

## AUTOMATED COMPOSER OF STYLE SENSITIVE MUSIC

Michael Chan<sup>1</sup>

<sup>1</sup> School of Computer Science and Engineering,  
University of New South Wales

**ABSTRACT:** Although much effort has been devoted to automating music composition through a variety of algorithms and methods, the resulting works seldom imitate musical style. This paper draws on existing ideas to propose the Automated Composer of Style Sensitive Music (ACSSM) system, with the goal of constructing musical work that imitates a given musical style, as well as conveying musical meaning. Various hierarchical structures are used to model music in order to adapt several existing algorithms and to provide a selection of analysis and generation methods based on techniques from Artificial Intelligence. The basis of ACSSM is a structural algorithm that imitates musical style, further extended by concepts derived from musical theory to model the musical intuitions of a listener. ACSSM accepts a corpus of classical pieces from a particular composer, in an XML-based format, and produces new musical works that sound more musical and in the style of the given composer than do most other algorithmic methods.

### INTRODUCTION

Music can be considered as a form or product of creativity and expression of emotion. Composers are inspired emotionally, in ways that are unique to humans, to create certain combinations of sounds. Even for experienced composers, music composition often is arduous work. In contrast, machines are convenient logical devices that are simply programmed with sets of instructions to perform tasks, such as the task of writing music. Because machines use logic-driven, objective processes, hence they do not possess emotions and feelings that make the music they produce sound meaningful to human listeners. The envisioned system, Automated Composer of Style Sensitive Music (ACSSM), concentrates in the intervening area in between human and machine music composition and makes use of the human emotions found in existing musical work, while also performing logical analysis using AI techniques. This concept allows stylish and musical-sounding work to be produced, with the benefits of using a machine.

Given a corpus of musical scores and indicating the music style of a particular score by the user, represented in XML, ACSSM reads this corpus and performs a collection of analyses on a range of musical attributes. It then constructs a new piece that is reminiscent of the indicated style. The functions of ACSSM are based on the piano music from the Classical, tonal music era, when beauty of form was emphasised strongly. There exist numerous algorithms and theories based on the Classical era, including *Experiments in Musical Intelligence* (EMI) by David Cope [Cop95] and *A Generative Theory of Tonal Music* (GTTM) by Fred Lerdahl and Ray Jackendoff [LJ83], which together form the basis of our system. Both of these theories are based on the insights of Schenkerian music theory and are designed solely for tonal music, which is pursued by ACSSM.

ACSSM adapts the style-imitative techniques developed in EMI, which is an automated algorithm consisting of several phases of analysis and AI techniques. To enhance the quality of the produced work, the musical "meanings" are empowered by making use of the musical structure unconsciously inferred by a music listener, which is introduced in the GTTM. While most types of algorithmic music lack musical sense and style, the methods we adapted should offer a new perspective that results in stylistic music.

### BACKGROUND

#### Experiments in Musical Intelligence (EMI)

These experiments were begun in the early 1980s by David Cope [Cop95] and have involved programs developed in LISP, which has been proven successful in producing music by using techniques of musical recombinancy, pattern matching, and augmented transition networks (ATNs).

EMI analyses existing scores of tonal music and recombines the musical segments from these scores into new compositions in accordance with the found style. The results from EMI strongly imitate the style of the original works, in contrast with scores composed by most other algorithmic composition tools.

Cope's philosophy is based on one of the first formal types of combinatorial music, the eighteenth-century *Musikalisches Würfelspiel*. The idea behind the philosophy is that composing music is a matter of recombining musical phrases. However, EMI is remarkably more sophisticated than the eighteenth-century work. Furthermore, EMI is considered to be corpus-based instead of rule-based, which allows greater flexibility and diversity. The intuition is that if the corpus is large enough, then the patterns identified in it can be of great use for knowledge-intensive tasks. These tasks spread across the four major components in the structural algorithm of EMI: analysis, pattern matching, deconstruction, and reconstruction.

