Generative Structural Representation of Tonal Music

Alan Marsden

Lancaster University, UK

This is a post-print (author's final draft) of an article published in *Journal of New Music Research,* vol.34 no.4 (2005) 409–428. Details of the definitive version are available at www.informaworld.com.

Abstract

The usefulness and desirability of representation schemes which explicitly show musical structure has often been commented upon. A particular aim of music theory and analysis has been to describe and derive musical structure, and this article discusses computational systems based on this work. Six desirable properties of a structural representation are described: that it should be constructive, derivable, meaningful, decomposable, hierarchical, and generative. Previous computational work based on the generative and reductional theories of Schenker and of Lerdahl and Jackendoff is examined in the light of these properties. Proposals are made for a representational framework which promises the desirable properties. The framework shares characteristics with earlier work but does not use pure trees as a representational structure, instead allowing joining of branches in limited circumstances to make directed acyclic graphs. Important issues in developing a representation scheme within this framework are discussed, especially concerning the representation of polyphonic music, of rhythmic patterns, and of up-beats. An example is given of two alternative representations within this framework of the same segment of music used to exemplify earlier work: the opening of the theme of Mozart's piano sonata in A major, K.331.

1. Rationale

Pieces of music are not physical objects, and so have no single 'ground' manifestation. The question of how to represent a piece of music for processing by computer therefore does not have a single answer. (See Marsden, 1996, 2000 and Wiggins *et al.*, 1993 as examples of discussion of the issues.) Here, I aim to explore some possibilities for systems which are explicitly rich in their representation of structure, and in particular those which reflect hierarchical structure in tonal music in a Schenkerian manner.

A representational system which is rich in structure has interest in itself for those who pursue music analysis, but it also has potential for music-processing applications. Simple tasks like segmentation depend on structure in order to be properly achieved. One can imagine wishing to divide a piece of music into two shorter segments. A representation which explicitly represents a musically meaningful structure for the piece would facilitate making the division at a musically appropriate point, and perhaps in a musically appropriate manner (for example, by adding suitable closing material to the end of the first segment). A task such as producing a variation of a piece of music would require an even more sophisticated representation of structure.

One candidate theory to guide such a structure-rich representation is Schenkerian analysis. Ideally we would have an empirical basis for this theory, though it is not immediately obvious what form such a basis would take. There is some direct experimental evidence that listeners perceive Schenker- or Lerdahl-and-Jackendoff-like hierarchical structures in melodies (Oura & Hatano, 1991; Dibben, 1994), and also some indirect evidence that melodies are segmented in perception in a manner which can be related to such hierarchical structure (Deutsch, 1999). At yet a further remove, the ubiquity of Schenkerian theory in writing about music in the last forty years is empirical evidence of a kind. The lack of empirical data on musical structure is largely because musicians perform tasks on the basis of a shared but essentially internal, non-explicit and quasi-intuitive understanding of a piece's structure: the empirical data on structure is hidden within data about musicians' behaviour. (I suspect, in fact, that musicians do not employ a single structural conception, but a set of interrelated, and possibly imprecise, conceptions. However, computer systems would do better to even use one before trying to become really human and use several!) More solid empirical data would require comparison of the efficacy of different structural representations in facilitating specified tasks, and this project can be viewed as step towards gathering such data. As a start, an early version of such a representation system has been shown to be effective in the representation of melodic pattern (Marsden, 2001) and in the generation of melodies with control over similarity (Marsden, 2004).

Schenker regarded his theory as applicable only to 'masterworks' from the period of Bach to Brahms (roughly). It has since been applied to a wider range of music, but it is clearly a theory of tonal music, and so the representation scheme described here is similarly limited in its scope, though I suspect that moderate adaptations would allow it to apply both to some earlier music, to some later music which remains in a similar tonal idiom (such as much film music), and to later popular music.

2. Requirements

At the outset, it is useful to set out the requirements of a representation system which could facilitate structural musical manipulations. At its most fundamental, a representation system must have two things:

- a language of symbols, meaning that any representation must be made up of elements drawn from a predetermined set of possibilities, and the possible configurations of elements must be defined; and
- a correspondence between configurations of symbols and objects in the world to be represented. (Note that this correspondence does not have to be one-to-one: a single object can have many alternative representations, and a single representation can correspond to a set of objects rather than to just one.)

Thus a score is a representation of a set of objects from the universe of music-as-sound because it is made up from a predefined set of elements (note symbols, etc.) whose possible configurations are defined (e.g., notes must sit on a stave or on ledger lines related to a stave), and it has a well-defined correspondence to actual sounds. A monophonic digital audio recording is also a representation because it is made up from a predefined set of elements (binary numbers in a certain range representing samples) in a very simple configuration (a sequence), once again with a well-defined correspondence to actual sound.

The following appear to be desirable features of a structural representation of music.

Constructive: it should be possible to create (an example of) the music represented from the representation.

Derivable: it should be possible to derive a representation from the music represented.

Meaningful**:** differences between two representations should correspond to significant differences between the musical objects represented. (‘Significant’ will have a task-specific meaning.) Furthermore, if the correspondence of representation to music is one-to-many, the set of pieces of music represented should be meaningfully related and meaningfully distinct from pieces outside the set. If the correspondence is many-to-one, the differences between alternative representations of the same piece should relate to alternatives in the manipulation of a piece of music (which will again be task-specific), for example, different ways of segmenting a piece. If the correspondence between representation and music is many-to-many, both of these should be true.

Decomposable: it should be possible to easily divide a representation into segments which correspond to significant segmentation of the music. Inversely, all meaningful segmentation of the music should correspond to simple segmentations of the representation, and, conversely, it should be possible to combine representations to make a single representation when it is also possible to combine the musical segments they represent into a single piece.

Hierarchical: it should be possible to distinguish different levels of detail in the representation, which correspond to levels of significance in the musical structure. For example, a manipulation of the music which deals with phrases should correspond to a manipulation of the representation at a corresponding level.

Generative: it should be possible to infer the interrelation of elements within a segment of a representation from the representation of that segment, but their interrelation with elements outside the segment should use information about the context of that segment also. Thus, for example, we might be able to infer the interval between two notes which are part of the same phrase unit, but we would need to check the context of that phrase unit to determine the interval to a note from another phrase unit, or indeed to determine the absolute pitch. This means that a high-level manipulation such as transposing an entire segment of music should require no change in the representation of the segment itself (since the interrelation of its elements remains unchanged) but rather a change in the representation of the context of that segment, i.e., at a higher level. When the transposed segment is constructed, the representation of that segment should now produce different details because of the changed context arising from the change at the higher level of the representation.

Note that this definition of ‘generative’ is different from the use of the word by Lerdahl and Jackendoff (1983). They stress that their aim is not a system which can produce music, but rather that they follow other principles from the branch of linguistics known as ‘generative’ (p.6). The simple connotation of ‘generative’ when applied to grammar is contained in my term ‘constructive’: that it is possible to construct the sentence from its representation, and that therefore the universe of all valid representations generates the universe of grammatical sentences. I intend to mean something stronger in my use of the word ‘generative’: that it is possible to construct at least something meaningful of parts of the music from the corresponding parts of the representation. The representation generates the music by ‘growing’ from the ‘genes’ of the representation rather than simply arranging or replicating its elements.

3. Previous work

Most computational systems for the representation of music are intended to allow the manipulation of music notation (such as NIFF), musical sound (the many digital audio representations (see Pope & Van Rossum, 1995), or performance gestures (such as MIDI). Although these do not preclude the manipulation of musical structures, they are not designed to facilitate this and do not generally exhibit the desirable characteristics listed above. (An outline of the requirements of such non-structural representations is given in Byrd & Isaacson (2003).)

Two well-developed systems which are intended for music-structural manipulations (perhaps among other uses) are Humdrum (Huron, 2002) and Charm (Wiggins & Smaill, 2000). MusicXML (Good, 2001) might be included here, but, though less tightly bound to the graphical aspects of notation than NIFF, it is primarily directed at representing music as notated, and so belongs more properly with the systems mentioned in the paragraph above. It is constructive, meaningful, decomposable, and hierarchical, but significance is generally defined in terms of notation rather than musical structure at a conceptual level. Humdrum (which is more a set of representation systems than one single system), is constructive, derivable, meaningful and decomposable, at least in some manifestations. However, it is not generally hierarchical, but on the contrary represents music in a rather ‘flat’ manner.

Charm, but contrast, is explicitly hierarchical, and also constructive and decomposable. It is not, however, generally derivable. While it would be possible to derive the set of elements which make up a Charm representation from a piece of music, the real power of the representation is in the grouping of elements into ‘constituents’, and the representation scheme does not, in itself, define these groupings or how they might be derived. (For applications which do not assume a particular musical ‘language’, this is an advantage.) It is similarly neutral with respect to meaningfulness: while the meaning of elements is well defined (though extensible), the meaning of constituents is open to user-definition. Furthermore, it is not generative.

Humdrum and Charm are both, after all, intended more to be frameworks or models for music-representation systems. Both are intended for broad applicability and so do not embody concepts of a specific musical ‘language’ which might give their representations meaning. Taking inspiration from computational linguistics, a number of researchers have explored the use of grammars to define more or less repertoire-specific musical languages. (See Baroni & Callegari 1984; Baroni, Dalmonte & Jacoboni, 1992, 1999; Kippen & Bel, 1992; Steedman 1984, 1995.) None has been adopted directly into a music-processing system; essentially, they have been attempts to make a formal description of the language of music rather than to solve music-processing problems. However, a parsing of a sentence (or melody) can also be a representation of that sentence (or melody), and one which explicitly relates to its functional composition. Indeed, for this reason, the system which underlay Kippen & Bel’s grammar has formed the basis of a tool for musical composition, the Bol Processor (Bel, 1998). Grammars are explicitly constructive and meaningful, and often generative also. Whether a representation derived from a grammar is decomposable varies from case to case and level to level. (A parsing of a sentence is not generally decomposable into smaller sentences, for example, though a phrase might well be decomposable into smaller phrases. The same holds true for many music-grammatical analogies.) Similarly, whether a grammatical representation is derivable or not depends on the grammar (but the science of this is well understood). Often a parsing is possible in principle, but special attention in the design of a grammar is necessary for it to be possible in practice.

The remainder of this article concerns representations which might be regarded as sitting between the language definitions of grammars, and the language-neutral structural representations of Humdrum and Charm. It draws particularly on the music theories of Lerdahl and Jackendoff (1983), and Schenker (1935). Computational representational systems based on these which have been developed by others will be discussed (Hirata & Aoyagi, 2003a in the case of Lerdahl and Jackendoff, and Frankel, Rosenchein & Smoliar 1976, 1978 (see also Smoliar 1980) and Kassler 1967, 1975, 1977, 1988 in the case of Schenker), before giving a framework for a representation system with (potentially) all the desirable characteristics described above which draws on ideas from Schenker, Lerdahl and Jackendoff, and this later computational work.

4. Representation derived from the theory of Lerdahl & Jackendoff

Grammars are generally designed to operate on sequences of symbols, which creates problems when applied to music which can consist of a number of simultaneous voices. (Of the three examples referred to above, for example, those of Baroni et al. and Kippen & Bel applied to monophonic music, while Steedman’s applied to sequences of chord symbols.) For a richer conception of musical structure, we have to turn to examples from music theory and analysis. The most well known, and most closely related to the concept of grammar, is the theory of Lerdahl & Jackendoff (1983). Like the grammatical research referred to above, their aim was not the representation of music but rather the description of its language. It is worth considering, however, how their theory might form the basis of a computational and structural representation of music, and this is most simply achieved through an example. It is the ‘time-span reduction’ which will be the focus of interest. (Representation on the basis of ‘prolongational reduction’ would also be possible, but would be more complicated with respect to time.)

