ensure that any assumptions are made explicit (Johnson-Laird, 1983; Longuet-Higgins, 1981; Simon & Kaplan, 1989). Finally, the theory is evaluated through empirical comparison of its behaviour with that of humans (Newell & Simon, 1976; Simon & Kaplan, 1989).

Marr (1982) introduced a framework for the understanding of complex information processing systems such as the mind/brain which has proved highly influential in modern cognitive science. Noting that different properties of such systems can only be understood at different levels of description, Marr isolates three general and independent levels at which questions may be asked about an information processing system: the computational level; the representational and algorithmic level; and the hardware level.

The first level asks *what* and *why* questions about the system: What is the goal of the computation? Why is it appropriate? And what is the logic of the strategy by which it can be carried out? The computational level theory attempts to describe the intrinsic nature and computational requirements of a cognitive task through a formal analysis of the various outputs resulting from different inputs. Through understanding the nature of the problem to be solved appropriate constraints may be placed on a cognitive model. It is only at the second level of analysis that we turn to the question of *how*, which involves specifying a *representation* for the input and output of the computation and an *algorithm* by which the computation may be achieved. The final level outlined by Marr concerns the physical realisation of the representation and algorithm.

Marr (1982) goes to great lengths to emphasise the importance of the computational level theory. For any particular problem there is usually a wide range of representation schemes and algorithms that could be used. He argues that these depend to a far greater extent on the constraints derived from analysing the problem at the first level of analysis than on the particular hardware mechanisms upon which their solutions are implemented. Speaking of human perception, he notes that "trying to understand perception by studying only neurons is like trying to understand bird flight by studying only feathers: it just cannot be done" (Marr, 1982, p. 27).

McClamrock (1991) argues that, independently of the types of question asked about a cognitive system, approaches vary in terms of the level of organisational abstraction under consideration. At any level of organisational abstraction, a cognitive theory specifies how functional components of a higher-level explanation may be decomposed into those at a lower level of abstraction. Thus, there are two basic issues to be addressed in any cognitive scientific research:

1. specify the kinds of question (computational, algorithmic/representational or implementational) that the research will address;

2. specify a level of organisational abstraction which is the prime focus of the research.

4 Algorithmic Composition

4.1 Motivation

Many who write programs for the composition of music are motivated by artistic goals: computer programs are used to generate novel musical structures, compositional techniques and even genres of music. Berg (1987, p. 161) writes that the fundamental contribution of the computer to music is that it empowers the composer to "hear that which could not be heard without the computer, to think that which could not be thought without the computer and to learn that which could not be learned without the computer. The computer can allow a composer to write music that goes beyond that which she is already capable of". An example of this motivation is provided by David Cope:[3]

> "EMI was conceived . . . as the direct result of a composer's block. Feeling I needed a composing partner, I turned to computers believing that computational procedures would cure my block." (Cope, 1991, p. 18)

We suggest that the existing term *algorithmic composition* should be applied exclusively to activities in which the motivations have this artistic character. When a computer program is used in this way, we must consider the program to be an integral part of the compositional process. The composer may use an existing computer program or she may write a program herself: since identical motivations are involved, we count both of these as algorithmic composition.

4.2 Methodology

If a computer program is written by the composer, the development of the program is an integral part of the compositional process (since ultimately it is driven by the same motivations). Since the program is designed solely for use by its developer, there are no methodological constraints placed on its construction. Furthermore, there is no need to define any rigorous criteria for success nor to use such criteria in evaluating the program and the compositions. If the composer intends the music for public consumption, then they may only be evaluated in the same way that composers and compositions are usually appraised: through audience reactions at performances, record sales, critical reviews and so on.

Finally, we suggest that if a composer publishes work on their techniques for algorithmic composition, they must demonstrate the practical or theoretical significance of the work. Many of those developing computer programs which compose music with other (non-artistic) motivations are often themselves musicians. That they should have artistic

---

[3]A more detailed analysis of Cope's work is beyond the scope of this paper. However, we believe that the relevance of his work for cognitive science or musicology is significantly compromised by his failure to clearly state the cognitive or stylistic hypotheses embodied in EMI and to adopt an appropriate methodology for refuting those hypotheses.

or aesthetic concerns seems almost unavoidable. It is imperative, from a methodological point of view, that these concerns should not be allowed to interfere with the achievement of the goals of the research. We hope that this will become apparent as we discuss the other motivations.

5 Design of Compositional Tools

5.1 Motivation

We now turn to projects whose motivation is to design compositional tools for use by composers other than the designer(s) of the system. Leman (1992, p. 296) uses the term "AI-Music" to refer to the discipline concerned with "the development of computer music systems for different practical usages of musical knowledge and musical communication". Since AI itself comes in several different flavours (see § 3.2), this rather vague term does not adequately distinguish this activity from the others. The importance of doing so derives from the fact that particular methodologies are required for designing compositional tools and evaluating whether they satisfy their design objectives. We suggest the term *design of compositional tools* fulfills this function rather better.

5.2 Methodology

The design of a system which composes music for use in algorithmic composition may be classified as applied AI and it should therefore adopt an appropriate methodology from the disciplines of engineering or design. Unfortunately, most reports of research into tools for algorithmic composition simply satisfy themselves with describing the system.

Let us illustrate the problem with an example. Biles (1994) describes a genetic algorithm (GA) which improvises jazz melodies over a supplied chord progression. He introduces his system (called *GenJam*) as "a genetic algorithm-based model of a novice jazz musician learning to improvise" (Biles, 1994, p. 131). This rather vague statement is ambiguous regarding both the motivations and aims of the research. It is only in the final paragraph that we find a suggestion that his intentions were to design a compositional tool when he concludes that 'GenJam shows that GAs can be a useful tool for searching a constrained melodic space" (Biles, 1994, p. 137). This statement is strangely at odds with the manner in which GenJam is introduced.

As a result of his failure to begin with a clear description of the motivations and goals of the project, Biles is unable to adopt an appropriate methodology and evaluation procedure. Regarding the latter, he simply notes that "almost all the new phrases sound reasonable" (Biles, 1994, p. 136). This statement gives the reader no indication of the usefulness of GenJam as a compositional tool (in the sense defined above) and does not justify his conclusion.