An alternative analytic method to traditional harmonic analysis was developed to aid the computational process. This analysis is called the *SPEAC (Statement, Preparation, Extension, Antecedent and Consequent)* system and is a functional analysis of the harmonic progression of a given corpus of music. The analysis proceeds along the lines of the work of Heinrich Schenker (1935), namely Schenkerian Analysis. SPEAC provides a level of abstraction for describing notes and harmonies; the acronym itself stands for five identifiers:

- Statement (S): Exists "as is" and is not the result of any other activity. Statements typically occur near the beginning of musical passages.
- Preparation (P): Are non-independent identifiers that precede statement or other identifiers, by which meanings are modified
- Extension (E): Behaves similarly to preparation; however, extension may follow any identifier other than another extension or a preparation.
- Antecedent (A): Causes a significant implication and requires resolution, typically a consequent.
- Consequent (C): Typically appears in response to an antecedent motion and is often the same chord as the statement.

SPEAC analysis attempts to use the characteristics of tonal music and determine the function and character of a group of pitches. An extensive SPEAC representation and harmonic vocabulary is used in a subset of the EMI project, SARA (Simple Analytic Recombinancy Algorithm).

The second phase, pattern matching, helps the produced music to better mirror a given style. [Cop91] defines musical style to be:

*the identifiable characteristics of a composer's music which are recognizably similar from one work to another.*

These signatures [Cop91], which are retrieved and pattern matched from a large database of scores, remain intact throughout the entire creation process.

The subsequent phases, deconstruction and reconstruction, are closely correlated. Deconstruction breaks down the input scores into small musical segments (usually measures), while reconstruction recombines these segments to form a new piece. An augmented transition network (ATN) constitutes the core of the reconstruction phase. An ATN is a network consisting of nodes and arcs, where the nodes represent different states of a process and the arcs represent the transition between states. Since the arcs of an ATN also can refer to other networks, hence it has the same descriptive power as context-free grammars. The power of an ATN resides in the augmentation of a set of tests to be satisfied before an arc is traversed and intermediate results or global states can be saved in *register* [Woo70]. These tests are used in EMI to help to define possible transitions over the SPEAC functions.

Augmented Syntax Diagram (ASD)

In ACSSM, the ATNs have been designed and implemented based on ASD. Similar to ATNs, the ASD, which has been developed by James A. Mason [Mas04], is capable of representing any context-

free grammar and can be augmented with registers that store intermediate values and tests that define the conditions of transitions. However, several important distinctions exist between the two representations, including diagram labelling and parsing approach. With ATN, the parsing is performed by a top-down approach, in contrast to the bottom-up approach in ASD. In addition, ATNs contain both node and arc labels, while ASD networks only contain only node labels, which correspond to the arc labels of an ATN.

#### A Generative Theory of Tonal Music (GTTM)

Certain aspects of structural hearing and hierarchal music comprehension in tonal music have been proposed, specifically in Fred Lerdahl and Ray Jackendoff's [LJ83] *A Generative Theory of Tonal Music*. Along the lines of the generative linguistic theory of Chomsky and its psychological concerns, and also in parallel with the perception of Schenkerian Analysis, the GTTM describes the experience and unconscious principles of a music listener with the tonal idiom. Hence, since music is made for the pleasure of listening, there exist possibilities of modeling music style based on this theory as well.

The theory comprises four types of analysis of musical compositions: grouping structure, metrical structure, time-span reduction, and prolongational reduction. Along with these analyses, two types of rules in turn contrive them: well-formedness rules, which describe the fundamental conditions for syntactically correct structures, and preference rules, which identify relatively more musically meaningful structures.

Both grouping and metrical structures are essential to the conception of musical rhythm in GTTM. The former refers to a hierarchical segmentation of the musical continuum into motifs, themes, phrases, and sections, whereas the latter expresses the intuition that the events of the piece are related to a regular alternation of strong and weak beats at hierarchical levels. Figure 1 shows an example of the grouping structure of the melody in the Mozart G Minor Symphony, K.550 [LJ83].