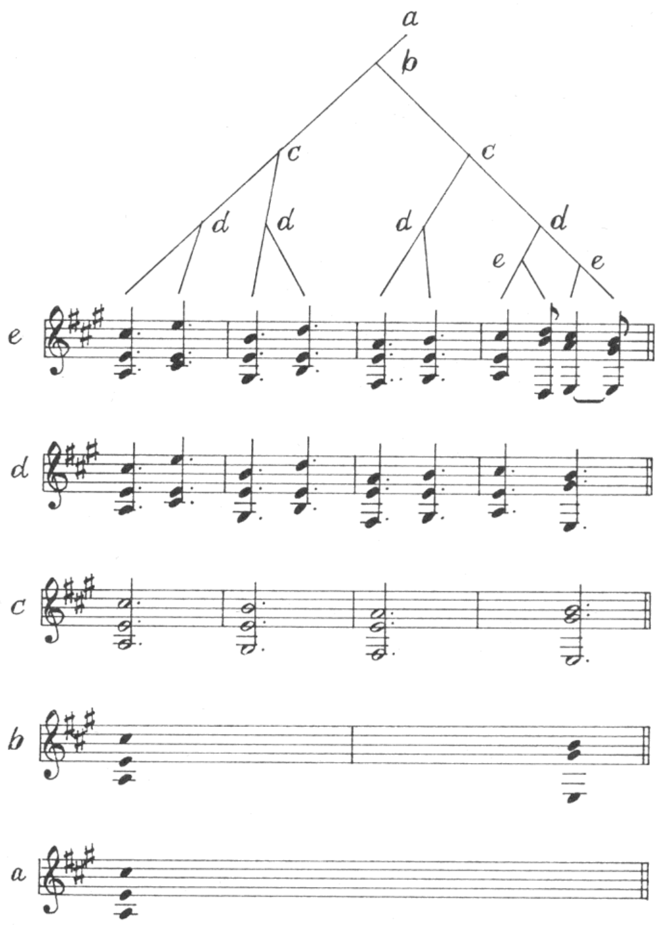

Figure 1. Lerdahl & Jackendoff's analysis of the first phrase of Mozart's piano sonata in A, K.331 (1983, p.227).

Figure 1 shows Lerdahl & Jackendoff's time-span reduction of a 'harmonic sketch' (a reduction) of the first phrase of the theme from Mozart's piano sonata in A major, K.331 (1983, p.227). (The representation of the actual theme rather than just a reduction will be discussed briefly later.) The top half (the tree) and the bottom half (the music notation) are equivalent. (But note that the line of notation on stave 'e' is a part of both representations.) This is translated into a computational form quite easily, using a structure which has two kinds of alternate elements: chords and elaborations. (Because they strictly alternate, it would be possible to conflate them into a single kind of node, but comparisons with later developments are simplified by keeping them separate.) A chord is a triple consisting of:

- a set of notes;
- a time span (a duration); and
- a reference to an elaboration (which can be null).

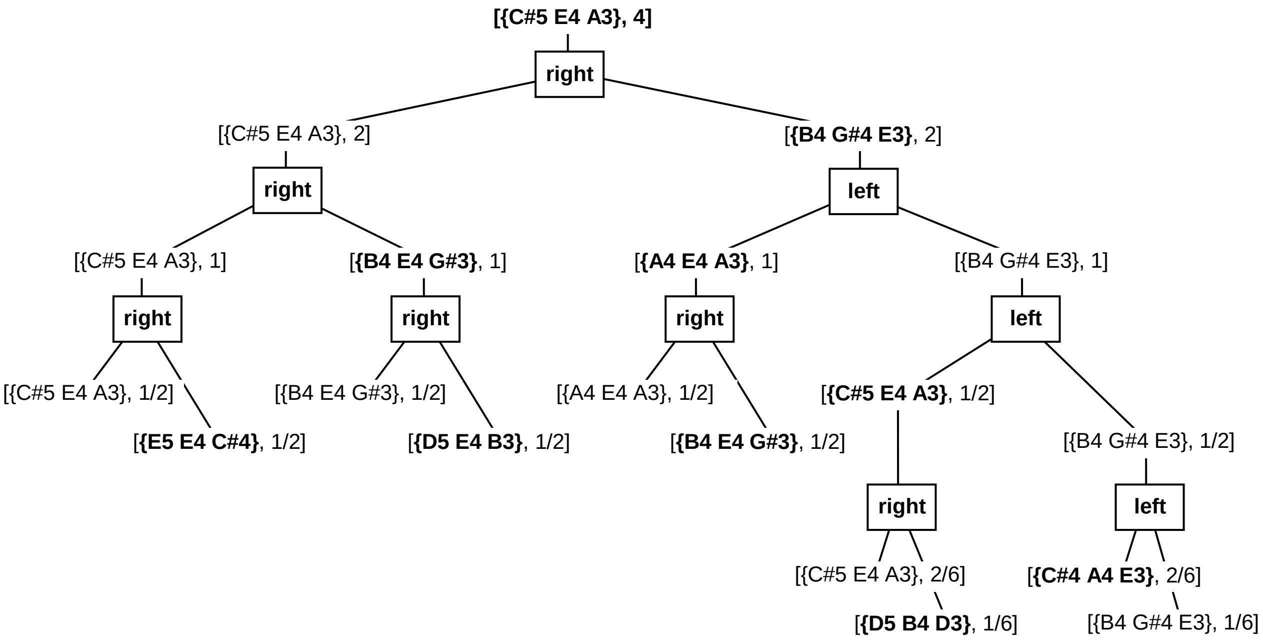

Figure 2. Tree representation derived from Lerdahl & Jackendoff's analysis of the first phrase of Mozart's piano sonata in A, K.331.

There is a small number of kinds of elaborations, which for the present will be confined to right-branching and left-branching. (Elaborations could alternatively be called 'reductions', more closely following the terminology of Lerdahl and Jackendoff; the two names reflect the two 'generative' perspectives: construction of the notes of the musical surface from the representation, and derivation of the structural analysis from the configuration of notes.) An elaboration here is a duple consisting of two chords. Each elaboration has a 'parent', which is the chord which refers to it. The kind of elaboration defines the relation of the parent to the two 'children': in the case of right-branching elaborations, the first child has the same pitches as the parent; in the case of left-branching elaborations, the second child has the same pitches as the parent. The time span of the parent chord is divided into two to yield the time spans of the children.

A diagram of such a representation is given in Figure 2. (Durations are indicated as a number of bars.) This representation is obviously hierarchical, and is clearly meaningful: a change in the pitches or durations corresponds to a change in the configuration of notes represented; a change in the kind of elaborations corresponds to a different conception of the harmonic structure of the piece. It is constructive in the sense that the information is contained which allows a reconstruction of the notes of the score, but those notes have to be directly specified rather than emerging from the representation of the structure. In other words, the representation is not generative (despite its origin in the *Generative Theory of Tonal Music*). It is possible to generate the durations of chords from the representation (assuming the overall duration of the top-level chord is specified, and assuming an underlying 6/8 metre), but the pitches of one of the children of each elaboration must be specified. The details which cannot be generated but must be explicitly specified are indicated in bold in the figure.

The representation is derivable to the extent that the rules of Lerdahl & Jackendoff's theory are computable, a topic which has been the subject of some research but which will not be pursued here. (In summary, the rules do not constitute a complete computable system as they stand, but developments which use them as a basis show some promise, except for significant problems over apparent circularity in the interrelation of different aspects of structure and in concepts such as parallelism.)

The representation is decomposable vertically into half-phrases, bars, etc., but it is not decomposable horizontally into voices; the music is essentially represented as a sequence of chords. This becomes particularly problematic if we wish to take the representation further, beyond the 'harmonic sketch' Lerdahl and Jackendoff present, to the actual theme. This would mean adding to the existing tree, dividing its bottom-level time spans into shorter spans, but at the level of semiquavers the score no longer contains chords but rather notes in the bass and melodic line alone (or in the last bar in the melodic line alone). It would be possible to represent these as chords with tied notes in the middle voice, but this approach would lead, in the case of other pieces, to a proliferation of redundant tied notes. Furthermore, this does not escape a fundamental problem which arises when the structure of time-span divisions is not congruent between different voices (see below).

A full implementation of the representations embodied in the theory of Lerdahl & Jackendoff would require a number of additional developments: different kinds of reduction ('fusion', 'transformation', and 'cadential retention'), a representation of 'retained cadences', and mechanisms to cope with 'augmented time spans' (i.e., those which have an anacrusis (up-beat)). Hirata & Aoyagi (2003a) present a system which satisfies two of these (different kinds of reduction, and augmented time spans; representation of retained cadences is not reported). The differences between their scheme and the outline described above are as follows.

> (1) There is one kind of node in their reduction trees, conflating the note and elaboration nodes (as suggested above) into a quadruple consisting of a head (a note or a chord), a reference to a time point, and two children (both sub-trees).
>
> (2) Information about duration and start time is separated. Duration is attached to individual notes, but start times are defined according to a referential structure of partial order. Thus information about the placement of notes in time is represented in a lattice structure separate from the tree of pitch structure.
>
> (3) In a departure from Lerdahl & Jackendoff's theory, notes retain their duration in reduction instead of taking on the duration of the full time span of which they are head.
>
> (4) Polyphony is accommodated by allowing branching of the tree not only to represent the division of a time span into two shorter time spans but division of a time span into two concurrent spans. Thus there can be concurrent sub-trees covering the same or overlapping periods of time, perhaps dividing it differently. This is similar to the solution Lerdahl proposes for the representation of polyphony (2001; see below).
>
> (5) A well defined concept of subsumption is incorporated, allowing for a degree of indeterminacy in pitch and time, and allowing a definition of the concepts of the common reduction of two trees and the concurrent combination of two trees.

The system has been used as a basis for creation of 'arrangements' by case-based reasoning and as a basis for determining melodic similarity (Hirata & Aoyagi, 2003b).

There are two significant shortcomings of both the simple system outlined above and the more complete system of Hirata and Aoyagi. The first concerns horizontal decomposability. The simple case cannot represent concurrent voices at all. Lerdahl (2001, p.32–34) discusses an ‘enrichment’ of the original concept of grouping structure in Lerdahl & Jackendoff (1983) to accommodate cases where groupings in different voices are not simply aligned but rather overlap. His solution essentially leads to the possibility that some notes on a time span might be reduced with a preceding time span through right-branching while other notes of the same time span might be reduced with a following time span through left-branching. The implied splitting of a single time span into two concurrent sets of notes which are reduced differently is equivalent to a limited form of the concurrent branching allowed in the system of Hirata and Aoyagi. Like Lerdahl, however, Hirata and Aoyagi seem to assume that concurrent time spans will be joined into single chords as soon as possible while allowing a proper representation of grouping structure. Thus a strong preference is shown for decomposition into chords rather than decomposition into voices, and no mechanism is described for extracting a single voice from a tree of chords.