There are four fundamental stages in most software engineering methodologies (Pfleeger, 1991). These are as follows:

1. *analysis*: the functional requirements of the system are specified – *what* it is to achieve and *why*;

2. *design*: the details of *how* the requirements of the system may be implemented are specified;

3. *implementation*: the application is built;

4. *testing*: tests are run to evaluate whether the software meets all its requirements.

The first stage contains an explicit statement of what the software is to achieve, why this is important and the criteria for success. It describes the high-level functional ontology of the system to be developed that is independent of architectural or implementational concerns and both typical and atypical usage scenarios that the software must be capable of handling. The end result of this stage is a detailed *requirements specification*.

As we have seen, Biles (1994) failed to present a clear analysis of the requirements GenJam was intended to satisfy. Other authors have been more explicit about their goals. Mozer (1994), for example, describes several limitations of Markov transition tables for algorithmic composition and declares that his aim is to show "how these limitations may be overcome . . . using connectionist techniques" (Mozer, 1994, p. 248). In the various experiments reported he consistently compares the performance of his neural network systems to that of second and third order transition tables. The clarity with which Mozer specified his aims allowed him to employ an appropriate methodology for achieving those aims. Consequently he was able to provide a relatively comprehensive analysis of the limitations of

using sequential neural networks for algorithmic composition and detailed suggestions for future research. However, he states neither the scenarios which the system is intended to cover nor the conditions under which the system will be deemed successful. This appears to be fairly common in the literature.

Regarding the apparent failure of many studies to link their research to a particular problem or aspect of algorithmic composition, we suggest that the field would benefit from greater interaction with composers. Researchers at IRCAM in Paris, for example, often work together with composers on their products (see e.g., Assayag et al., 1999). The composers are able both to supply ideas for research and to provide feedback on the development of applications. The detailed analysis of composers at work can be a useful source of information about typical compositional problems. Polfreman & Loomes (2001), for example, used *task analysis* techniques to study the process of composition. Their goal was to improve user-interface design in compositional tools by acquiring a deeper "understanding of the nature of music composition tasks, the environment within which such tasks are typically carried out and how these tasks are organised collectively" (Polfreman & Loomes, 2001, p. 75).

The design phase is concerned with how the requirements may be achieved in an implemented system. The end result of this phase is a detailed *architecture specification* which describes the proposed architecture in terms of its various software components, how these components interface with one another and the interactive behaviour of the components. An *implementation plan* is also developed which establishes a detailed schedule for the completion of the system as well as a list of critical tasks that must be accomplished in the implementation phase. The final part of the design phase concerns the development of a detailed *test plan* which defines all the tests that the software must pass for it to be declared complete. Typically such tests are related to the performance of the software in the typical and atypical scenarios presented in the analysis stage. The end result of this phase is a detailed *architecture specification*.

In the implementation phase, the requirements and architecture specifications are used to write the code for the various components of the system. The end result of this phase is the completed system. Typically in the literature describing tools for algorithmic composition, the architecture of the completed system is described (with varying levels of specificity). However, there often seems to be little evidence of a design phase in which the architectural choices and decisions made are related to the requirements the software is intended to fulfill.

The final stage of development is concerned with evaluating the *quality* of the completed system in relation to its requirements specification. There are typically four types of testing that are used: first, *internal tests* which ensure that all the low-level functions or components of the software behave as desired; second, *unit tests* which ensure that the user-visible behaviour of the software as a whole conforms to the requirements; third, *application tests* examine the performance of the system on all the typical and atypical scenarios described in the requirements specification; and finally, *stress testing* evaluates the performance of the system under a range of environments that are more stressful than those the application would normally encounter.

Depending on the precise design objectives not all these kinds of test need be carried out. However, the testing of tools for algorithmic composition is frequently neglected in the literature. When the software is evaluated, a paper will typically conclude with a comment such as "these improvisations are still quite pleasing musical forms" (Kohonen et al., 1991, p. 240) or "almost all the generated individuals were pleasant to listen to" (Johanson & Poli, 1998, p. 185). This is unsatisfactory for two reasons: first, evaluating the music produced by the system reveals little about its utility as a compositional tool; and second, qualitative and subjective evaluation by the designers of the system reveals little about the value of the tool to other composers.

6 Computational Modelling of Musical Styles

6.1 Motivation

We turn now from practical motivations for writing computer programs which compose music to theoretical ones. Roads (1985, p. 175) suggests the term "generative modelling of music" to describe those activities where the driving motivation is not to compose aesthetically pleasing pieces of music nor to design a useful compositional tool but to propose and verify hypotheses about the stylistic attributes defining a body of works. The term *generative* is derived from computational linguistics and describes a formalism (usually a grammar) which *explicitly* enumerates and assigns structures to all and only the well-formed sentences in the language it is intended to describe using a finite set of rules:

"If the grammar is ... perfectly explicit ... we may ... call it a *generative grammar*" (Chomsky, 1965, p. 4). However, a generative theory is often misinterpreted as one which is intended to describe mental processes. For example, Baroni et al. (1992, p. 188) state that a generative analysis "observes some types of regularity present in the organisation of musical events and formulates hypotheses about the processes of thinking inherent in this organisation". Therefore, we suggest that the term "generative modelling of music" be avoided since it has the potential to create confusion about the activities it refers to.

Ames (1992, p. 55) makes a distinction between "active style synthesis" (algorithmic composition in our terminology) and "empirical style modelling", which he defines as the development and use of computer programs to determine the "necessary criteria for generating known musical styles". However, this term captures neither the musical object of study nor the use of computational procedures that are fundamental to the activity. We suggest the term *computational modelling of musical styles* to describe those activities where the motivation is to propose and evaluate hypotheses concerning the important stylistic properties of a corpus of musical compositions and to distinguish this activity from those with other motivations. Since the objects of study in the computational modelling of musical styles are existing musical works, we consider the discipline to be a branch of musicology.

