An Artificial Intelligence Approach to Tonal Music Theory

James R. Meehan

ABSTRACT. There are virtually no computer models of tonal music theory, largely due to a preoccupation with syntax that comes from traditional linguistics, particularly from transformational grammar. Narmour's recent book, Beyond Schenkerism [6], defines Schenker analysis as a transformational system, refutes the central parts of that theory, and suggests an alternative theory of analysis. This paper draws a parallel between some of Narmour's ideas and current work in natural language processing, Schank and Abelson's Knowledge Structures [11]. The principal correspondence is between Narmour's "style forms" and the AI notion of semantic primitives. It may now be possible for music theorists to share the philosophy and methodology of AI researchers in producing programs to compose and analyze tonal music.

Technical Report #124

August 1978

Department of Information and Computer Science
University of California, Irvine
Irvine, California 92717

An Artificial Intelligence approach to tonal music theory

James R. Meehan

Although computers are commonly used in the composition and performance of contemporary music, there is very little computer research on models of tonal music theory, for either analysis or composition of music. This seems surprising since tonal theory is in the standard music curriculum and shares many aspects of natural language, a favorite target from the earliest days of computer science. There are programs that use little or no standard theory to compose nursery tunes [7], rounds [9], and even cowboy songs [8]. The early literature shows a strong influence of information theory and theories of composition based on Markov chains, weighted probabilities of state transitions, generate-and-test models, and so on. Few authors of such systems made claims of generality or extensibility, and indeed, no such system has caught on.

There are _very_ few analysis programs, good, bad, or otherwise, the principal exception being a 1968 paper by Terry Winograd [16]. Since there are strong parallels between music and natural language, and since natural language processing is such an active research field, what's missing?

In a review of books on computers and music [15], Barry Vercoe of MIT said:

> we seem to be without a sufficiently well-defined "theory" of music that could provide that logically consistent set of relationships between the elements which is necessary in order to program, and thus specify, a meaningful substitute for our own cognitive processes.

On that same point, Andy Moorer [5] wrote in 1972:

> ... any attempts to simulate the compositional abilities of humans will probably not succeed until in fact the musical models and plans that humans use are described and modeled.

The currently dominant theory of tonal music is that of Heinrich Schenker (1868-1935) who defined a transformational system for music analysis long before Noam Chomsky did the same for linguistics [1]. The transformations reduce groups of notes on one level to single notes on the next higher level in a fashion not unlike parsing context-free grammars. The higher-level notes are said to be "prolonged" by the lower-level notes; a C at one level, for instance, might be prolonged by stepwise motion (C-D-E-F-G). These reduction rules theoretically apply at all levels. Finally, one is left with a two-voice structure known as the <u>Ursatz</u>, for which there are three possibilities: the melodic part (<u>Urlinie</u>) may descend a third (scale degrees 3-1), a fifth (5-1), or an octave (8-1). The <u>Ursatz</u> itself is a prolongation of the triad. Melodic (horizontal) motion is thus viewed as a temporal expansion of harmonic (vertical) structure. Of the many structural levels, three are distinguished: the <u>foreground</u>, which is the surface representation; the middleground; and the background, which is the Ursatz.

The Schenker theory has greatly enhanced our understanding of musical structure by relating the harmonic and melodic aspects of music. But as widely as the theory is accepted and taught, it remains incomplete, imprecise, and a constant subject of debate by music theorists. Two major works have appeared recently that discuss the Schenker theory from a "modern" point of view. The first is a paper by Fred Lerdahl and Ray Jackendoff [3], who have attempted to formalize music theory, improving on Schenker, from within the paradigm of generative Transformational Grammar. The second is a book by Eugene Narmour, Beyond Schenkerism [6], in which the author discusses numerous weak points in the Schenker theory in particular, and in Transformational Grammar in general. He also proposes a new way to look at music. Neither of the two new approaches is yet complete; both sets of authors promise upcoming books in which all the details will be worked out. Yet the two "revisions" of Schenker are utterly antithetical, a tribute, if nothing else, to Schenker's influence.

Narmour's theory, called the implication-realization model, is of special interest to Artificial Intelligence (AI) researchers because it bears a strong resemblance to recent models of natural language processing, particularly the Knowledge Structures defined by Schank and Abelson [11].