The second, and more serious, shortcoming of these representations based on time-span reduction is that they are not generative. The pitches of lower-level elements are explicitly specified rather than arising from the structure of elaborations. A manipulation such as a transposition would require not just transposing the chord at the head of a tree structure, but transposing all the constituent chords also. It would be possible to define the pitch of children by reference to their intervals from the pitches of parents, but this would only allow simple transpositions. More complex operations such as changing from major to minor could not be accommodated, and some operations exhibit changes of intervals according to harmonic context. (This is common in the adaptations of the subject of a fugue to form a ‘tonal answer’, for example.) A more sophisticated means of describing pitch elaborations is desirable.

5. Previous implementation of Schenkerian theory

Heinrich Schenker’s theory was not presented in a formal manner, and it certainly was not developed with the issue of computational representation in mind. However, it does embody a highly developed conception of musical structure which has a number of points of contact with grammatical theories. It is not surprising, therefore, that two substantial projects in computer implementation of Schenker’s theory took place in the 1970s and 80s. Frankel, Rosenschein and Smoliar (1976, 1978) developed a set of procedures in the programming language LISP which allowed the progressive generation of a data structure which represented, at each stage, the reduction of a piece of music. This was later refined by Smoliar (1980) into a tool intended for use in the creation of an analysis. (Essentially, the role of the tool is to confirm that the analyst’s decisions are valid in the theory and do indeed produce the musical structure envisaged, allowing experimentation in the construction of an analysis.) A piece of music is represented as a tree whose terminal nodes (leaves) are notes and whose non-terminal nodes are structures either of class ‘SEQ’, representing sequences of notes or structures, or of class ‘SIM’, representing notes or structures which begin at the same time. Procedures operate on this structure (or designated parts of this structure) to produce a new structure representing the next level of elaboration. The tree structure for the first phrase of the Mozart theme is shown in Figure 3, at a level of reduction equivalent to Figures 1 and 2, but omitting the middle voice (Frankel, Rosenschein and Smoliar, 1978, p.137).

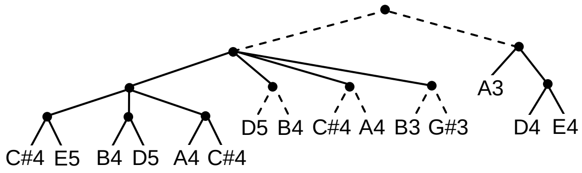

Figure 3. Partial representation of a reduction of the first phrase of Mozart's piano sonata in A, K.331 according to Frankel, Rosenschein & Smoliar (1978, p.137). Solid lines indicate sequences; broken lines indicate simultaneities. The diagram is rotated through 90° from the original in order to match the orientation of other trees, and octave numbers are translated to the ISO convention.

This representation structure is hierarchical, meaningful, and decomposable both horizontally and vertically. It has not been established to be derivable. It is not, however, constructive, because rhythmic information is lost. Schenker's graphs do not show duration (at least not above the foreground level), but they do show simultaneity by the vertical alignment of notes. As pointed out by Rahn (1980), the SIM/SEQ trees of Frankel, Rosenschein and Smoliar fail to show some simultaneities: where two sequences are part of a simultaneity, we know that the two first notes start at the same time, but we have no information about how the remaining notes of the sequences align with each other in time.

The structure is not generative, but the whole system is in the sense that notes are generated by well defined procedures. Thus the nested calls to the 'PROLONG' function, such as in Figure 2 of Frankel, Rosenschein and Smoliar (1976, p.30) do constitute a generative representation of a piece. In itself, though, this is a totally non-decomposable structure (because each call is applied to the whole structure), and it is not perfectly meaningful (because a different ordering of calls can produce the same result without representing any significant difference in structure). Some reference to the procedures which, in the process of elaboration, produce new notes, and to the higher-level notes to which the procedure is applied, could be included at an appropriate point in the tree structure of notes. The procedures apply to sub-trees, which could be individual notes but often are trees of SEQ and/or SIM nodes. Thus, effectively, a third kind of branching is introduced, which corresponds to the elaboration of a sub-tree. An illustration of such a hypothetical generative representation is given in Figure 4 for the first phrase of the Mozart theme, to the same level of reduction as for the examples taken from Lerdahl and Jackendoff. (A partial representation of the second phrase is also given in order to be able to incorporate the highest levels in the representation and show the generation from a single note.) The tree structure is not clear in the figure, but can be understood if the sub-tree generated by each procedure (i.e., the sub-tree attached to the bottom of each box) is considered to be attached to the top of the sub-tree to which the procedure is applied (i.e., the top of the sub-tree within the box), via a node corresponding to the type of procedure. While the inclusion of this additional branching causes the representation to become generative, it does considerably complicate decomposition.

It is worth pointing out two other difficulties with the scheme of Frankel, Rosenschein and Smoliar. Firstly, it includes a REMOVE procedure which means that it is possible to construct an infinite sequence of procedures which produces a tree of only finite size. Thus any mechanism to derive analyses for a piece of music based on this scheme would have to guard against infinite derivations. Secondly, although the scheme includes only a small number of procedures, it would be possible to generate *any* configuration of notes using these procedures, not just those configurations which are representatives of tonal music. While it is not necessary *per se* that a system be incapable of representing non-tonal music, this does raise a question about its meaningfulness.

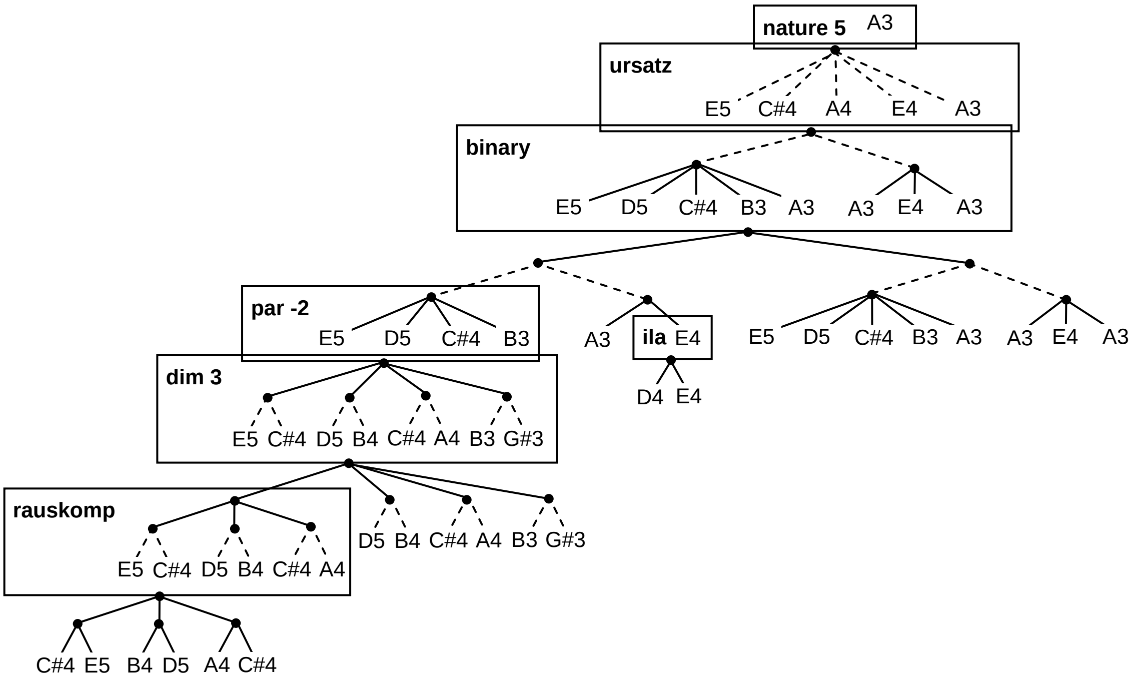

Figure 4. Generative representation derived from Frankel, Rosenschein & Smoliar (1978) by adding the procedures by which notes are generated. Procedures, indicated by boxes, act on the sub-tree which is contained in the box, generating the sub-tree below the box: ‘ila’ means ‘incomplete lower auxiliary’; ‘dim’ means ‘diminution’; and ‘rauskomp’ means ‘reverse *auskomponierung*’.

The other project from this era is that of Michael Kassler (1967, 1975, 1977, 1988). The fundamental idea is once again a set of procedures which elaborate a musical structure. These were originally expressed in the form of a mathematical logic, and later in the programming language APL. It was much more closely based on Schenker’s theoretical exposition as found in *Der friei Satz*, appropriate to the project’s aim of ‘proving’ Schenker’s theory. It only covered the background and middleground levels of the theory, and so it cannot be readily transformed into a full system of representation. As is appropriate for a system dealing with background and middleground only, no account is made of duration, but Kassler’s system does preserve full information about temporal alignment between voices because it is based on an array rather than a tree. Furthermore, it has the advantages over the system of Frankel, Rosenschein and Smoliar that it represents only tonal pieces, and it has been shown to be derivable (Kassler, 1975, 1977, 1988).

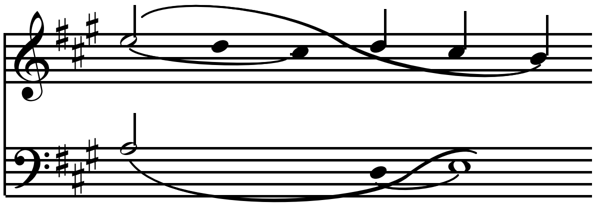

Figure 5. Schenker's analysis of the first phrase of Mozart's piano sonata in A, K.331 (Schenker, 1935, Fig.157).

6. Representation derived from Schenkerian analysis

It is instructive to return to Schenker's own representation system, his analytical graphs, to consider how a computational system can be built on the foundation of his theory, taking into account the lessons learned from the earlier work described above. Figure 5 shows Schenker's analysis of the same first phrase of the Mozart sonata (1935, fig.157). (Some elements (numbers, and the beginning of a broken slur), which relate to the representation of the whole theme rather than just the first phrase, have been omitted.) As with the example from Lerdahl & Jackendoff, there are two significant elements in the representation: notes (instead of the chords found in the time-span reductions of Lerdahl and Jackendoff) and slurs (instead of branches). From this it is easy to derive a similar tree structure. Notes which are grouped by a slur are children of a single elaboration; the difference between note heads (white instead of black; with a stem instead of without) indicate which of the notes is the 'parent' which participates at the next higher level. The two significant differences from Lerdahl & Jackendoff, however, are that voices are treated independently (resulting in this case in two separate but concurrent trees) and (following the direction pointed by Kassler and by Frankel, Rosenschein and Smoliar) the elaborations belong to certain types of 'prolongation', which generate the new notes at each level. A note is here a duple consisting of a pitch and a reference to an elaboration (which can be null). There are several kinds of elaborations, and each is a tuple (whose size depends on the kind of elaboration) consisting of a sequence of notes. This results in the representation shown in Figure 6 (which could be extended without difficulty to cover all the details of the score rather than just the details shown in Schenker's graph, though a third tree would be required for the middle voice). This is generative, to the degree that the pitches of all notes except the two roots can be generated from the given elaborations, assuming an overall tonal context of A major. It is meaningful to the same degree as the representation in Figure 2, but its derivability is less certain, because there exists no rule system for Schenkerian analysis comparable to that of Lerdahl & Jackendoff. (For proposals towards systematisation, see Plum 1988 and Schachter 1999.) Horizontal decomposition is possible because of the separate trees. Vertical decomposition is possible, but complicated by the requirement to synchronise the decomposition of concurrent trees.