Figure 1. The melody of Mozart's G Minor Symphony, K.550 and its grouping structure. The levels away from the note level are relatively more pronounced.

The music listener's attempt to organize all the pitch-events of a piece is modeled by the reductions into a single coherent structure, such that they are perceived to be heard in a hierarchy of relative importance. Time-span reduction concerns the relative stability within rhythmic units, whereas prolongational reduction analyses the expression of relative stability in terms of continuity, progression, and the movement toward tension and relaxation.

#### SYSTEM OVERVIEW

A high-level block diagram of the system components is shown in Figure 2. The user is required to supply the system with a database of musical scores in the format of MusicXML [Rec04]. Beneath the front end lie the *Harmonic Analyser* and *GTTM Engine*, which both examine the musical attributes – even in completely different approaches – and prepare for the *Deconstructor* to break down the music. Finally, the *Reconstructor* utilises the results and produces the new musical work. Although the structural algorithm of EMI constitutes that of our system, modifications and alternatives have been introduced to allow improvements.

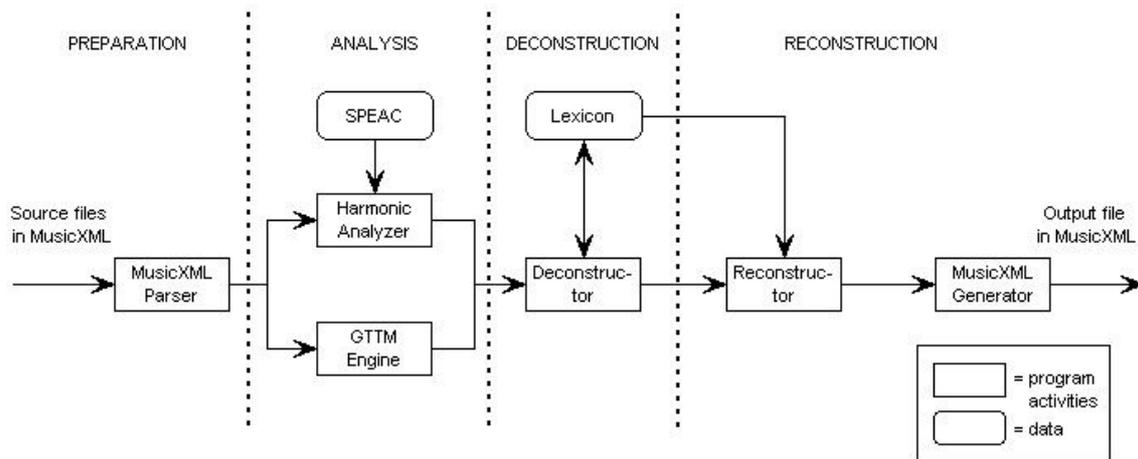


Figure 2. Overview of System

### Methodology

MusicXML, which is an XML-based music interchange language that serves the purpose of notating Western music, has been used to represent the input and output musical score files in a consistent and computable format. MusicXML has the ability to preserve ample articulations and ornaments, and it is relatively convenient because of its compatibility with a range of professional music notation software including Finale and Sibelius. Both a MusicXML translator and a MusicXML generator are required to interchange between the XML-based and internal representations, and vice versa.

The analysis phase involves the use of the SPEAC system in *Harmonic Analysis* and a superficial analysis using the rules provided by the GTTM. Using the original SPEAC “vocabulary” developed in SARA (a subset of EMI), ACSSM conducts harmonic analysis that captures the harmony of the chords from the given scores. These analyses for harmony meanings will then contribute to the degree of style imitation concerned in a later phase. GTTM by itself is often regarded as a plausible theory and method of modeling of musical hearing; ACSSM attempts to integrate with it to improve the musical meanings in the produced music. In the context of the functionalities of ACSSM, the grouping and metrical structures in GTTM are incorporated into the system because these two structures should provide more direct knowledge than do the reductions.