6.2 Methodology

Ames & Domino (1992, p. 186) state that their Cybernetic Composer was developed "to show how artificial intelligence can be applied to musical composition". But Ames also argues that:

> "composing programs such as CHORAL [Ebcioğlu 1988, 1992] and CYBERNETIC COMPOSER [Ames & Domino 1992] ultimately serve an analytic purpose ... The difference between active style synthesis [algorithmic composition] ... and empirical style modelling [computational modelling of musical styles] is simply the extent to which the generative criteria have been devised 'before' or 'after' the fact" (Ames, 1992, p. 55).

Ebcioğlu (1988) also appears to have both theoretical and practical motivations when he describes CHORAL as "a rule-based expert system ... for harmonisation and Schenkerian analysis of chorales in the style of J.S. Bach" (p. 43) and states that he hopes the system "will be of use to researchers in algorithmic composition" (p. 49).

We reject Ames' implied claim that the differences between algorithmic composition and computational modelling of musical style are negligible. As we have seen, computational modelling of musical styles may be distinguished from both algorithmic composition and the design of compositional tools on the basis of their motivations. Once again, the importance of making such distinctions depends on the fact that different methodological approaches are required in each of the different disciplines. In order to make this argument concrete, we shall consider the examples cited by Ames (1992) in a little more depth.

First, we must be clear about the benefits afforded by the computational approach. The implementation of stylistic hypotheses as a computer program (which can generate music) has two potential advantages (Camilleri, 1992; Sundberg & Lindblom, 1976, 1991). First, while musicology has traditionally adopted a speculative methodology, the computational approach requires that all assumptions included in the theory (self-evident or otherwise) are explicitly and formally stated. The second potential advantage is that the implemented model may be evaluated through comparison of the compositions it generates with the human-composed pieces which the theory is intended to describe. We shall consider each of these advantages in turn.

Formality

Both Ebcioğlu (1988, 1992) and Ames & Domino (1992) describe in some detail how their programs emulate the intended style. Since the programs embody their stylistic hypotheses, we may say that they are presenting their theories. However, we feel that the detailed description of a computer program which generates compositions in a particular style is not necessarily the most perspicuous means of presenting a theory of musical style. Ebcioğlu (1992), for example, spends half of his chapter providing a detailed description of the logic programming language which he developed "so that [he] could use it to generate very high quality music in a reasonable time" (Ebcioğlu, 1992, p. 296). It must be noted that the description of his theory of harmonisation in the style of Bach suffers from a lack

of detail as a result.[4] Similarly, Ames & Domino (1992, p. 190) state that "the program also biases pitch-selection with statistical feedback ... so that melodies do not linger unintentionally on particular scale degrees". It is unclear whether this aspect of the program is part of the stylistic theory or simply a detail of the implementation that helps the program produce acceptable music.

In both cases, the stylistic hypotheses embodied in the program are not clearly and precisely stated, nor are they distinguished from aspects of the program which are not intended to be part of the theory. This seems to have arisen from the failure to make a sharp distinction between algorithmic composition or the design of compositional tools and computational modelling of musical styles. Unless the theory that is implemented in the composing program is clearly and precisely stated, and separated from implementational or control details, the advantages afforded by the computational approach in terms of formality and explicitness are lost.

Evaluation

It is often claimed that the computational approach to modelling musical styles allows hypotheses to be scientifically evaluated (see e.g., Ames & Domino, 1992; Camilleri, 1992; Laske, 1988; Roads, 1985; Sundberg & Lindblom, 1976, 1991). Sometimes, this appears to be associated with the implicit assumption that the formality and precision required of a computer model of itself makes an implemented theory scientific. As we have seen in § 3.3, this is not the case. Even when it is acknowledged that the scientific significance of implementing a theory is to enable its evaluation, empirical experiments that may refute the theoretical hypotheses are rarely proposed.

Ames & Domino (1992, p. 186), for example, claim that "With the arrival of digital computers and artificial intelligence, it now becomes possible to subject theoretical speculations about music to rigorous empirical evaluation". However, all they have to say about the compositions produced by Cybernetic Composer is that they "are realistic enough that an unknowing listener cannot discern their artificial origin" (Ames & Domino, 1992, p. 186). Such anecdotal and subjective evaluation of computational models of musical styles is common in the literature. Authors typically conclude with a sentence such as "[the program] seems to be capable of producing musical results" (Ebcioğlu, 1988, p. 49) or "The general reactions of Swedes listening to these melodies informally are that they are similar in style to those by Tegnér" (Sundberg & Lindblom, 1976, p. 111).

There would appear to be two solutions to this conflict between theory and practice. The first is to declare that musical styles are not an appropriate topic of scientific research. According to this view, computational methods may be incorporated into speculative arguments concerning musical styles, but adopting a scientific approach to the empirical corroboration is inappropriate. If, on the other hand, we view the computational modelling of musical styles as a scientific activity, it is important to consider how such a model would be refuted.

Meredith (1996) proposes that a theory for a musical style should take the form of an *algorithmic style theory* which he defines to be a hypothesis that a *composing algorithm* generates all and only those pieces that are in the style being modelled. Taking *weak generative capacity* rather than *descriptive adequacy* as a criterion for refutation, he notes that a theory of a musical style may be falsified in two ways: first, it may fail to generate compositions that are in the style (*undergeneration*); and second, it may generate compositions which are not in the style (*overgeneration*).[5]

Meredith points out that one cannot test such a theory for undergeneration or overgeneration unless one has an algorithm for deciding for any given piece whether or not it is 'in the style' that one is attempting to model. He calls such an algorithm an *acceptability algorithm* and proposes that a piece should be defined to be *in the style of* some well-defined corpus of pieces if and only if it is either a member of this corpus or it is determined by means of the acceptability algorithm to be in the style of this corpus. Meredith (1996, p. 30–32) suggests that the acceptability algorithm could take the form of an experiment in each trial of which a subject is presented with the piece whose stylistic acceptability is to be determined together with a randomly selected sample of pieces from the corpus. The subject's task is to identify the test piece. Standard statistical techniques can then be used to determine whether or not the subjects are able to identify the test piece with a frequency that is significantly better than that which would be expected if they were making their selections at random. Clearly, if they can do this with a sufficiently high frequency,

---

[4]Ebcioğlu (1987) describes the musical theory in more detail in an IBM research report. However, following common scientific practice, we take the peer reviewed and published papers as representative of Ebcioğlu's attempt to communicate the practical or theoretical value of his work to the academic community.