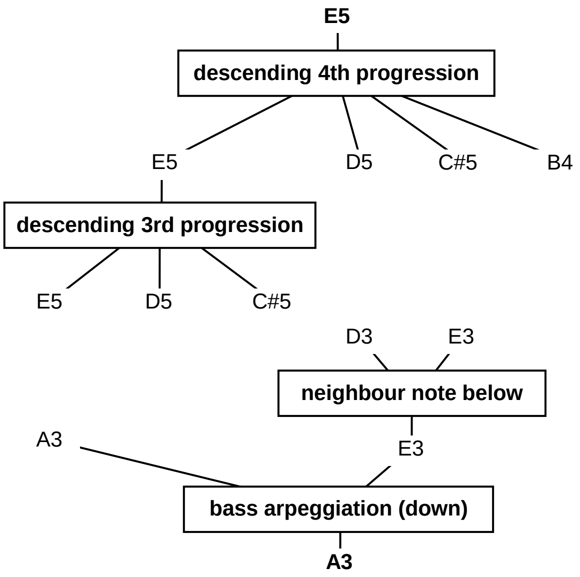

Figure 6. Tree representation derived from Schenker's analysis (Figure 5).

On the other hand, even less rhythmic information is contained than in the case of the system of Frankel, Rosenschein and Smoliar, because all information about temporal alignment is lost. (The vertical alignment which is included as a convenience in the diagram is not reflected in any way in the computational structure.) One possibility would be to move towards the array representation used by Kassler, and make temporal alignment explicit, for example by adding a field to each note which can be used to refer to other notes with which it is aligned. Alternatively, notes could have explicit durations (like the durations for chords in the system derived above from Lerdahl & Jackendoff), which would make alignment implicit in sets of notes which have the same starting time. Furthermore, elaborations could be defined to divide the durations of parent notes in a specific way (according to metrical context), similar to the skeletal system above from the theory of Lerdahl and Jackendoff. This reflects the approach to adding rhythmic information to Schenkerian analysis taken by, among others, Arthur Komar (1971).

7. Issues in generative tree-like representations

To develop these ideas into a fully operational representation system which has the desirable properties described above requires that the following issues be addressed.

7.1 Voices and chords

Any piece of music which is not simply a single sequence of notes poses an issue about how to represent the two 'dimensions' of music which on a score are shown horizontally for sequences and vertically for chords. In some cases it is sufficient to represent a piece as a set of concurrent sequences, so giving precedence to the horizontal dimension. In other cases it is sufficient to give precedence to the vertical dimension and to represent a piece as a sequence of chords. However, neither of these is generally the case throughout a piece and for all purposes. It is often better to allow easy access to information on either dimension or to allow representations which do not consistently give one dimension precedence over the other. Some systems explicitly use two-dimensional arrays, such as that of Kassler referred to above.

Humdrum rotates the two dimensions of a score in text files, so that sequences become columns and (normally) chords become lines. Such array-like representations are difficult to make hierarchical, however, and they do not easily accommodate voices appearing and disappearing (see below).

The two dimensions, however, are not really similar. Temporal relations (at least to the extent of what precedes what) are often fixed; the assignment of notes to voices, on the other hand, is often subject to interpretation. (For example, one might regard the first phrase of the Mozart theme as consisting of three distinct voices throughout. However, it is also possible to hear the voice of repeated Es as simply ceasing on the first beat of the last bar and the notes B4, A4 and G♯4 in that bar to derive from the upper voice earlier in the phrase. This is clearly Schenker's interpretation, as shown in his figure 132.6.) A representation which gives precedence to the horizontal dimension (i.e., representing a piece as a set of concurrent voices) can be divided into segments along the vertical dimension by some automatic procedure which examines temporal order. By contrast, a representation which gives precedence to the vertical dimension (representing a piece as a sequence of chords) cannot easily be divided automatically along the horizontal dimension into separate voices. (See Marsden, 1992 for a discussion of the complexity of this issue.) (Temporal relations are not always fixed: indeterminate temporal relations are possible (see Marsden, 2000), and there can be differences even in perception of temporal order at time scales of fractions of a second (Van Noorden 1975, pp.47-48; Warren *et al.*, 1969). However, both of these occur generally in situations where streaming into voices is strong, and in these situations the horizontal dimension should properly predominate anyway.) There are grounds, therefore, if a tree representation is to be used because of its hierarchical structure, for using concurrent trees whose nodes are notes (as in Figure 6) rather than using a single tree whose nodes are chords (as in Figure 2). Special procedures will be required to ensure the synchronisation of concurrent trees, and to make vertical decompositions of concurrent trees, but these will be less problematic than procedures to make a horizontal decomposition of a tree of chords into several concurrent trees of notes.

Nevertheless, there are kinds of music for which a representation as a set of chords is entirely appropriate (some kinds of guitar music or keyboard music, for example, or figured bass). It would be worth allowing trees of chords as a possibility for those situations, perhaps in parallel with trees of notes to represent situations of tune and accompaniment, for example.

7.2 Splitting and joining of voices

Not only is the assignment of notes to voices subject to interpretation, but voices come and go in the course of a piece of music. A single voice can split into two (as happens in the last bar of the Mozart example in one interpretation), and two voices can join to make a single voice. Furthermore, what appears on the surface as a single voice (or at least a single sequence of notes) can, at a deeper structural level, consist of more than one voice (a phenomenon sometimes called 'pseudo-polyphony' and most obviously manifest in the pieces for solo violin and solo cello by J.S. Bach). Thus, while a representation consisting of concurrent trees of notes might appear best from the foregoing discussion, the trees cannot be truly independent (and therefore perhaps not truly trees!).

Splitting of one voice into two is most easily achieved by allowing a single parent to have more than one elaboration. Thus a single note at a higher structural level can be elaborated in more than one way, resulting in two or more sequences of notes at a lower level. This means that trees can have two kinds of branching: normal branching in the temporal domain to divide time spans into shorter time spans; and special branching in the 'voice' domain to spawn concurrent sub-trees. (This is the solution to the representation of polyphony adopted in Hirata & Aoyagi (2003a).)

The same procedure can produce joining of voices on a single note at a later stage if that note is at the level on which the splitting took place, or higher. However, to accommodate situations where the joining occurs at a lower level, it might be necessary to allow a single note to be simultaneously a child of more than one elaboration. This will produce structures where a sub-tree can be shared by more than one tree.

Pseudo-polyphony, on the other hand, requires a special kind of elaboration which has more than one parent. Another of Schenker's graphs for the Mozart theme (his figure 141) shows that he regards the C♯5–E5 in the first bar as an 'unfolding' whereby a chord consisting of these two pitches (shown explicitly in his figure 132.6) is elaborated to present the two pitches in sequence. A representation of this would have an 'unfolding' elaboration with the sequence of notes C♯5–E5 as children, a parent E5 in one tree (the upper tree in Figure 6), and a parent C♯5 in another tree (not shown in Figure 6). (For a representation embodying this interpretation, see Figure 10 below.) As above, this results in sharing of sub-trees, but this time the top of the shared sub-tree is an elaboration rather than a note, and the details of the sub-tree cannot be generated from the parent in one tree alone but depend on the parents in both trees.

7.3 Tonal and harmonic contexts

It was stated above that the notes in Figure 6 could be derived from the root note for each tree (E5 in the upper case and A3 in the lower) plus the pattern of elaborations, assuming an overall context of the key of A major. Thus, for example, the descending fourth progression applied to E5 produces the sequence E5–D5–C♯5–B4 rather than, for example, E5–D5–C♮5–B4 if the context were A minor. In some cases a harmonic context will be required also (for example, for an 'arpeggiation' or 'consonant skip' elaboration). For simple pieces, the tonal context will remain constant throughout the piece, but for many more pieces it will change as the piece is elaborated, and the harmonic context is liable to much more frequent change. Each note or elaboration in a tree structure therefore needs to be associated with a specific tonal and harmonic context. In most cases, the children of an elaboration will inherit the tonal context of the parent, and in many cases the harmonic context too. However, the children will often require a new harmonic context and sometimes a new tonal context, and, for the representation to be properly generative, these must be generated by the specified elaborations. Thus, for example, a 'passing-note' elaboration might indicate a harmony for the newly generated note, and this harmony should be specified by relation to the harmonic context of the parent rather than being directly specified.

There are two problems with this approach. Firstly, it is not generally the case that the harmonic context for newly generated notes can be simply specified in relation to the parent harmonic context for all elaboration types. In the case of a passing-note elaboration, for example, we know from harmony lessons that a passing note descending between the third and first degree of a major key, assuming chord I for the third degree, can be accompanied by either chord V or chord vii°. A diminished seventh is also possible, or even chord ii if the following chord is a second inversion tonic chord rather than root position or first inversion. Possibly a field could be added to elaborations to specify the kind of harmonic sequence which is to be generated.

The second problem is that, at least for pieces in the intended domain of tonal music, the tonal and harmonic context is the same for all voices over a given time span, and therefore it must be the same in corresponding branches of concurrent trees also (though segments of music with pedal notes are a restricted kind of exception). One solution to this would be to have a separate tree of harmonic (and tonal) contexts which is parallel to and shared by all the trees in a representation. This would reflect the normal theoretical account of harmonic structures, often expressed as a series of Roman numerals or bass figures, underlying all the voices of a piece of music and having a quasi-autonomous existence. However, it results in a structure where consistency is not easy to maintain, especially where changes in a tree of elaborations require changes in the tree of harmonic contexts, but the change in the elaborations is later reversed. To keep track of all the relevant dependencies might be more complicated than to maintain separate records of harmonic and tonal contexts in each concurrent tree of elaborations.

The best approach might therefore be to keep information about harmony and key within each concurrent tree of elaborations, but for that information to be about harmonic and tonal *constraints* rather than definitely specifying a particular harmony and key. This information could be a set of harmonies and associated keys with which a note or elaboration is consistent. When it is necessary to determine the actual harmonic context covering all voices for a particular time span, that context could be derived by taking the intersection of these sets. The intersection must never be empty.

As before, however, it might be useful to retain the possibility of a separate tree of harmonic and tonal contexts for those situations where a sequence of harmonies does genuinely have an existence autonomous from any specific voice. This could be achieved by a tree representing a voice whose ‘notes’ are all indeterminate in pitch but have specified harmonic and tonal contexts.

7.4 Metric contexts and uneven rhythms

Just as harmonic and tonal contexts are required for elaborations to generate the pitches of new notes, so metrical contexts are required to generate their time spans. At its simplest, the information required is whether the time span of the parent note divides naturally into two or three sub-spans. (Other divisions are either multiples of two and/or three or occur rarely enough to be ignored for the present.) Where even time divisions are not possible the repertoire in question seems to prefer to put long notes first. Thus, when three notes are required in a time span which divides into two, the normal pattern is to put one note in the first sub-span, and two notes in the second sub-span. A 4/4 metre, for example, divides each bar into two at two levels: a semibreve (whole note) divides into two minims (half notes) which each divide into two crotchets (quarter notes). The normal way of distributing three notes in a bar of 4/4 is a minim followed by two crotchets. Following the same long-short norm, two notes in a bar of 3/4 normally occur as a minim followed by a crotchet. (Note that this has not always been the case: mediaeval music in Europe often followed a short-long pattern.)

A metrical context could be attached to a note (or an elaboration) in the same way as a tonal and harmonic context, and, once again, it would often be inherited from its parent. However, at times the metrical context must change, and this must be represented in some manner in the trees of elaborations. Furthermore, it must be possible to have different metrical contexts in concurrent trees (since a simultaneous division into two in one voice and three in another is not rare), but the highest metrical level in all concurrent trees must be the same. (This is not necessarily the level of the notated bar. In cases such as the famous multiple simultaneous dances in the finale of Act 1 of Mozart's opera *Don Giovanni*, the bars of the three orchestras do not coincide, but they all share a common 'hyper-measure', i.e., a metrical level above that of the bar.)