With EMI, a large corpus is needed to produce work that is relatively true to the style and is especially vital in the phase of *pattern matching*. Only a limited number of suitable scores could be obtained during our development of ACSSM. Because this was one of the limitations of this project, the phase *pattern matching* in EMI is removed, and a more attentive reconstruction method has been used. As a result, *signatures* have not been regarded in ACSSM.

Once we have analysed the musical attributes of the original scores, these scores are broken down or *deconstructed*, a process that comprises the method from EMI. This method scans across each of the scores and “cuts up” the score into small segments. Each of these segments retains information about the first ensuing note found in the original score, a note that Cope [Cop95] calls the *destination*. The destination notes are to be used in reconstruction to provide better connectivity between the new segments. Finally, these deconstructed segments are stored in the corresponding lexicon, together with other segments with the same SPEAC meaning. Despite the fact that the size of the music segments used in EMI is of one measure – or, if smaller, one beat – ACSSM attempts to determine the segment size with the grouping structure detailed in GTTM. With this in mind, the segment size in ACSSM is dynamic rather than being discrete, as in EMI, and each segment should be relatively more befitting the musical sense, as grouping structure suggests the method to determine the boundary of the perceived musical partitions.

A recombinatory algorithm based on linguistic techniques and AI technologies, ATN, constitutes the core component of the reconstruction process, which intends that the produced work maintains the style found in the indicated score. The musical segments extracted from the database in the earlier phase are analysed in terms of their SPEAC meaning. The reconstructed piece is in fact a logical

rearrangement of the segments that exist in the database. For an arrangement of these segments to resemble the source work, the harmonic grouping of the selected segments must match the harmony network of the reconstruction. In fact, in our system, the final output is generated using an ASD, which is conceptually similar to ATN, having the SPEAC derived in the indicated score as its nodes. Furthermore, the rhythmic aspect is to be maintained by using the metrical structure from GTTM. The original metrical structure can then be used to lay the metrical “backbone” of the reconstructed piece.

To connect the segments chosen in the reconstruction, a seeking method that simulates a coupling technique was developed and has been used in EMI. The connection is established when the first notes of any segment match the destination notes of the preceding segment. If there exists a segment in the database that contains the same destination notes but subsequently moves in different directions, and it is selected due to the pseudorandomness in decision making, then a new coupling of segments eventually can be created to produce the new-sounding music.

## DESIGN AND IMPLEMENTATION

Java has been chosen as the software platform for the development of ACSSM, primarily for its platform portability, object-oriented concept, support for XML, and good availability for external modules. One of the advantages of XML is that there already exist many types of software for free use, including parsers. With these considerations in mind, the MusicXML translator designed for our system uses the JDOM API to read and write MusicXML documents. In contrast, ATN packages are less common; moreover, ATNs are designed for more specific and technical usage. Nonetheless, Mason, the inventor of ASD, has been developing a Java API that reasonably suits the purpose of the reconstruction phase. One of the main issues with adopting the ASD is that the parsing and generation approach is required to be reversed to top-down from bottom-up.

### Data Hierarchy

To internally represent physical scores, ACSSM uses a hierarchy of data objects that closely reflects to their structure, as shown in Figure 3. This structure also approximately reflects to MusicXML because it is the source format of data.