[5]The terms overgeneration and undergeneration are taken from Johnson-Laird (1991).

then the piece is stylistically distinguishable from the corpus pieces and thus must be defined to be *not in the style*.

To test such a theory for undergeneration, one takes a piece that is in the style (i.e., either in the corpus or determined to be in the style by the acceptability algorithm) and gives the piece as input to a *parsing algorithm* which proves whether or not the piece is a member of the set of possible outputs from the composing algorithm. To test the theory for overgeneration, one uses the composing algorithm to generate a new piece and then determines whether or not this piece is in the style using the acceptability algorithm.

An algorithmic style theory constructed according to Meredith's specification is always either true or false and always either refuted or unrefuted. Such a theory can never be verified but Meredith claims that such a theory should be tested for overgeneration and undergeneration until it is refuted.

A final issue to be considered concerns the question of how we might choose one unrefuted theory of a style over another: "imagine that we have two grammars of a well-defined set of compositions whose outputs and data coverage are identical but that go about their descriptive tasks in different ways. How do we decide between these descriptively equally successful grammars?" (Sundberg & Lindblom, 1991, p. 269). Sundberg & Lindblom suggest that we should prefer the grammar for which there is more "independent motivation". They argue that this may be interpreted as independent support for the explanatory adequacy of the grammar as measured by its learnability. This does not seem appropriate for two reasons. First, since procedures have not yet been widely applied to corroborating the weak generative capacity of computational theories of musical styles, the development and testing of an explanatory theory make this unlikely to prove a practically useful measure (Meredith, 1996). Second, this would take a theory of musical style firmly into psychological territory.

We propose that independent evidence for two unrefuted computational theories of musical style might be practically found by considering the predictions they make about issues commonly addressed in musicology. One example would be the ability of the models "to distinguish ...structures typical of particular epochs and also ...structures belonging to particular repertoires" (Baroni et al., 1992, p. 187). For example, Hild et al. (1992) have developed a hybrid model (called HARMONET) of chorale harmonisation.[6] HARMONET has been used to extract stylistic features from harmonisations by composers such as J. S. Bach. Furthermore, independent support has been found for predictions made by the model. For example, Hörnel & Olbrich (1999) used HARMONET to predict that a certain chorale harmonisation (attributed to Bach) had not been composed by Bach himself since it coincided to a greater degree with the analysis of an earlier style of harmonisation than with the analysis of Bach's own chorale harmonisations. This prediction was confirmed in the musicological literature.

7 Computational Modelling of Music Cognition

7.1 Motivation

The motivations of authors such as Steedman (1984) and Johnson-Laird (1991) were drawn from cognitive science rather than musicology (see § 2). They were not primarily concerned with whether their systems generated pleasing music or whether they functioned as useful compositional tools. Crucially they were not attempting to propose theories of a musical style and were interested only in the degree to which their models help us understand the underlying cognitive processes involved in human composition.

As we noted in § 6, this kind of project has sometimes been grouped together with research in the computational modelling of musical styles under the generic rubric *generative modelling of music* (Roads, 1985). Laske (1988, p. 43) proposes that the object of study in "the new science of cognitive musicology" is "musical intelligence as a distinctive human faculty". This term is clearly intended to capture projects with cognitive motivations as well as those with musicological goals. However, the motivations involved in the two types of project are separable and furthermore, "cognitive models need not reflect current music-theoretic constructs, nor must models of musical knowledge have cognitive pretensions" (Desain et al., 1998, p. 152). We prefer the term *computational modelling of music cognition* (Desain et al., 1998) to describe specifically those activities where the factors motivating the development of a computer program which performs musical tasks are drawn from cognitive science.

---

[6]The system uses both neural networks and a symbolic constraint satisfaction system.

7.2 Methodology

Again, the importance of making this distinction is methodological. Although the potential advantages of implementing a cognitive theory are analogous to the potential advantages of implementing a theory of a musical style, different methodologies are required for the implementation and evaluation of the theory. We shall present an example to illustrate the point. Baroni et al. (1992, p. 189) declare that none of their grammars "should be thought capable of describing the actual mental processes implied in musical composition". However, in a dramatic reversal of opinion, (Baroni, 1999, p. 9) states of one of their grammars that "syntactic rules can be conceived in any case as real cognitive processes". To justify this claim, he simply notes that "the technical procedures of a computer do not coincide with those of a human composer, but it is quite plausible to think that analogous processes do exist" (Baroni, 1999, p. 9). He fails to:

1. clearly state the hypotheses embodied in the model concerning the cognitive mechanisms of composition, in particular, the degree of functional organisation they address and kinds of question they pose;

2. derive these hypotheses from the results of psychological experiments on musical composition as opposed to musicological analysis of the corpus;

3. evaluate these hypotheses through systematic and empirical attempts to refute them based on the compositions produced by the model.

Unless such methodological requirements are met, Baroni's claim is void of any significance for cognitive science (see § 3.4). We now consider each of these three issues in further depth.

Stating Cognitive Hypotheses

A major problem with implementing cognitive models of musical composition is our lack of knowledge about the cognitive mechanisms underlying composition. This lack of understanding stems partly from the difficulty of probing such a complex cognitive process as composition and partly from the individual variation inherent in the process. Given this lack of understanding, we suggest that a useful point of departure would be to ask questions at the computational level (see § 3.4).