However, not all rhythms follow the natural divisions of the metre, of course; it must be possible to represent rhythms with a short-long pattern, for example, and also syncopations. There are two possible ways of accommodating this. One is to require all divisions to follow the metre at some level, changing the metre as necessary, but to allow elaborations which effectively allow a single note to be represented by more than one node in the tree. Thus, for example, if one wanted to represent the rhythm of a dotted minim followed by a crotchet in a bar of 4/4, this could be done in two stages. In the first an elaboration would divide a semibreve (whole note) occupying the entire span of the bar and produce two minims as children, but because this is actually a non-elaboration, the two minims would be tied together to make a single note. Then at the next level below, the second child minim would be parent to an elaboration which divided this into two crotchets, the first of which would remain tied to the preceding minim, so producing the desired rhythm. The alternative is to add a field to elaborations so that their division can be explicitly specified, perhaps as a ratio of integers. The rhythm of a dotted minim followed by a crotchet would then be represented by a single elaboration with the specified ratio 3:1. The first solution allows a simpler definition of elaborations but leads to more complicated tree representations. The second leads to simpler trees but requires a more complicated definition of elaborations and temporal structure. Which is more appropriate would probably depend on the intended application and empirical studies on actual data.

A third possibility is a middle way (proposed in Marsden, 2001) which allows temporal divisions to be specified as short-long, even, or long-short. In this case the (default) even division is defined as before, according to the metre. A short-long division places the division at the point in time when the first segment of an even division would be divided, and a long-short division places it where the second segment would be divided, unless the first segment is already longer than the second. Thus the rhythm of a dotted minim followed by a crotchet in 4/4 would be generated by an elaboration which is specified to be long-short, but this same elaboration in a context of 3/4 metre would produce the rhythm of a minim followed by a crotchet. In the case of some rhythmic sequences, it will still be necessary to use one of the two techniques outlined above (either representing a single rhythm in stages, or specifying the temporal ratio directly), but the three possibilities of short-long, even and long-short do cover a very large proportion of the rhythmic patterns found in pieces of music in the intended repertoire. Furthermore, some have proposed, on the basis of empirical evidence, that musical rhythms are typically conceived of in these three terms (see Clarke, 1999, p.490).

7.5 Large-scale temporal structures

As mentioned above, Schenker did not indicate durations in graphs above the level of the foreground. Indeed, while some pieces use a consistent phrase length throughout (often four bars), and this consistency can continue in higher levels to cover the entire piece (eight bars, sixteen bars, thirty-two bars, etc.), this is not generally the case. Normally there always comes a level where any regular 'hyper-metrical' pattern gives way to patterns of less regular durations. These could be represented using explicit ratios, as outlined above, but the exact ratios of sectional divisions are rarely considered to have great significance. A more natural procedure would be to allow temporal divisions to remain unspecified at higher levels, and to emerge as a result of the detail of durations at lower levels. This might require specifying a 'tactus' at a particular level (such as the bar) when durations become significant, and allowing elaborations at that level to make explicit reference to this. Some flexibility might be required in the level at which the tactus is specified, since this is not always clear in any particular case.

7.6 Context-dependent elaborations and up-beats

Lerdahl and Jackendoff noted that grouping structure and metrical structure do not always coincide, i.e., a phrase or other smaller or larger unit of what they call 'grouping' can begin part-way through a metrical unit, producing an anacrusis or 'up-beat'. Because the notes of an up-beat belong with the following phrase, they should, to allow vertical decomposition in a hierarchical representation, be joined to the sub-tree which represents that phrase. As described above, Lerdahl and Jackendoff use the idea of an 'augmented time span' to cope with such situations, which effectively introduces a additional kind of branching into their reduction trees, one which causes the time span of the up-beat to merge with the following time span to form a new time span which is actually shorter than the two combined (because it starts later, at the down-beat), rather than the normal branching which (in reduction) causes two time spans to join to make a single time span equal to the sum of their durations.

A problem with this is that the sub-tree which represents the phrase before an up-beat does not have any indication in its structure that the final time span must be shortened to allow for that up-beat. Indeed, if a tree structure like Lerdahl and Jackendoff's were used constructively, it would always be necessary to generate the first few time spans of the following phrase before the last of the current phrase could be generated, in order to know whether time must be left for an up-beat. This severely compromises vertical decomposability—the representation of one phrase is not independent of the following one in the case of an up-beat, and, worse, the representation of a phrase does not contain any indication of whether or not an up-beat follows, so a generation procedure would have to look for one in every case.

The root of the problem is that an up-beat is strongly related to the note(s) which follow, and information would be required about the *following* higher-level note(s) in order to generate the up-beat(s), yet they occupy part of the time of the *preceding* higher-level note(s) at a higher level. It is for this reason (among others) that in earlier work (Marsden, 2001), I proposed that elaborations should not occupy a tree structure but be linked to both preceding and following higher level notes in a network. This also allows the natural accommodation not only of those elaborations which typically generate up-beats (anticipations and preceding incomplete neighbour notes) but also of other elaborations like passing notes which require information from both the current and the following time spans.

Another kind of elaboration, the suspension, requires contextual information not from the following time span but from the preceding one. A suspension could be regarded as being a kind of accented incomplete neighbour note, like an appoggiatura, but with the special characteristic that the non-harmony note is not attacked but tied to a preceding note. However, this requires information about the previous context, to be sure that there is a note of the correct pitch to tie to, and it might well be necessary, in realising that earlier note (called the 'preparation' in the theory of suspensions), to know that it does not end but is tied to a following note.

Any representation which properly accommodates such kinds of elaboration (which are quite common) must therefore either include specific links 'backwards' and 'forwards' to encode the required contextual information, or context-checking procedures (which could be quite extensive) will be required in construction. It would appear that no elaboration requires information about both the preceding and following metrical units. Special consideration needs to be given to situations where a context is absent, such as in the case of an up-beat at the beginning of a piece. One possibility is to add dummy 'rest' events so that the representation begins with a complete metrical unit.

8. A representation framework

It is premature to precisely define a representation scheme, because the best details will depend on empirical investigations and probably vary from application to application. However, a framework for a class of representation schemes can be given, building on the discussion above and ensuring the desirable properties listed earlier.

A representation will be a structure composed of notes and elaborations. Rests are included as a special kind of note (a silent note which has no pitch). A note contains information about pitch and time. (Other aspects, such as dynamics, timbre and articulation, are not (necessarily) represented.) The time of a note represents a specific span (i.e., with a start point and a duration) within the temporal frame of the piece represented. Similarly, a specific pitch is represented within the pitch frame of the piece. The time and pitch information does not need to be explicitly specified for every note, but, if not, there must be a definite procedure which can derive the time and pitch from a note's context within the representation (see below). This ensures that a representation will be constructive.

An elaboration has at least one note as its parent. If it has more than one, they must be equal in span (i.e., equal in both start point and duration). A note can be a parent of more than one elaboration. An elaboration has a sequence of two or more notes as its children. A note can be a child of more than one elaboration. The sequence of children of an elaboration occupies the same span of time as the parent. This has the consequence that the child notes of an elaboration are always shorter than the parent note(s) (and so infinite derivations are not possible). Allowing more than one parent means that elaborations and notes form a directed acyclic graph rather than simply a tree. (Formally, a directed acyclic graph is a structure consisting of nodes (or points or vertices) connected by arcs (or lines or edges), where each arc has a 'direction' (i.e., it makes a difference which node is at which end) but there are no cycles by which one could return to any node by following arcs in their given direction. A tree is also a directed acyclic graph but one in which no node had more than one arc leading to it.) An elaboration or note can be described as an ‘ancestor’ or ‘descendent’ of another elaboration or note, and no note or elaboration can be both an ancestor and a descendent of any other note or elaboration. One or more notes will have no ancestors, and this/these will be referred to as the root(s). It is not required that all roots be equal in time, but this might be a characteristic of good representations. To represent an actual piece of music, the pitch and time of the root notes must be explicitly specified. (Otherwise the representation represents an indeterminate musical pattern.) A representation in which all notes are root notes is therefore valid, but representations with a minimum number of roots will generally be preferred. For every valid elaboration, if the pitch and time of the parent note(s) are either explicitly specified or derivable, then there must be a definite procedure which specifies the pitches and times of all of the elaboration’s children on the basis only of the pitch and time of the parent(s) (including their tonal, harmonic and metrical contexts; see below), plus the pitch of the preceding or following ‘context note’ linked to, if any (see below). This (together with the decomposability of a representation into segments, discussed below) ensures that a representation is generative. The music represented consists of the set of notes which are not parents of any elaboration (which will be called ‘leaf notes’). Two pieces of music are considered equivalent if their sets of leaf notes which are not rests are equal. Thus two representations might include different rests among the leaf notes, but nevertheless represent the same piece of music.

An elaboration can have at most one link to another note from which it takes contextual information. This ‘context note’ cannot be an ancestor or a descendent of the elaboration (so the representation remains a directed acyclic graph). Furthermore, it must either immediately follow or immediately precede the elaboration’s parent(s). In the case of elaborations which link to a following context, there must exist alternative elaborations which, for the child notes whose pitch is determined by reference to the following context, produce rests of equivalent time. The alternative elaborations therefore require no link to the following context, and will be used in situations where a segment followed by an up-beat is extracted (see below).

It is an underlying assumption that the elaborations and links in a representation correspond to meaningful groupings of notes, at various levels of structure. To be more precise, where two notes *x* and *y* form a sequence (i.e., the time of *y* immediately follows the time of *x*) and either the pitch of *x* is determined by virtue of a link to *y*, or *x* and *y* have the same parent, and the pitch of *y* does not depend on a link to a following note, then notes *x* and *y* belong together. To be even more precise, any grouping of notes which includes *x* and any other note which is not a descendent of *x* must include *y*, and *vice versa*. This is assumed to indicate grouping both ‘vertically’ into phrase units and also ‘horizontally’ into voices. However, the latter assumption might not prove valid under empirical investigation. The latter part of the definition above (‘the pitch of *y* does not depend on a link to a following note’) is intended to allow a following upbeat to be detached from the preceding phrase unit, even though it shares a parent with the end of that phrase unit by virtue of its rhythm. This separation might not be appropriate, though, in dividing a segment of music into voices. I suspect that in many cases, ancestors of the separated notes *x* and *y* will be related and so cause *x* and *y* to be grouped into the same voice anyway, but there seems no reason to believe that this will always be the case. It might prove necessary to remove that final part of the condition when determining grouping into voices.

A representation is therefore ‘decomposable’ by extracting a note or a set of notes, plus their descendents (or, rather, some of their descendents; see below) plus any linked up-beat, to form a new representation of a segment of the original piece, which has this/these extracted note(s) as its root(s). Where a note is parent to more than one elaboration, at least one of those elaborations and its children should be included in the extracted representation, but it is not necessary to include them all.

Because representations are not necessarily trees, it is not the case that any note plus any valid selection of its descendents constitutes a valid representation: it is possible, because of the possibility of ‘upwards branching’ which makes a representation not exactly a tree, that information is required in the generation of descendent notes which comes from notes or elaborations which are not themselves descendents of the putative extracted root note, and so will not be part of the extracted representation. (The extracted representation would therefore not be constructive, and so not valid.) There are three possible situations to consider with respect to any descendent note *x* of the putative extracted root, illustrated in Figure 7.