Parallels drawn between music and language often center around the issue of syntax, and linguistics seems to be a natural choice for a discipline in which both music and language might be studied. Computer scientists have taken the notion of syntax, formalized it, and have turned the problem of syntax-driven translation of programming languages into a technological skill taught to undergraduates. The problems of natural (human) language, however, are not solvable by syntactic methods, and the split between the "computational linguists" and the traditional linguists widened. In computer science, the area called Natural Language Processing shares very little with the philosophy and methodology of Linguistics. The key to processing language is to look beyond syntax and to represent meaning via semantic primitives, relying on the notion that, at the literal level at least, you can get most people to agree on the meaning of simple sentences such as "John gave Mary a book" and "John gave Mary a kiss"; while the syntax of those two sentences is identical, the meanings are not even close. Roger Schank developed a set of "primitive acts" to describe everyday, physical actions. The theory, Conceptual Dependency [10,11], has been used in various computer programs that understand newspaper stories, make inferences, translate, paraphrase, summarize, answer questions, and write stories. Neither the text-analysis programs nor the text-generation programs rely on grammars of English syntax.

In the recent work of Schank and Abelson, structures above the level of individual actions are organized, particularly those that deal with aspects of human problem-solving. Routine sequences of actions, such as what one normally does in a restaurant, are described as scripts. Above scripts (in a problem-solving hierarchy) are plans, which involve more choices and decisions. In theory, some scripts "evolve" from plans by learning from repeated experience; while most people understand the principle of shelving books in a library, for example, and could figure out what to do if confronted with such a task, their initial behavior would likely differ from that of skilled library workers. Plans are driven by goals, which can be permanent (e.g., staying healthy), temporary (mowing the lawn), cyclic (eating food), and so on. Finally, themes account for behavior associated with roles (fireman), interpersonal relationships (spouse), and lifestyle (jet set). Understanding language, in this view, requires an understanding of people; it has much to do with cognitive psychology and very little to do with grammar.

The programs that parse English into these Knowledge Structures are based on expectations of meaning, both at the lower levels (after reading "John gave", expect a person or an object to be mentioned) and at the higher levels (after finding out that the waitress brought John a burnt hamburger, expect him to indicate his displeasure). The programs that produce English text, such as the program that makes up stories [4], use these Knowledge Structures to simulate rational problem-solving behavior.

In transformational grammar, one seeks a correspondence between deep structure and surface structure, and the issue of grammaticality is paramount. In Schenker analysis, one seeks a correspondence between the background and the foreground, already knowing what transformations are possible. If we're interested in the phenomenon of understanding, then from the fact that people often make perfect sense of sentences that are ungrammatical, we can conclude that grammar isn't very important. Likewise, not even musicians are troubled by having failed to detect the <u>Ursatz</u> when they hear a Beethoven symphony.

Is it possible, then, to define an expectation-based procedure for analyzing music? Narmour seems to think so. There is a difference in terminology between Narmour and the AI researchers. To Narmour, the term "expectation" means "prediction with absolute certainty," which is not the AI sense. As I understand it, his implications are <u>possible</u> consequences, and realizations are <u>actual</u> consequences. When Narmour says that X implies Y, I take it to mean that if we see X in a piece, then we should not be surprised to see Y later on. In this sense, the ascending line B-C-D-E, in the key of G, "implies" an F# (among other things). Implications are not only melodic, as this example was, but are associated with each of the "parameters" of music, harmony, rhythm, meter, etc.

Much of Narmour's argument seems to focus on properties inherited from Schenker anaylsis, such as whether a structure is "closed" or "unclosed." In AI terms, this is analogous to asking whether inferences are limited to exactly one domain of information; in language analysis, the answer is certainly no. The ramifications of "John gave Mary a kiss" are not narrowly defined, and the position that musical inferences are similarly broad would not seem to require such a lengthy defense.

I disagree with some of Narmour's musical analyses, and a more detailed discussion of those arguments is in preparation. Narmour presents Schenker's own analysis of Bach's Sarabande from Suite No. 3 for Unaccompanied Cello, for example, and points out some suspicious readings that are "correctly" explained in his own system. It is quite possible, however, to describe the piece "correctly" within the Schenker system, too. Schenker was not an infallible interpreter of his own theory.