An example of this approach can be found in Johnson-Laird (1991) who developed a theory of jazz improvisation. On the basis of an analysis of different computational systems, he draws various conclusions concerning what the mind has to compute in order to produce acceptable improvisations in the jazz idiom. In contrast to Baroni (1999), Johnson-Laird states clearly the nature of his cognitive hypotheses and takes pains to present the results of his enquiry in terms of constraints imposed on the cognitive processes responsible for competent improvisation. For example, his analysis suggests that while a regular grammar (or formally equivalent procedure) is adequate for computing the contour, onset and offset of the next note, a computational procedure equivalent in power to a context-free grammar is required to compute its pitch.

Using Experimental Data

Johnson-Laird (1991) presents his conclusions at an extremely high level of functional organisation. However, Desain et al. (1998, p. 156) have argued that "a comparison based on global input/output behaviour (i.e., functional equivalence) is too coarse to make substantial claims about the psychological validity of the model". Ideally, we would like to relate the functional organisation of the model at a slightly lower level of abstraction to that of human cognition. How might we generate hypotheses regarding the finer grained functional organisation of cognitive processing in musical composition?[7]

Sloboda (1985, ch. 4) notes that there exist four main sources of information for those interested in the psychology of musical composition:

1. the examination of the history of a particular composition as displayed in the composer's written manuscripts;

---

[7]Note however that the model is still concerned with questions of the computational variety.

2. the examination of what composers say about their own compositional processes;

3. the live observation of composers during a session of composition;

4. The observation and description of improvisatory performance.[8]

The first of these protocols suffers from its failure to document a complete record of the internal processes of the composer (to varying degrees depending on the composer) and can only provide information specific to a particular composition (or composer). The second, on the other hand, has a more direct handle on the internal cognition of the composer but suffers from the problem of subjectivity of introspective reports and the fact that much of our cognitive processing is unavailable for conscious introspection. The last two sources of information, however, have the benefits of allowing a more objective analysis of all aspects of the compositional process in a controlled environment.

Sloboda (1985) suggests two lines of research regarding the live observation of composers at work. The first involves the detailed study of composers in a simple compositional environment in which all the cognitive steps which underlie the composer's choices are specified. Such an analysis would "lead naturally to some form of computer simulation to test the adequacy of one's theory" (Sloboda, 1985, p. 138). Laske (1992), for example, describes his work using task based protocols to acquire non-verbalisable compositional knowledge. The goal of this research was to answer questions which ask *why* various compositional decisions were made. The strategy employed is to use composer's *behaviour* on a computer based compositional task to produce a *performance trace* and thereby derive hypotheses concerning the *cognitive schemata* underlying that behaviour.

Polfreman & Loomes (2001) suggest that their methodology (see § 5) could be used as a framework for cognitive research relating to musical composition. Although similar methodological frameworks may be used for knowledge acquisition in both the design of compositional tools and the computational modelling of music cognition, it is important to distinguish the two. In the former, the relevant knowledge pertains to typical compositional problems and the usefulness of software tools which attempt to solve those problems, while in the latter, the relevant knowledge concerns the cognitive processes of the composer.

The second approach involves asking a range of musicians of varying ability and experience to undertake the task. Through comparative analysis of the results, "it might be possible to specify some of the ways in which compositional skills change with experience" (Sloboda, 1985, p. 138). Davidson & Welsh (1988) and Colley et al. (1992), for example, have compared the strategies used by expert and novice composers on simple compositional tasks.

Evaluation

The purpose of implementing a cognitive theory is partly to enable the evaluation of the theory based on a detailed empirical analysis of the similarities and differences between its behaviour and the human behaviour it is intended to explain (see § 3.4). While in the computational modelling of musical styles, we are concerned with the weak generative capacity of the model, in the computational modelling of music cognition, we are interested in evaluating the behaviour produced by the particular hypotheses under investigation.

The hypotheses defined by Johnson-Laird (1991) concerned the computational requirements of the entire task. If, on the other hand, the hypotheses concern a more detailed level of functional description, then we must extend the definition of behaviour for the model beyond input-output specification. Desain et al. (1998) illustrate this point with a hypothetical model of metre perception which decomposes the task into two sequential sub-processes: initial establishment of a beat and the subsequent tracking of it during tempo changes. They suggest that it would be possible to measure the changes in behaviour (e.g., the manner in which the range of timing disturbances that do not throw the model off-track changes) as the model switches from the first to the second stage of processing. This data could then be compared with the performance of humans on the same stimuli in an attempted refutation of such a model of the functional organisation of the cognitive mechanisms involved in meter perception. If the human data varied in the same manner, this would count as evidence corroborating the model.

Can a similar approach be used to evaluate a cognitive theory of musical composition? Very few computational models of musical cognition have been proposed. In the rare cases when cognitive aspirations are stated, the issue of evaluation seems to have been largely ignored (Hodgson, 1999; Ramalho & Pachet, 1995; Reis, 1999). Johnson-Laird

---

[8]It seems likely that the computational analysis presented by Johnson-Laird (1991) would have benefitted from the live observation of jazz improvisers.

(1991, p. 317) simply notes that "The program performs on the level of a competent beginner" and gives an informal account of how the program undergenerates and overgenerates. In short, "the gap between the cognitive work, that brings forward these models, and the experimental approach, that should validate them, is huge" (Desain et al., 1998, p. 154). This lack of emphasis on evaluation has the effect of making it very difficult to intersubjectively compare and contrast different theories.

There are at least two ways in which compositional systems may be evaluated as cognitive models: first, through an analysis of their behaviour during composition; and second, through an analysis of the music they compose. Regarding the former, it would be possible to keep a trace of the program's compositional behaviour and then compare it to traces of human composers performing the same compositional task. This bears a certain resemblance to *protocol analysis*, a standard methodology in cognitive science (Simon & Kaplan, 1989). However, it would be hard to define an appropriate level of abstraction at which to make the comparison: we would not expect the compositions themselves to be the same at the note level. However, a number of studies have demonstrated broad strategic differences between composers (Folkestad et al., 1990) and related such differences to degrees of compositional expertise (Colley et al., 1992; Davidson & Welsh, 1988). A basis for the comparison of cognitive models with human data might be found at this level of general strategy.