The first (A in Figure 7), where *x* is a child of more than one elaboration, is not a problem. In such cases, there must be an elaboration which is a descendent of the extracted root which is a parent of *x*, and it is required that every elaboration be capable of generating all of its children, so the pitch and time of *x* can still be determined. The fact that it also derives from some other elaboration is incidental.

The second situation (B in Figure 7) is where *x* is a child of an elaboration with more than one parent, and not all the parents are descendents of the extracted root. In this case those parents must be added to the extracted representation as additional roots. (Often it will be preferable to include ancestors of these notes also to arrive at an extracted representation which has roots of equal time.)

The third situation is where *x* derives from an elaboration which is linked to a preceding or following ‘context note’ which is not a descendent of the extracted root. If it is a preceding context (C in Figure 7), once again the extracted representation should be expanded to include that note. If it is a following context (D in Figure 7), the elaboration generates an up-beat to the following segment, and so it should be replaced by an alternative elaboration which replaces that upbeat by one or more rests and which no longer requires a link to the following context. The condition stated above ensures that such a replacement is always possible.

Up-beats at the beginning of a segment (E in Figure 7) should be included in the representation of that segment, as follows. Besides elaborations which are descendents of the extracted root(s) (except those replaced by rest-generating elaborations, as explained above), the extracted representation should also include all elaborations which are not descendents of the extracted root(s), but which are linked to following context notes which are descendents. In the extracted representation, the parents of these additional elaborations should be replaced by rests with the same time.

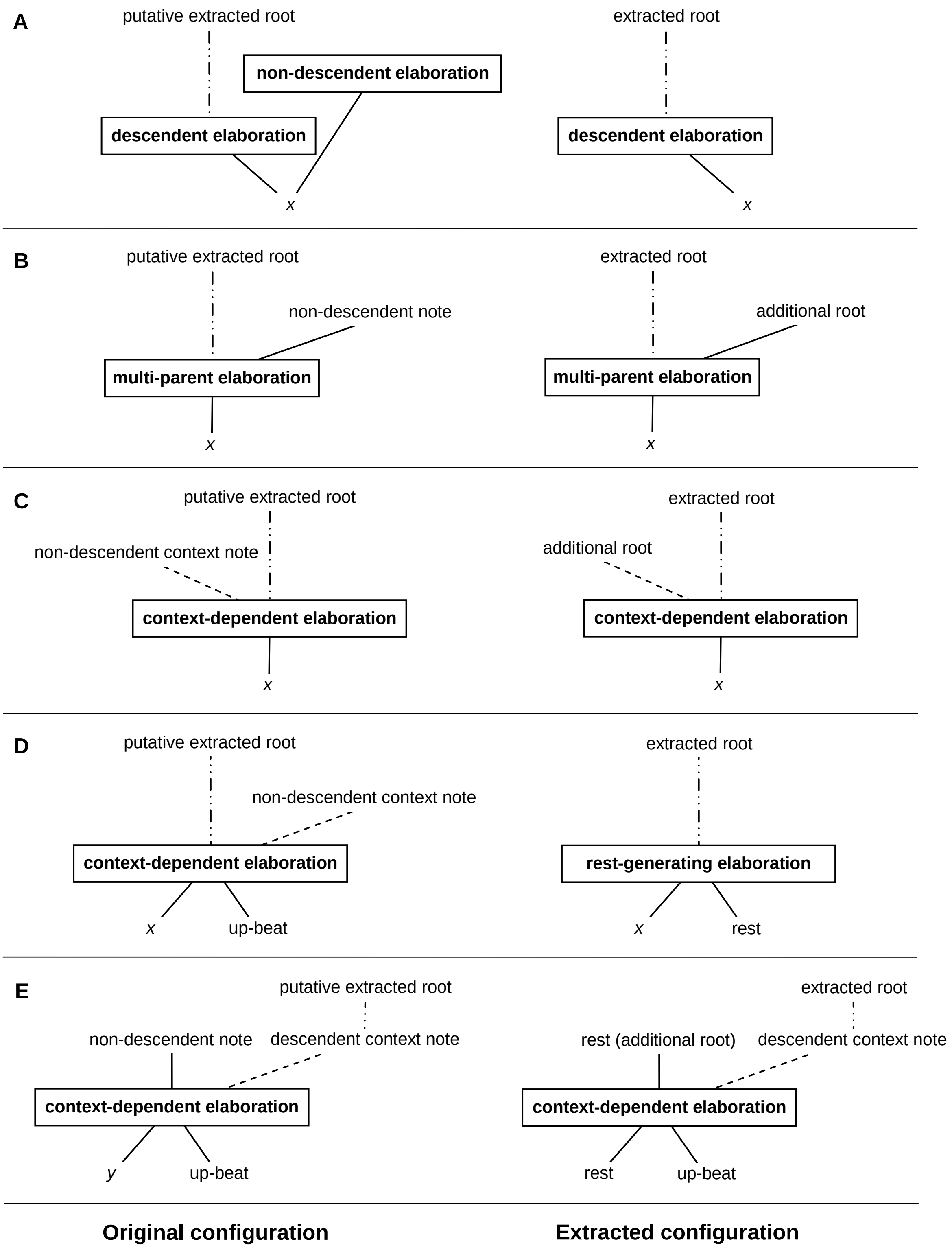

Figure 7. Decomposition of representations in the presence of ‘upwards branching’ (i.e., where the representation deviates from a simple tree structure).

The definition of decomposability also requires that the representation of segments which can be combined into a single piece should be combinable into valid representations. Combining two representations ‘horizontally’ to correspond to putting two voices together to sound simultaneously simply requires adding one representation to the other and ensuring the times of the roots of the two representations are suitably aligned. Constraints to ensure that tonal, harmonic and metrical contexts are consistent will apply just as they apply to the combination of two voices. Combining two representations ‘vertically’ to correspond to putting two segments of music one after the other can be more complex. A naïve combination is simply to define the time(s) of the root(s) of the second segment to follow the root(s) of the first. It would be preferable, though, to derive one or more elaborations with the roots of the original representations as children, creating one or more new roots for the resulting combined representation and suitably representing the structure of the combined piece rather than simply having one representation follow the other. Furthermore, if the second representation contains an up-beat, this ought to be joined with the end of the first representation. If the first segment ends with rests whose combined time is at least as long as that of the up-beat, sufficient elaborations in the first segment which generate those rests should be removed to make time for the up-beat, and the parents of the elaborations which generate the up-beat be assigned to the parents of those removed rest-generating elaborations. Effectively, this replaces the rest(s) at the end of the first segment by the up-beat of the second. Alternatively, if the first segment ends with at least one note which is longer than the up-beat, one or more such notes can become the parent for the up-beat generating elaboration(s) from the second segment, causing a shortening of these notes at the end of the first segment. (This corresponds to the ‘regular’ concatenation of musical segments described in Marsden, 2000, p.140.) Finally, in joining the two segments with new elaborations or via any up-beat, decisions would have to be made about how the voice(s) of the first segment relate to the voice(s) of the second.

A representation of this kind is clearly hierarchical. Furthermore, a manipulation which changes the pitch or time of higher-level notes (e.g., transposing them) will similarly affect all the lower-level descendents of those notes.

It was earlier stated, to ensure that a representation is generative, that it is a condition of elaborations that, given a parent note (including its tonal, harmonic and metrical context) and any necessary linked context note, there must exist a definite procedure which generates the details of the child notes for any elaboration which is valid for that parent and context note (if appropriate). To ensure that a representation is derivable also requires a definite procedure to determine for any sequence of notes—plus potential preceding and following context notes—either an elaboration and parent note(s) which can generate that sequence or that no such elaboration exists for that sequence. With this condition, and provided that the set of possible elaborations for any possible sequence of notes is finite, a representation is theoretically derivable from any finite configuration of notes. The number of sequences and sub-sequences contained in a finite configuration is finite, and so, by the condition above, the size of the set of all possible elaborations which could produce those sequences is also finite. These elaborations produce new parent notes, which in turn produce new sequences, but the length of the longest such sequence decreases at each step: every elaboration has at least two time spans among its child notes, but only one in its parent(s), even if it has more than one parent. Thus, the process of finding possible elaborations must eventually reach a situation where there are no new possible elaborations: if it has not already reached such a situation, the longest sequence will eventually become just one note long, which cannot be a child of any elaboration because all elaborations produce sequences of at least two notes. Finally, it would be possible to design a procedure which, for a given piece (i.e., a given set of ‘leaf notes’), searched this space of all possible elaborations and parent notes in a depth-first, backtracking manner, eventually finding a complete representation with the desired properties, if one exists. A representation is thus, subject to a finite set of possible elaborations and a definite procedure for determining valid elaborations in any local context, theoretically derivable.

However, the set of all possible configurations of elaborations and parents for a given piece increases in size factorially with the number of notes in the piece. (This complexity is potentially increased enormously by allowing the representation to have the structure of a directed acyclic graph instead of a tree, but the proposal here imposes tight restrictions on the possible graph configurations, importantly limiting the increase in complexity.) A naïve exhaustive-search strategy is therefore not practical. Indeed, it is clear that derivation is the hard task with respect to structural and generative representations of music. As mentioned above, the rules for derivation of an analysis by Lerdahl and Jackendoff are not directly computable, and there is, as yet, no systematic method for derivation of a Schenkerian analysis. In the domain of computational systems, Frankel, Rosenschein and Smoliar in 1976 anticipated that they 'should be able to formulate an experimental grammar to be used for automated analysis' (p.30), but by 1978 they questioned 'whether a program which produced Schenkerian analyses may be designed without a peripheral "world model" of musical perception' (p.134). Hirata and Aoyagi in 2003 'do not offer any ideas on how a time-span tree can be algorithmically derived from a musical structure' partly because 'excessive automation may yield unintended analysis results' (p.88). (However, Hamanaka, Hirata and Tojo (2004) report progress at least in automatically deriving the grouping on which a time-span reduction can be based.) Only Kassler reports automatic derivation of an analysis (1975, 1977, 1988), but his system has yet to be extended to the foreground, and the longest middleground example he gives contains just 39 notes, not counting rests and ties (1977, p.79). Yet at least semi-automatic derivation of representations is vital to their usefulness. Hirata and Matsuda report that creating a representation in their system took three hours for four bars of music (c. 51 notes), even with a specially designed editor (2003, p.172). Such a speed for what is effectively data entry is quite impractical for any real application. Derivability should therefore be a major concern in the detailed design of a representation system, and factors which are known to lead to complications, such as long-range contextual dependencies, should be avoided.

9. 'Vocabulary' of elaborations

The framework above has not specified the set of possible elaborations from which a representation should be constructed, and it would be better for this to be determined empirically. Different repertoires of music are likely to require different vocabularies of elaborations, and in some cases different ways of defining tonal, harmonic and even metrical contexts. Furthermore, some principles would be required to guide empirical investigation. One obvious principle is that the vocabulary of elaborations should allow the representation of all the pieces in the intended repertoire, but principles to restrict the size of the vocabulary will also be required. Such a principle might be parsimony—representations should use as small a set of elaboration types as possible, or representations should be as small as possible, or some combination of the two—and other principles might be derived from definition of how a representation might be 'meaningful'.

Another approach to guide the definition of a vocabulary of elaborations is to use some pre-existing music theory, a role which could be played by Schenkerian theory. As mentioned, Kassler has based his system closely on *Der freie Satz,* and it could also provide a basis for the definition of elaborations in the framework described above, since Schenker's exposition is largely by description of classes of prolongational techniques. Another possible route is to use Lerdahl's 'tonal pitch spaces' (2001), and derive sets of elaborations which pass from one note to another through a given level of the pitch space, producing various kinds of 'passing' elaborations, and sets of elaborations which move to the next note at a given level of the pitch space, producing various kinds of 'neighbouring' elaborations.