A major point of disagreement that I have with both the Narmour system and the Lerdahl-Jackendoff system is something on which they surprisingly agree. They each describe analysis as a process that uses rules recursively, implying that music analysis starts from notes and applies rules to transform these into other notes, which are further transformed by the same rules, on and on. The idea has a certain mathematical appeal to it. However, Schenker as written does not necessarily match Schenker as taught. Moreover, the experience with natural language tells us that using the same representation on many levels is unnecessary. Conceptual Dependency expressions, for instance, are not English sentences in some "reduced" form of the language. They represent meaning by means of an *interlingua*, corresponding to no particular human language. Inferences are not keyed by specific English sentences but rather by the representation of meaning. Otherwise, all synonymous sentences would have to be listed explicitly, which is not simply inefficient but also psychologically dubious. It also seems inconsistent of Narmour to support recursion, since that leads directly to the phenomenon of the one, single "explanation" (*Ursatz* in Music, S in linguistics), that analytic superstructure against which he argued so vehemently. Indeed, it is the lower levels of analysis that are most convincing in Schenker analysis, the explanation of the *local* connections between melody and harmony. By the same token, some of the Lerdahl-Jackendoff theory seems usable, particularly their preference rules. It is much more useful to know the principles of metrical grouping, for example, than to define a transformation to enforce conformity with the data structure of trees, which seems an exercise in notational convenience, at best.

What are the elements of the musical <u>interlingua</u>, the musical semantic primitives? Narmour calls these <u>style forms</u>,

> those parametric entities in the piece which achieve enough closure [local explanation] to enable us to understand their intrinsic functional coherence without reference to the functionally specific, intraopus context from which they come. [6,p.164]

In our terminology, style forms are the semantic primitives of the "problem domains" (melody, harmony, rhythm). They are patterns that "make sense" by themselves. The goal of analysis, then, is to discover "what's going on" melodically, harmonically, etc., to see how all the domains are being integrated.

The number of domains is not limited to three. Some others are meter, articulation, timbre, and tempo. When we say that a certain piece is more interesting harmonically than rhythmically, for example, what we mean is that the rhythmic patterns are simple, whereas the harmonic patterns are not. Bach four-part chorales, for example, exhibit less variety in rhythm than in harmony.

In this sense, even the simplest music is more complex than the natural language that can currently be processed by machine. Computers are good at understanding what one might describe as an environment rich in problem-solving behavior. Scripts can be seen as solutions to problems, worked out in advance. Plans are used when there is no ready-made solution. For representing literal meaning, that's fine.

But those programs work on ordinary prose such as newspaper stories. What would they do with literature, either prose or poetry? They cannot recognize literature because there's no representation of those domains that define literature as something beyond simple prose. That's not a criticism of the Schank-Abelson work; they never set out to process art. For that matter, nor has anyone else. Of course, just as there are music programs that "compose" nursery tunes, there are language programs that produce "poetry," too, but they're on equally weak foundations. It's hard enough modeling the "literal" domains, and logically, they must precede the modeling of other domains. The richness of the real world, even in simple texts, makes that problem very hard indeed.

Simple tonal music, in contrast, would seem to have more domains but less complexity within each domain. Musical scripts abound (e.g., cadences). The few, common meters provide "solutions" that make rhythmic expectations very simple. (Of course, just as in language, one aspect of what it means to be interesting is to avoid the easy solutions, which may explain why so much of the popular music of the 1950's, with endless repetitions of I-vi-IV-V7, is mind-numbing.) Integration of domains is a more obvious problem in music than in language, but I believe that the similarities exist. Narmour, among others, points out the differences between music and <u>the linguists' view</u> of language, and I certainly agree with him. It is not at such a low level as syntax that the two are similar. Even less apt is Smoliar's comparison between computer programs and music [12], or his more recent work [13] in which each Schenker transformation is represented by a procedure that manipulates a global data structure. That system is to music what a text editor is to writing.