We turn now to the second approach. Pearce & Wiggins (2001) have recently reported preliminary work towards a framework for the evaluation of machine-generated compositions. Here we shall propose how the framework may be used to refute or corroborate hypotheses about the cognitive mechanisms supporting musical composition. The framework has several phases. First, it is necessary to precisely define a corpus of compositions; the cognitive model is intended to account for some aspect of the compositional competence able to generate that corpus. Second, the goals of the research must be clearly stated as hypotheses regarding the cognitive mechanisms involved in composing pieces in the corpus. The nature of the hypotheses define the level of abstraction at which the cognitive model is to be evaluated. Third, a program is constructed (using either a machine learning or a knowledge engineering approach) which embodies these hypotheses at an appropriate level of functional abstraction and is used to generate a test set of compositions. Any assumptions guiding the choice of corpus, the development of the hypotheses and decisions made during the development of the representation scheme and algorithms employed by the program are potential sources of bias and "must be made explicit, as they also determine what conclusions may legitimately be drawn from the results of the experiments" (Widmer, 2000, p. 72).[9]

A central feature of the framework is the design of experiments which may refute claims (based on the stated goals) made about the system generated compositions. In order to illustrate the nature of the experimental procedures involved, we shall consider a concrete example involving a hypothesis stating that the ability to flexibly move between different representations of the compositional problem (e.g., by considering different combinations of tonal, rhythmic and dynamic constraints) is characteristic of expertise (Davidson & Welsh, 1988).[10]

Two programs are generated from the corpus, one of which represents multiple features of musical events and one which does not. Sets of experimental stimuli are composed consisting of equal numbers of compositions generated by the first and second programs. Subjects are required to listen to the experimental stimuli and asked to judge the success of each composition on a scale between one and five. This is clearly a rather simplistic measure of expertise but it will serve for the purposes of illustration.[11] In the final phase of the evaluation, statistical methods are used to test the hypothesis that the mean success rating of the compositions produced by the first program are higher than that of those produced by the second program. If this test fails to reject the hypothesis, we may consider the conjecture that the ability to represent multiple features of musical events is characteristic of expert composers to have been scientifically corroborated *given the assumptions made in designing the model*. It must then be tested using different kinds of model embodying different assumptions. Otherwise, the hypothesis is considered to have been refuted *given the assumptions made in designing the model*. Consequently, the theory or the assumptions of the model must be changed, re-implemented and re-evaluated.

---

[9] One criticism that may be levelled at Johnson-Laird (1991) is his failure to be explicit about the body of musical pieces from which his computational level theory was derived. This means that it is hard to know precisely what corpus of jazz compositions the theory is able to account for.

[10] Note that this example is provided for illustrative purposes only.

[11] Davidson & Welsh (1988) found that this measure showed a high correlation with the experience of the students of composition in their study.

Besides allowing hypotheses about the cognitive processes involved in musical composition to be refuted, the framework has the advantages of placing no restrictions on the genre of music to be modelled, the cognitive hypotheses proposed or the computational techniques used. Once the theory has been corroborated at one level of functional organisation, it may be broken down to a finer level. In the case of our example, this might mean stating hypotheses concerning the particular kinds of feature that are represented. These hypotheses are then implemented and evaluated in the same manner.

In the computational modelling of music cognition,

> "proposing a new model ... can hardly be seen as a contribution to the field anymore. Recently a methodology has been emerging in which a working computational model is seen much more as the starting point of analysis and research, than as an end product ... This approach needs further elaboration, because it promises a way out of the present stagnation. A computational model is thus no longer an aim unto itself, but a means to compare and communicate theories between different research communities" (Desain et al., 1998, p. 153).

We have argued that our framework can facilitate the evaluation, improvement and comparative analysis of cognitive models of compositional processes.

8   Interdisciplinary Research

Countless authors have stressed the importance, indeed the necessity, of an interdisciplinary approach to both theoretical and practical research in music. Desain et al. (1998), for example, note that the processing and representation of musical structures can provide a common ground for research between disciplines. However, they are careful to distinguish the roles of different disciplines:

> "Such structures can be stated formally or informally within music theory, their processing can be investigated by experimental psychology, both of these aspects can be modelled in computer programs and can be given an architectural basis by neuroscience." (Desain et al., 1998, p. 153)

Others, however, have failed to adequately distinguish different disciplines on the basis of their motivations and methodologies. Laske (1988), for example, includes all four of the different activities we have discussed in the discipline of "cognitive musicology". Leman (1992, p. 296) goes so far as to suggest that "the approaches currently developed in computer-oriented music research might merge into an interdisciplinary field called 'semiotics of music'". We have argued that such agglomeration of disciplines can only serve to increase the methodological malaise that already afflicts music related research.

In contrast, we suggest that each of the fields we have described should *embrace* rather than become one with the others (see Gjerdingen, 1999). We agree that work inspired by any one of our four motivations may have implications for, or be inspired by, research taking any of the other positions. Indeed, we welcome such an interdisciplinary state of affairs. However, since each of the motivations implies different methodologies and criteria for assessment, these relationships should not be made part of the main body of a research project. If two or more of the stated motivations are adopted for any one project, the different goals and motivations should be clearly differentiated and appropriate methodologies adopted for each of them.

9   Synthesis and Conclusions

We have identified four general motivations which have inspired the development of computer programs for the composition of music. These motivations, the fields they define and their parent disciplines are shown in Table 1. We consider it imperative that developers of such systems be clear about their motivations. This would have potential benefits both for the research itself and for others interested in their work. We have seen that the appropriate methodologies for the development and evaluation of programs for the composition of music vary greatly depending on the reasons for designing the system. Moreover, unless the motivations and aims of the research are stated and appropriate methodologies and assessment procedures adopted, it is hard for other researchers to appreciate the practical or

Domain	Activity	Motivation
Composition	Algorithmic Composition	Expansion of compositional repertoire
Software Engineering	Design of compositional tools	Development of tools for composers
Musicology	Computational modelling of musical styles	Proposal and evaluation of theories of musical styles
Cognitive Science	Computational modelling of music cognition	Proposal and evaluation of cognitive theories of musical composition

Table 1: Motivations for Developing Computer Programs Which Compose Music

theoretical significance of the work. This, in turn, hinders the communication of theories between different academic communities and the comparison of different theories and practical applications. We have argued that a failure to distinguish the different reasons for developing programs which compose music has encouraged the stagnation of the fields of research involved.