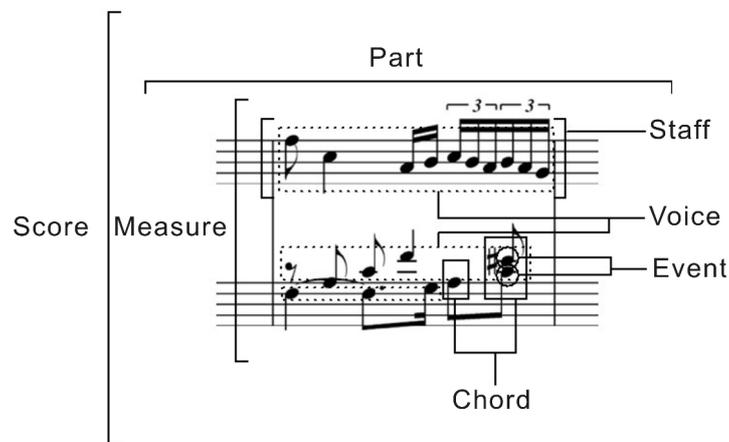


Figure 3. The representations used in ACSSM

On the lowest two levels in the hierarchy lie the Event object and the Chord object. The Event object, which is the most atomic block, contains the information about a note, mostly extracted from the `<note>` elements in the MusicXML files, including:

- Pitch: The equivalent MIDI note number
- Duration: The length of the note in MusicXML measurement

A musical chord can be represented by a Chord object, which is simply a list of Events that have the same starting time (*on-time*) and duration, while also existing on the same musical staff.

Because MusicXML is reasonably powerful in preserving music attributes that are found in physical scores, it allows the notes of the same on-time to have different time durations as well. This layout of music can be represented by grouping the notes of different durations to different voices. Hence, in

ACSSM, a Voice object contains all the chords from a measure, and these are allocated to the corresponding voice represented in the original XML file.

The next two levels up in the hierarchy are the Staff object, which is simply a collection of voices, and the Measure Object. Like physical musical scores, staves are clef lines; therefore, the same is being mirrored in ACSSM. Likewise, the Measure object represents a measure/bar in physical scores and contains all the staves that belong to it. Because ACSSM is designed primarily for piano music, it is assumed that each measure should contain two staves – that is, a grand staff.

Further up in the hierarchy lies the Part object, which represents the musical part played by each instrument. In actuality, because ACSSM is designed for piano music, there is only one part to be used. Nonetheless, the current structure should provide ample extra space for later expansions, when multi-part music is to be accepted into the database.

Finally, a Score object is at the highest level in the data hierarchy for representation of music. Since a score represents a piece in the database, it contains all the parts played in the piece (only one in the current stage of ACSSM). Besides maintaining a list of parts, the Score object also contains the grouping and metrical structures from the GTTM analysis.

### RESULTS AND ANALYSIS

Observing an example of the generation of music can be useful in evaluating the effectiveness of the system. To this end, an excerpt of music that was produced by ACSSM is shown in . The input corpus used for this run consists of Bach Inventions no. 3 and no. 10, and Invention no. 2 by Ryan Sheeler. The style to be imitated is that of Invention no. 3.



Figure 4. An excerpt of music produced by ACSSM



Figure 5. Bach, Invention (Sinfonia) no. 10 (measures 21-24)



Figure 6. Sheeler, Invention no. 2 (measures 13-15)

The first two beats of the first measure are a transposition of the first two beats of measure 21 of Invention no. 10 (the leftmost measure in Figure 5). The ensuing notes, after the end of the second beat of measure 21, are the destination notes for this segment; they consist of an A sharp. This A sharp can be subsequently transposed to a D sharp in C major, which explains why the third beat of the produced music consists of a D sharp. This segment is of a length of two beats, but not all segments are of the same length. This can be demonstrated by the third beat of measure 1 of the produced music. This segment is longer than the previous and extends into the end of the first beat of measure 2. Similarly, these three beats were drawn from the corpus – measures 13 and 14 of Invention no. 2, shown in Figure 6. We can see that the use of grouping structure techniques allows the segments to size dynamically, while the ATN and the pseudorandomness in decision making present some rather creative and original combinations. In fact, the use of randomness means that the output is not deterministic; however, the resulting work is much more effective and contains less “gibberish” than that produced by several other techniques, such as Markov Chains and fractals [Jer04]. The harmonic network is one of the vital attributes of musical style, and the ACSSM process worked deliberately to produce a composition with a harmonic network similar to that of Invention no. 3. Some critics may argue that the style produced in the example shown is not truly Bach’s, but the overall musical quality nevertheless can be considered to be reasonably high.