Further work is clearly required both to define the criteria for vocabularies of elaborations and to discover appropriate vocabularies for particular repertoires of music.

10. An example

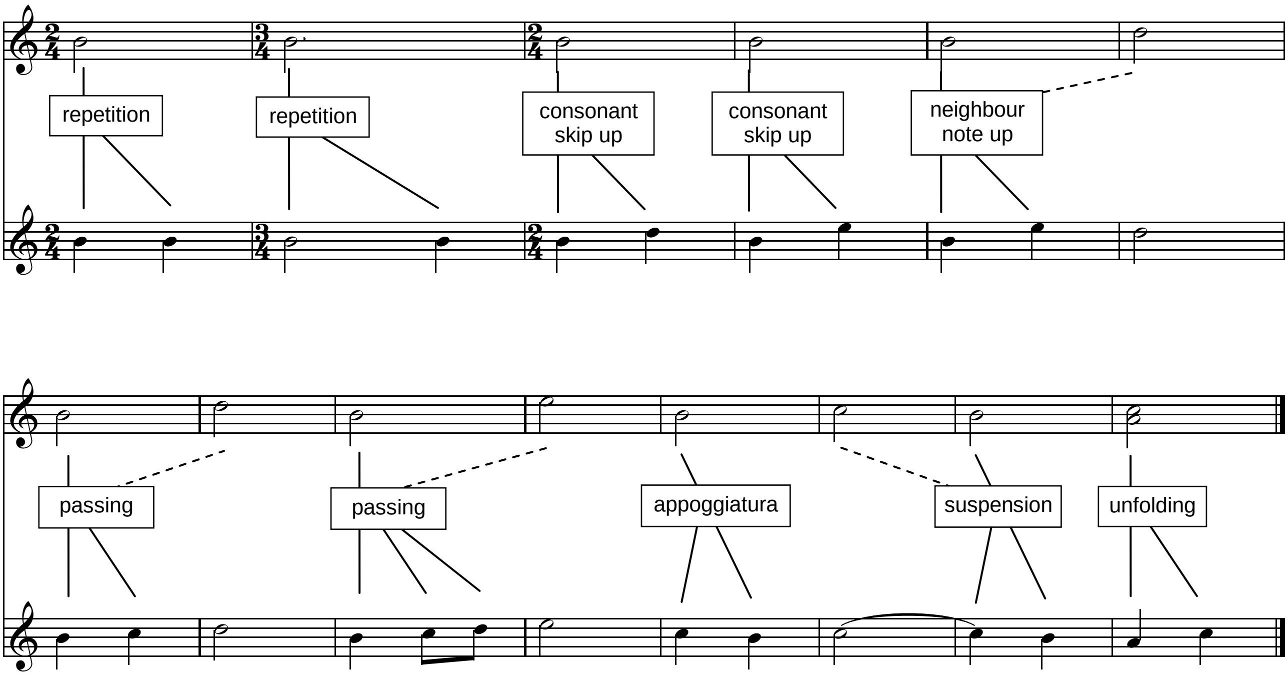

Figure 8. Examples of elaborations. The upper stave of each pair shows parents, the lower stave children.

Without prejudging the issue of a desirable vocabulary of elaborations, and without making any claim to practical derivability, the ideas in the paper are illustrated in a possible representation of the same opening phrase from Mozart's piano sonata in A major as used in the examples above. The vocabulary of elaborations used is illustrated in Figure 8. Rhythms are assumed to follow the metre unless an explicit ratio is specified. The first 'repetition' elaboration in Figure 8 applies to a minim in a 2/4 metrical context, and so two crotchets are generated. The second 'repetition' elaboration, on the other hand, applies in a 3/4 context, and so the two generated notes are not of the same duration but follow the common 2:1 pattern of minim-crotchet. Similarly, the two 'consonant skip' elaborations apply in different harmonic contexts (though this is not specified in the figure): G major in the first case produces a D following B, whereas E minor in the second case produces an E. With a following D, this last sequence of notes could be produced also by a ‘neighbour-note’ elaboration, with E as upper neighbour to the following D, as shown in the last example in the upper system of the figure, which also provides an example of an elaboration which takes pitch information from a following ‘context note’. ‘Passing-note’ elaborations take information both from the parent note and from a following context note. Furthermore, as illustrated at the beginning of the second system in the figure, they can generate a sequence of more than two notes. Although the pitch of the parent note is often copied in the first of the children, this is not the case for some elaborations, such as ‘appoggiaturas’ and ‘suspensions’, as shown in the figure, where it is the second child note which has the pitch of the parent. Suspensions also require a reference to the preceding context. Finally, an ‘unfolding’ elaboration has two (or more) parent notes of the same time (forming a chord) and generates as children notes of the same pitch, but forming a sequence.

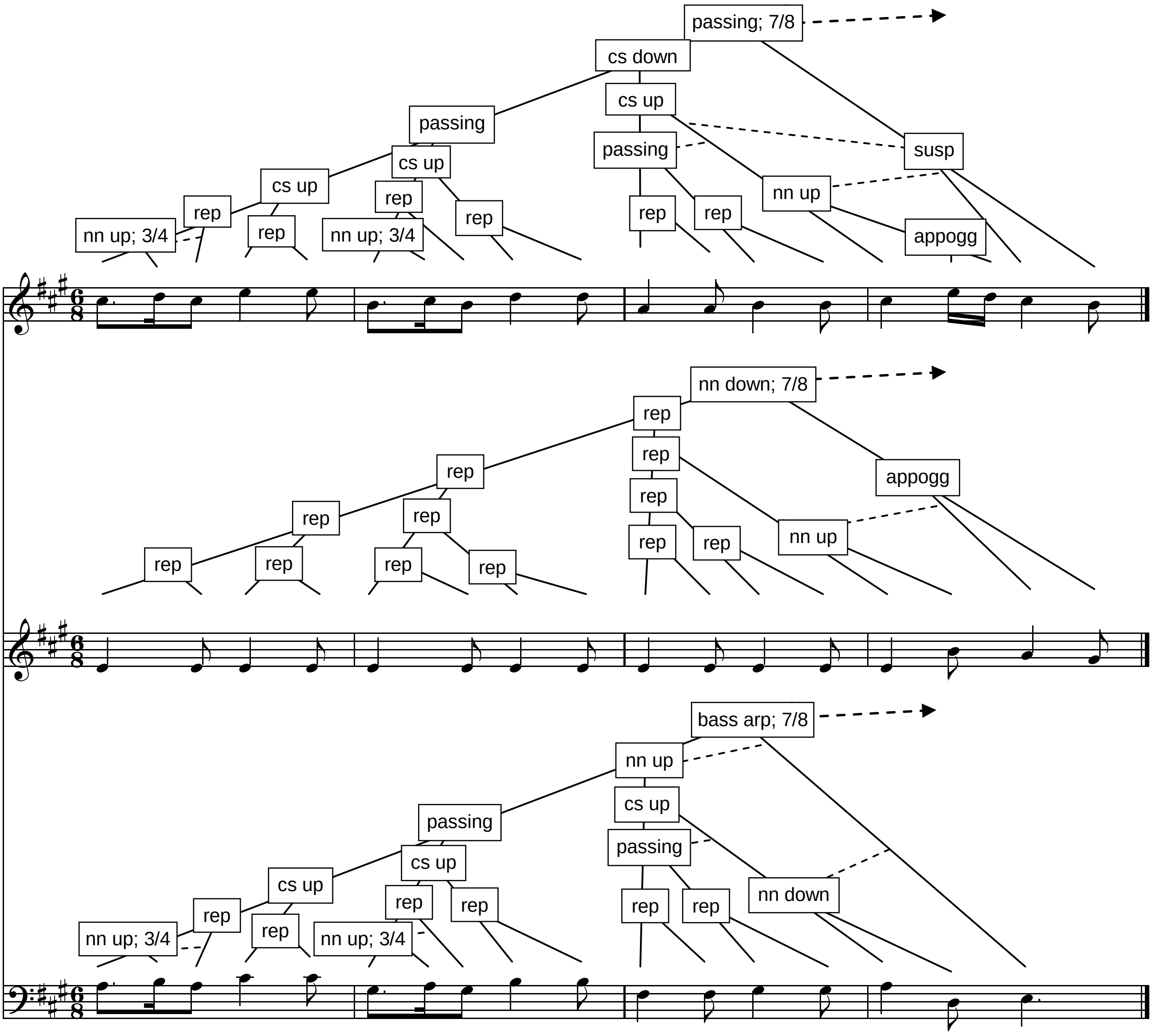

Figure 9. Simple elaboration-tree representation of the first phrase of Mozart’s piano sonata in A major, K.331.

Figure 9 shows a representation of the first phrase of the Mozart theme in this manner, representing each of the three voices by a separate ‘tree’. The ‘leaf notes’ are shown in music notation, but notes at other levels are omitted from the diagram for clarity. The pitch and time of the omitted parent notes is indicated in the manner of the time-span trees of Lerdahl and Jackendoff, i.e., the pitch of the parent of an elaboration is the same as the pitch of the note pointed to by the line which passes through the box representing the elaboration without deviating. Its time begins at the same time as that note and extends to beginning of the note which is pointed to by the branch from the elaboration directly ‘above’ this one. Thus the parent of the first ‘neighbour-note’ elaboration on the top system is C♯5 beginning at the start of the bar and extending for the duration of a crotchet to the second C♯5 in that bar (pointed to by the ‘repetition’ elaboration which comes immediately above). Broken lines indicate links to context notes and pass from the elaboration using the link to a solid line indicating the note linked to. That note is the parent of the elaboration on the solid line immediately below the point on where the broken and solid lines join, or the leaf note the line points to if there is no elaboration below. Thus the context note for the first ‘neighbour-note’ elaboration is the second C♯5, the leaf note below the point where the broken line from the box representing that elaboration joins with the following solid line. The context note for the ‘passing’ elaboration in the third bar of the first system, however, is a C♯5 which is the parent of the ‘neighbour-note’ elaboration at the start of the fourth bar. (The broken lines with arrows from each of the three highest-level elaborations are intended to connect to context notes in the following phrase.)

However, a representation which simply divides this music into three voices is not necessarily appropriate. As mentioned above, Schenker regarded the rising thirds in the first three and a half bars of the phrase to indicate an ‘unfolded’ two-voice structure, made explicit in the parallel thirds in the second half of the last bar. Figure 10 therefore shows (for the upper two voices only) an alternative representation of the music which embodies this more subtle analysis. (Some abbreviations are used in the figure and some elaborations are omitted.) Two similar and concurrent trees generate the voices in parallel thirds underlying the upper voice. ‘Unfolding’ elaborations in each of the first three bars convert these into a single voice (which is then elaborated further in the same manner as in Figure 9 in details omitted here for clarity). The final G♯4 in the last bar, and the preceding B4, are both children of two ‘neighbour-note’ elaborations which derive from the middle voice repeated E4 and the ‘lower-upper-voice’ A4 (though the first of these ‘neighbour-note’ elaborations is omitted in the figure for clarity). This illustrates the two kinds of ‘upwards branching’ which cause the joining of two voices into one: elaborations which have more than one parent note (the unfoldings) and notes which are children of more than one elaboration (the neighbour notes).