In order to help distinguish different kinds of research, we have introduced a terminology which we believe better reflects these categories and may be used to avoid confusion about the purpose of developing compositional systems. Furthermore, for each motivation we have suggested appropriate methodological approaches for achieving the goals of the research and evaluating its success. We have resisted the temptation to categorise further the variety of motivations involved. Our concern here has been to differentiate motivations drawn from different disciplines and discuss the methodological issues that this differentiation raises. At this general level the four motivations discussed may be considered to be relatively autonomous. The more subtle differences in motivation present in the literature are more likely to be drawn from the same discipline. Finally, although we consider the four domains and their associated motivations to be relatively separable we do not deny the importance of interdisciplinary research. However, while a project may be inspired by research in any of the four fields, such interdisciplinary relations must be distinguished from the primary motivations and goals of a research project.

Although we have focused on the development of computer programs which compose music, the issues raised and the arguments presented are of potentially much wider relevance to practical and theoretical music research. The use of computational techniques in music research is a relatively recent activity. We have provided suggestions for countering the current stagnation in the fields discussed and putting each of them on a sound epistemological and methodological footing. We consider this to be essential if our research is to progress.

Acknowledgements

References

Ames, C. (1987). Automated composition in retrospect: 1956–1986. *Leonardo*, *20*(2), 169–185.

Ames, C. (1992). Quantifying musical merit. *Interface*, *21*, 53–93.

Ames, C. & Domino, M. (1992). Cybernetic Composer: an overview. In M. Balaban, K. Ebcioğlu, & O. Laske (Eds.), *Understanding Music with AI: Perspectives on Music Cognition* (pp. 186–205). Cambridge, MA: MIT Press.

Assayag, G., Rueda, C., Laurson, M., Agon, C., & Delerue, O. (1999). Computer assisted composition at IRCAM: from PatchWork to OpenMusic. *Computer Music Journal*, *23*(3), 59–72.

Baroni, M. (1999). Musical grammar and the cognitive processes of composition. *Musicæ Scientiæ*, *3*(1), 3–19.

Baroni, M., Dalmonte, R., & Jacoboni, C. (1992). Theory and analysis of European melody. In A. Marsden & A. Pople (Eds.), *Computer Representations and Models in Music* (pp. 187–206). London: Academic Press.

Berg, P. (1987). PILE – a language for sound synthesis. In C. Roads & J. Strawn (Eds.), *Foundations of Computer Music* (pp. 64–108). Cambridge, MA: The MIT Press.

Berlyne, D. E. (1971). The new experimental aesthetics. In D. E. Berlyne (Ed.), *Studies in the New Experimental Aesthetics: Steps Towards an Objective Psychology of Aesthetic Appreciation* (pp. 1–25). Washington: Hemisphere Publishing Co.

Biles, J. A. (1994). GenJam: a genetic algorithm for generating jazz solos. In *Proceedings of the 1994 International Computer Music Conference*, (pp. 131–137). San Francisco: ICMA.

Bundy, A. (1990). What kind of field is AI? In D. Partridge & Y. Wilks (Eds.), *The Foundations of Artificial Intelligence* (pp. 215–222). Cambridge, UK: Cambridge University Press.

Camilleri, L. (1992). Computational theories of music. In A. Marsden & A. Pople (Eds.), *Computer Representations and Models in Music* (pp. 171–185). London: Academic Press.

Chalmers, D. J. (1994). On implementing a computation. *Minds and Machines*, *4*, 391–402.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. Cambridge, MA: MIT Press.

Colley, A., Banton, L., Down, J., & Pither, A. (1992). An expert–novice comparison in musical composition. *Psychology of Music*, *20*, 124–137.

Cope, D. (1991). *Computers and Musical Style*. Oxford: Oxford University Press.

Davidson, L. & Welsh, P. (1988). From collections to structure: the developmental path of tonal thinking. In J. A. Sloboda (Ed.), *Generative Processes in Music: the Psychology of Performance, Improvisation and Composition* (pp. 260–285). Oxford: Clarendon Press.

Desain, P., Honing, H., van Thienen, H., & Windsor, L. (1998). Computational modelling of music cognition: problem or solution. *Music Perception*, *16*(1), 151–166.

Ebcioğlu, K. (1987). Report on the CHORAL project: An expert system for harmonizing four–part chorales. Technical Report RC 12628, IBM, Thomas J. Watson Research Center, Yorktown Heights, NY 10598.

Ebcioğlu, K. (1988). An expert system for harmonising four–part chorales. *Computer Music Journal*, *12*(3), 43–51.

Ebcioğlu, K. (1992). An expert system for harmonising chorales in the style of J. S. Bach. In M. Balaban, K. Ebcioğlu, & O. Laske (Eds.), *Understanding Music with AI: Perspectives on Music Cognition* (pp. 294–333). Cambridge, MA: MIT Press.

Folkestad, G., Hargreaves, D. J., & Lindström, B. (1990). A typology of compositional styles in computer–based music making. In Gabrielsson, A. (Ed.), *Proceedings of the Third Triennial ESCOM Conference*, (pp. 147–152). Uppsala, Sweden: Uppsala University Press.

Ghahramani, Z. & Jordan, M. (1997). Factorial hidden markov models. *Machine Learning*, *29*, 245–275.

Gjerdingen, R. (1999). An experimental music theory? In N. Cook & M. Everist (Eds.), *Rethinking Music* (pp. 161–170). Oxford: Oxford University Press.

Hild, H., Feulner, J., & Menzel, D. (1992). HARMONET: a neural net for harmonising chorales in the style of J.S. Bach. In R. P. Lippmann, J. E. Moody, & D. S. Touretzky (Eds.), *Advances in Neural Information Processing 4* (pp. 267–274). San Francisco: Morgan Kaufmann.