## CONCLUSION AND FUTURE DEVELOPMENT

Our integration of the concepts of GTTM into the structural algorithm of EMI resulted in a method capable of imitating music style and producing works with musical meaning. The AI techniques, used successfully, guide composition of the new piece so that its style resembles that of a given corpus of work and its harmonic backbone resembles that of the indicated work to be imitated. Besides having a harmony resembling that of the work, the music produced can be regarded to be rather musical overall.

Undoubtedly we consider the current system to be rather primitive, and there exist several areas for further research and development. The results highlight that the harmony may not be captured properly at times, which may lead to issues such as unreliable style imitation and cadence allocation. To ameliorate the weaknesses of this aspect of ACSSM, it would be desirable to adjust the harmony analysis phase to become more effective. This might be done by enhancing the current method with a more profound harmony vocabulary and a more nuanced harmony checking technique. This type of enhancement would require extensive musical knowledge.

Besides producing music that better resembles the harmony of source music, our system could be enhanced in regard to other aspects of style imitation through further adoption of the GTTM. The reductions in GTTM relate more to note stability and the expression of tension and relaxation, and modeling of these might be attempted as an extension of the current work. We believe that the

concepts suggested in GTTM can model significant style attributes, and these concepts represent potentials to improve the overall effectiveness of style imitation.

#### ACKNOWLEDGEMENTS

I am most grateful to my supervisor, A/Prof. John Potter, whose great effort, expertise and guidance have steered me clear of troubled waters, and Dr Emery Schubert for his friendly and constructive comments on my work, which were invaluable.

#### REFERENCES

- [Cop03] David Cope. Computer Analysis of Musical Allusions, *Computer Music Journal*, 27:1, pp.11-28, 2003.
- [Cad90] Allen Cadwallader. Trends in Schenkerian Research, 1990.
- [Cop91] David Cope. Computers and Musical Style, 1991.
- [Cop92] David Cope. Computer Modeling of Musical Intelligence in EMI, *Computer Music Journal*, Vol. 16, No. 2, 1992.
- [Cop95] David Cope. Experiments in Musical Intelligence, 1995.
- [Cop97] David Cope. The Composer's Underscoring Environment: CUE, *Computer Music Journal*, 21:3, pp. 20-37, 1997.
- [Cop99] David Cope. Facing the Music: Perspectives on Machine-Composed Music, *Leonardo Music Journal*, Vol. 9, pp. 79-87, 1999.
- [CW92] Francis Chin and Stephen Wu. An Efficient Algorithm for Rhythm-finding, *Computer Music Journal*, 16:2, 1992.
- [Jan92] Thomas Janzen. Algorhythms: Real-Time Algorithmic Composition for a Microcomputer, 1992.
- [Jer04] Gustavo Diaz-Jerez. FractMus: Fractal Music Composition Software, 2004. Available at <http://www.geocities.com/SiliconValley/Haven/4386/overview.html>
- [LJ83] Fred Lerdahl and Ray Jackendoff. A Generative Theory of Tonal Music, 1983.
- [Mas04] James A. Mason. Augmented Syntax Diagrams, 2004. Available at <http://www.yorku.ca/jmason/asdindex.htm>
- [Nar92] Eugene Narmour. The Analysis and Cognition of Melodic Complexity, 1992.
- [Rec04] Recordare. MusicXML Definition, 2004. Available at <http://www.recordare.com/xml.html>
- [Sch43] Joseph Schillinger. Encyclopedia of Rhythms: Instrumental Forms of Harmony: A Massive Collection of Rhythm Patterns", 1943.
- [Wid95] Gerhard Widmer. Modeling the Rational Basis of Musical Expression, *Computer Music Journal*, 19:2, pp. 76-96, Summer 1995.
- [Woo70] W. A. Woods, Transition network grammars for natural language analysis, *Communications of the ACM*, v.13 n.10, p.591-606, Oct. 1970.