Structural representations which are constructive, meaningful, decomposable, hierarchical and generative are therefore possible. The proposed scheme is based on the tree structures common from earlier work, but it deviates from a simple tree structure (towards a directed acyclic graph) in specified and local ways, where the musical details cannot be adequately described by a tree structure alone. Important areas for future work are criteria for distinguishing good and bad alternative representations of the same music, criteria for ‘languages’ of possible elaborations, definition of languages for different musical repertoires, and definition of procedures for derivation of representations from scores (or perhaps MIDI files). In addition empirical work is required to test the adequacy of both the overall representation schemes and specific languages of elaborations, to test derivation procedures, and to test the efficacy of representations in facilitating musical processing.

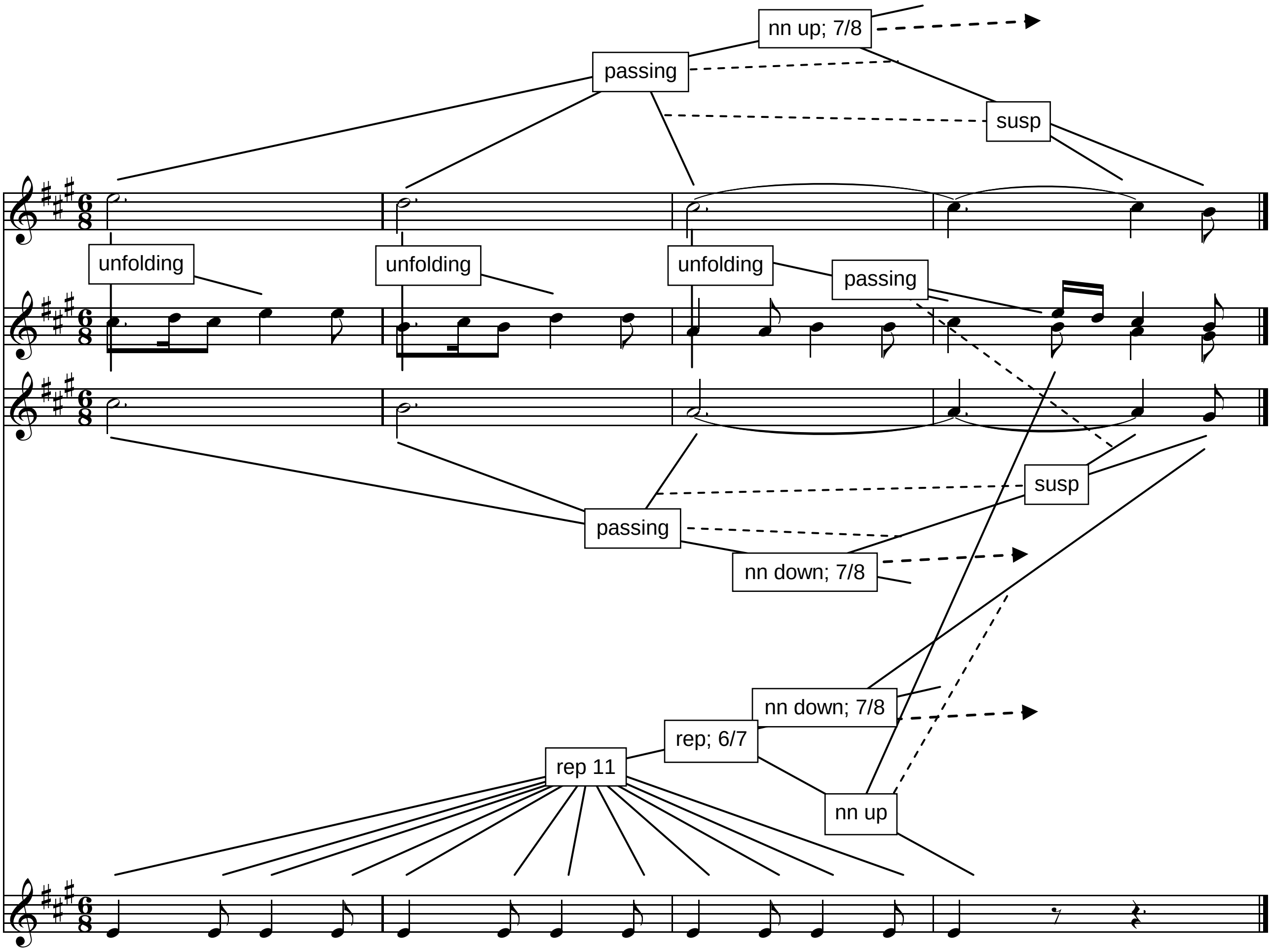

Figure 10. Alternative representation of the first phrase of Mozart's piano sonata in A major, K.331 (upper two voices only). The lowest levels of elaboration for the upper voice, which duplicate those shown in Figure 9, are not shown here.

References

Baroni, M. & Callegari, L. (1984). *Musical Grammars and Computer Analysis*. Florence: Olschki.

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. 187–205). London: Academic Press.

Baroni, M., Dalmonte, R. & Jacoboni, C. (1999). *Le regole della musica. Indagine sui meccanismi della comunicazione*. Turin : Manuali EDT/SIdM.

Bel, B. (1998). Migrating musical concepts, an overview of the Bol Processor. *Computer Music Journal*, *22*(2), 56–64.

Byrd, D. & Isaacson, E. (2003). A music representation requirement specification for academia. *Computer Music Journal*, *27*(4), 43–57. CMJ article

Clarke, E.F. (1999). Rhythm and timing in music. In D. Deutsch (ed.) *The Psychology of Music* (2nd edition), San Diego: Academic Press, 473–500.

Deutsch, D. (1999). The processing of pitch combinations. In D. Deutsch (ed.) *The Psychology of Music* (2nd edition), San Diego: Academic Press, 349–411.

Dibben, N. (1994). The cognitive reality of hierarchic structure in tonal and atonal music. *Music Perception*, *12*, 1–25.

Forte, A. & Gilbert S. (1982). *Introduction to Schenkerian Analysis*. New York: Norton.

Frankel, R.E., Rosenschein, S.J. & Smoliar, S.W. (1976). A LISP-based system for the study of schenkerian analysis. *Computers and the Humanities*, 10, 21–32.

Frankel, R.E., Rosenschein, S.J. & Smoliar, S.W. (1978). Schenker's theory of tonal music—its explication through computational processes. *International Journal of Man-Machine Studies*, *10*, 121–138.

Good. M. (2001). MusicXML: An internet-friendly format for sheet music. *XML 2001 Conference Proceedings*. Orlando, Florida, December 9-14.

Hamanaka, M., Hirata, K. & Tojo, S. (2004). Automatic generation of grouping structure based on the GTTM. Proceedings, *International Computer Music Conference 2004* (pp.141–144). ICMA, Miami, Florida.

Hirata, K. & Aoyagi, T. (2003). Computational music representation based on the Generative Theory of Tonal Music and the Deductive Object-Oriented Database. *Computer Music Journal*, *27*(3), 73–89.

Hirata, K. & Matsuda, S. (2003). Interactive music summarization based on Generative Theory of Tonal Music. *Journal of New Music Research*, *32*, 165–177.

Huron, D. (2002). Music information processing using the Humdrum Toolkit: Concepts, examples and lessons. *Computer Music Journal*, *26*(2), 15–30.

Kassler, M. (1967). *A Trinity of Essays*. PhD thesis, Princeton University.

Kassler, M. (1975). *Proving Musical Theorems I: The Middleground of Hienrich Schenker's Theory of Tonality* (Tech. Rep. No. 103). Sydney, Australia: University of Sydney, School of Physics, Basser Department of Computer Science.

Kassler, M. (1977). Explication of the middleground of Schenker's theory of tonality. *Miscellanea Musicologica: Adelaide Studies in Musicology*, *9*, 72–81.

Kassler, M. (1988). APL applied in music theory. *APL Quote Quad*, *18*, 209–214.

Kippen, J. & Bel, B. (1992). Modelling music with grammars: Formal language representation in the Bol Processor. In A. Marsden & A. Pople (Eds.) *Computer Representations and Models in Music* (pp.207–238). London: Academic Press.

Komar, A. (1971). *Theory of Suspensions*. Princeton: Princeton University Press.

Lerdahl, F. (2001). *Tonal Pitch Space*. Oxford: Oxford University Press.

Lerdahl, F. & Jackendoff, R. (1983). *A Generative Theory of Tonal Music*. Cambridge, Mass.: MIT Press.

Marsden, A. (1992). Modelling the perception of musical voices: a case study in rule-based systems. In A. Marsden & A. Pople (Eds.) *Computer Representations and Models in Music* (pp.239–263). London: Academic Press.

Marsden, A. (1996). Symbolic representation for music. In C. Mullings, S. Kenna, M. Deegan & S. Ross (eds.) *New Technologies for the Humanities* (pp.115–137). East Grinstead: Bowker Saur.

Marsden, A. (2000). *Representing Musical Time: A Temporal-Logic Approach*. Lisse: Swets & Zeitlinger.

Marsden, A. (2001). Representing melodic patterns as networks of elaborations. *Computers and the Humanities*, *35*, 37–54.

Marsden, A. (2004). Novagen: A combination of Eyesweb and an elaboration-network representation for the generation of melodies under gestural control. Proceedings *COST287-ConGAS Symposium on Gesture Interfaces for Multimedia Systems (GIMS)* (pp.52–57). AISB 2004, Leeds University.

Oura, Y. & Hatano, G. (1991). Identifying melodies from reduced pitch patterns. *Psycholgica Belgica*, *31*, 217–237.

Plum, K-O. (1988). Towards a Methodology for Schenkerian Analysis (trans. and ed. William Drabkin), *Music Analysis*, 7, 143–164.

Pope, S.T. & Van Rossum, G. (1995). Machine tongues XVIII: A child's garden of sound file formats. *Computer Music Journal*, *19*(1), 25–63.

Rahn, J. (1980). On some computational models of music theory. *Computer Music Journal*, *4*(2), 66–72.

Schachter, C. (1999) Either/Or. In C. Schachter (ed.) *Unfoldings*, Oxford: Oxford University Press, 121–133.

Schenker, H. (1935). *Der Freie Satz*. Vienna: Universal Edition. Published in English as *Free Composition*, translated and edited by Ernst Oster. New York: Longman, 1979.

Smoliar, S.W. (1980). A computer aid for Schenkerian analysis. *Computer Music Journal*, *4*(2), 41–59.

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

Steedman, M.J. (1995). The blues and the abstract truth: Music and mental models. In A. Garnham & J. Oakhill (Eds.), *Mental Models in Cognitive Science* (pp.305–318). Mahwah, NJ: Erlbaum.

Van Noorden, L.P.A.S. (1975). *Temporal Coherence in the Perception of Tone Sequences*. PhD dissertation, Technische Hogeschool, Eindhoven, Netherlands.

Warren, R.M., Obusek, C.J., Farmer, R.M. & Warren, R.P. (1969). Auditory sequence: confusions or pattern other than speech or music. *Science*, v.164, 586–7.

Wiggins, G., Miranda, E., Smaill, A. & Harris, M. (1993). A framework for the evaluation of music representation systems. *Computer Music Journal*, *17*(3), 31–42.

Wiggins, G. & Smaill, A. (2000) Musical Knowledge: What can Artificial Intelligence Bring to the Musician? In Eduardo Reck Miranda (ed.) *Readings in Music and Artificial Intelligence*, Amsterdam: Harwood Academic Publishers, 29–46