The final part of the analogy to be discussed is the synchronic/diachronic nature of the theory. In any system, it is tempting to seek a set of factors that, with one set of values, describes one historical point or style, with another set of values, a different point, and so on. You can then invent a theory about the nature of the changes from one set to the next, and you claim you have a diachronic model, an epistemological philosopher's stone. Narmour hopes to do the same with compositional styles: what distinguishes Beethoven's Fifth from the Sixth? In natural language processing, this is described as the problem of learning. It's hard enough, right now, to model how people <u>use</u> knowledge structures, for instance; modeling how they <u>acquire</u> them is a higher-level problem, requiring a synthesis of all the experience gained from building many individual models. In other words, it's too soon to expect any significant answers about The Big Problem, the "universals" of music. One seriously doubts Lerdahl and Jackendoff's claim:

> Preliminary investigation has indicated that the theory can be modified to produce structural descriptions of pieces in styles as diverse as Macedonian folk music, North Indian music, and

> 14-century French music, by changing various specifics of rhythmic and pitch structure. [3,p.166]

Where does one start, then? With a music theory textbook, perhaps, but reading it with the task in mind of representing the information there in a computer program. To what end? A theory of music should certainly explain aspects of the undeniably tonal music with which we are daily bombarded, such as the Bee Gees or Barry Manilow. Even the 5-note theme from _Close Encounters of the Third Kind_ is (alarmingly) tonal. But my own preference is to concentrate on harmony and melody -- it is difficult to model even the simplest tonal music without them -- and, to a lesser extent, rhythm, with an initial goal of writing chorales. This is the type of experiment in choosing semantic primitives, forms of representation, and control structure for which there is ample precedent in AI. If our analogy with research in natural language processing is valid, a knowledge-based system will provide better results than previous attempts.

REFERENCES

1] Noam Chomsky.
Syntactic Structures.
Mouton, The Hague, 1957.

2] Otto E. Laske.
Understanding the behavior of users of interactive computer music systems.
Revised version. School of Music, University of Illinois, Urbana, Illinois, 1977.

3] Fred Lerdahl and Ray Jackendoff.
Toward a formal theory of music.
Journal of Music Theory 21(1):111-172, Spring 1977.

4] James R. Meehan.
The Metanovel: Writing Stories by Computer.
PhD dissertation, Yale University, 1976.
Research Report 74, Yale Computer Science Department.

5] James Anderson Moorer.
Music and computer composition.
Communications of the ACM 15(2):104, February 1972.

6] Eugene Narmour.
Beyond Schenkerism.
University of Chicago Press, 1977.

7] R. C. Pinkerton.
Information theory and melody.
Scientific American 194(2):77, 1956.

8] H. Quastler.
Discussion, following Mathematical theory of word formation, by W. Fucks.
In E. C. Cherry, editor, Information Theory -- Third London Symposium, Academic Press, New York, 1955, p. 168. Cited in Joel E. Cohen, Information theory and music, Behavioral Science 7, 1962, p. 163.

9] Gary M. Rader.
A method for composing simple traditional music by computer.
Communications of the ACM 17(11):631-642, November 1974.

10] Roger C. Schank,
Conceptual Information Processing.
American Elsevier, New York, 1975.
Includes contributions by Neil M. Goldman, Charles J. Rieger, and Christopher K. Riesbeck.

11] Roger C. Schank and Robert P. Abelson.
Scripts, Plans, Goals, and Understanding.
Lawrence Erlbaum, Hillsdale, New Jersey, 1977.

12] Stephen W. Smoliar.
A Parallel Processing Model of Musical Structures.
PhD dissertation, MIT, 1971.

13] Stephen W. Smoliar.
Schenkerian analysis without angst.
Music Project Report 8, Department of Computer and Information Science, University of Pennsylvania, 1976.

14] Barry Truax.
A communicational approach to computer sound programs.
Journal of Music Theory 20(2):227-300, Fall 1976.

15] Barry Vercoe.
Review of Harry B. Lincoln, The Computer and Music, and Barry S. Brook, Musicology and the Computer.
Perspectives of New Music 9(1)-10(2).

16] Terry Winograd.
Linguistics and the computer analysis of tonal harmony.
Journal of Music Theory 12(1):2-49, Spring 1968.