Hiller, L. & Isaacson, L. (1959). *Experimental Music*. New York: McGraw–Hill.

Hodgson, P. W. (1999). Modelling cognition in musical improvisation through evolution. In *Proceedings of the AISB'99 Symposium on Musical Creativity*, (pp. 15–19). Brighton, UK: SSAISB.

Hörnel, D. & Olbrich, F. (1999). Comparative style analysis with neural networks. In *Proceedings of the 1999 International Computer Music Conference*, (pp. 433–436). San Francisco: ICMA.

Johanson, B. & Poli, R. (1998). GP–Music: an interactive genetic programming system for music generation with automated fitness raters. In *Proceedings of the Third International Conference on Genetic Programming*, (pp. 181–186). San Mateo, CA: Morgan Kaufmann.

Johnson-Laird, P. N. (1983). *Mental Models*. Cambridge, MA: Harvard University Press.

Johnson-Laird, P. N. (1991). Jazz improvisation: a theory at the computational level. In P. Howell, R. West, & I. Cross (Eds.), *Representing Musical Structure* (pp. 291–325). London: Academic Press.

Kohonen, T., Laine, P., Tiits, K., & Torkkola, K. (1991). A nonheuristic composing method. In P. Todd & D. G. Loy (Eds.), *Music and Connectionism* (pp. 229–242). Cambridge, MA: MIT Press.

Lakatos, I. (1970). Falsification and the methodology of scientific research programmes. In I. Lakatos & A. Musgrave (Eds.), *Criticism and the Growth of Knowledge* (pp. 91–196). Cambridge, UK: Cambridge University Press.

Laske, O. (1988). Introduction to cognitive musicology. *Computer Music Journal*, *12*(1), 43–57.

Laske, O. (1992). The OBSERVER tradition of knowledge acquisition. In M. Balaban, K. Ebcioğlu, & O. Laske (Eds.), *Understanding Music with AI: Perspectives on Music Cognition* (pp. 258–289). Cambridge, MA: MIT Press.

Leman, M. (1992). Artificial neural networks in music research. In A. Marsden & A. Pople (Eds.), *Computer Representations and Models in Music* (pp. 265–301). London: Academic Press.

Longuet-Higgins, H. C. (1981). Artificial intelligence – a new theoretical psychology? *Cognition*, *10*, 197–200.

Marr, D. (1982). *Vision*. San Francisco: W. H. Freeman.

McClamrock, R. (1991). Marr's three levels: a re–evaluation. *Minds and Machines*, *1*, 185–196.

Meredith, D. (1996). *The Logical Structure of an Algorithmic Theory of Tonal Music*. PhD thesis, Faculty of Music, University of Oxford. Submitted. Available online at `http://www.titanmusic.com/papers/public/thesis1996.pdf`.

Mozer, M. C. (1994). Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi–scale processing. *Connection Science*, *6*(2–3), 247–280.

Newell, A. & Simon, H. A. (1976). Computer science as empirical enquiry: symbols and search. *Communications of the ACM*, *19*(3), 113–126.

Pearce, M. T. & Wiggins, G. A. (2001). Towards a framework for the evaluation of machine compositions. In *Proceedings of the AISB'01 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, (pp. 22–32). Brighton, UK: SSAISB. Available online at `http://www.soi.city.ac.uk/~ek735/papers/aisb01.pdf`.

Pfleeger, S. L. (1991). *Software Engineering: The Production of Quality Software* (2nd ed.). New York: Macmillan Publishing Company.

Polfreman, R. & Loomes, M. (2001). A TKS framework for understanding music composition processes and its application in interactive system design. In *Proceedings of the AISB'01 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, (pp. 75–83). Brighton, UK: SSAISB.

Popper, K. (1959). *The Logic of Scientific Discovery*. London: Hutchinson and Co.

Pylyshyn, Z. (1989). Computing in cognitive science. In M. I. Posner (Ed.), *Foundations of Cognitive Science* (pp. 51–91). Cambridge, MA: MIT Press.

Ramalho, G. & Pachet, F. (1995). What is needed to bridge the gap between real book and real jazz performance? In *Proceedings of the Fourth International Conference on Music Perception and Cognition*, (pp. 349–350). Liège, Belgium: ESCOM.

Reis, B. Y. (1999). Simulating music learning: on–line, perceptually guided pattern induction of context models for multiple–horizon prediction of melodies. In *Proceedings of the AISB'99 Symposium on Musical Creativity*, (pp. 58–63). Brighton, UK: SSAISB.

Roads, C. (1985). Research in music and artificial intelligence. *ACM Computing Surveys*, *17*(2), 163–190.

Simon, H. A. & Kaplan, C. A. (1989). Foundations of cognitive science. In M. I. Posner (Ed.), *Foundations of Cognitive Science* (pp. 1–47). Cambridge, MA: MIT Press.

Sloboda, J. (1985). *The Musical Mind: the Cognitive Psychology of Music*. Oxford: Oxford Science Press.

Steedman, M. (1984). A generative grammar for jazz chord sequences. *Music Perception*, *2*(1), 52–77.

Sundberg, J. & Lindblom, B. (1976). Generative theories in language and music descriptions. *Cognition*, *4*, 99–122.

Sundberg, J. & Lindblom, B. (1991). Generative theories for describing musical structure. In P. Howell, R. West, & I. Cross (Eds.), *Representing Musical Structure* (pp. 245–272). London: Academic Press.

Widmer, G. (2000). On the potential of machine learning for music research. In E. R. Miranda (Ed.), *Readings in Music and Artificial Intelligence* (pp. 69–84). Amsterdam: Harwood Academic Publishers.

Wiggins, G. A. & Smaill, A. (2000). What can artificial intelligence bring to the musician? In E. R. Miranda (Ed.), *Readings in Music and Artificial Intelligence* (pp. 29–46). Amsterdam: Harwood Academic Publishers. Available online at `http://www.soi.city.ac.uk/~geraint/papers/miranda_collection.pdf`.

Xenakis, I. (1971). *Formalised Music*. Bloomington, Indiana: Indiana University Press.