A Structured Framework for Representing Time in a Generative Composition System

Francisco Pereira (*), Carlos Grilo (*), Luis Macedo (**), Amílcar Cardoso (*)
(*) CISUC - Center for Informatics and Systems, Univ. Coimbra, Polo II, 3030
Coimbra, Portugal
(francisco@alma.uc.pt, grilo@alma.uc.pt, amilcar@dei.uc.pt)
(**) Instituto Superior de Engenharia de Coimbra, 3030 Coimbra, Portugal
(macedo@alma.uc.pt)

Abstract

The representation of music structures is, from Musicology to Artificial Intelligence, a widely known research focus. It entails several generic Knowledge Representation problems like structured knowledge representation, time representation and causality.

In this paper, we focus the problem of representing and reasoning about time in the framework of a structured music representation approach, intended to support the development of a Case-Based generative composition system. The basic idea of this system is to use Music Analysis as foundation for a generative process of composition, providing a structured and constrained way of composing novel pieces, although keeping the essential traits of the composer's style.

We propose a solution that combines a tree-like representation with a pseudo-dating scheme to provide an efficient and expressive means to deal with the problem.

1. Introduction

The problem of time representation is, as with other abstract entities, a problem of representing a strictly structured and organized world through the use of mathematical abstractions. From a computational point of view, a convenient abstraction should account for the expressiveness needs and the complexity of the situation.

For tasks like music composition or story making, the ability to reason about temporal and hierarchical relations between events is far more important than knowing their absolute temporal positions.

In this paper, we present the use of a kind of "pseudo-date" [1] way of representing time in domains in which we must deal with temporal objects that are defined hierarchically, which we call structured temporal objects. The dating representation, as old as the notion of time

itself, seems to be a very coherent and mature way of representing structured time environments. We apply this concept to relate an object's position in the hierarchy to what we call a temporal address. This representation has the peculiarity that one may always get the needed granularity, which seems useful in music analysis and composition, since the analyst comprehends time according to level (e.g., an analyst may divide a Bipartite Sonata in it's first and second part, according to tonality changes, then divide each part on its sequence of sections, each of which will consist on a sequence of motives, etc.). This hierarchical view of time in music is referred in other works [4,11] but none has explicitly used its potential.

In Section 2 we'll make a brief description of some music representations. We don't describe them in detail because we think a presentation of generic ideas will suffice for the purpose of this paper.

Section 3 is dedicated to our generative composing system. A brief description is made in order to provide an easy understanding of the problems we need to solve and the applicability of the proposed solutions. Those problems are described in more detail in Section 4, while Section 5 is entirely dedicated to the presentation of our approach and the discussion of its applicability to composition in structured domains, and its advantages and limitations.

In section 6, we give an example of the applicability and usefulness of the representation.

Finally, in section 7, we present some conclusions and discuss some pointers for further work around this subject.
Ver

2. Music Representations And Time

There exist many music systems, devoted to tasks ranging from simply playing to more elaborated ones,

such as composition and analysis, each with a particular approach to time representation.

If the goal is to represent music for purposes of playing a sequence of sounds, as in the listener's point view, we may use representations like the universally used MIDI representation, the WAVE format, or any other sound sequence representation.

If the goal is to represent a sequence of finger positions, we might use representations like the one proposed by Charnassé et al [5].

For analysis purposes, structured representations like Hierarchical Music Structures representation [4,11] or Abstract data type representation [13] would be appropriate.

For composition, the representation must be somewhat generative, like in a grammar representation [9,14] and also hierarchically structured, like the ones referred before.

Both the analyst and the composer use a hierarchical structure to represent music complemented by an antecedent-consequent (causal) relation network. The difference is in the scheme used by each other when processing the information. The analyst deals with complete information when determining the structure for a given music piece: he knows it beforehand. The composer deals with incomplete information when synthesizing a novel music, by creating an underlying structure and, based on it, applying the ideas that come to his mind so to create a complete and structured piece of music (or set of combined ideas).

The goal we have in mind is analysis-based composition, therefore we should seek for a commitment between these two ways of dealing with music structures.

In the above mentioned approaches, the representation of time is either based on a flat and continuous sequence of points (e.g., based on a unit like the second or the quaver) belonging to a temporal line, or implicitly determined by the sequence of events, the relations among events being themselves structured or not.

We need a representation that enables us to know what is the time position of a given event (a note, a chord) and also which relations (temporal - e.g., A before B - and causal - e.g., harmonic, melodic and structural) hold between it and all other events and groups of events (e.g. a motif).

So, a mixed representation should be a good choice: one that gives a precise information describing the location, relative to the structure of a musical event and, at the same time, provides a simple and easy way to obtain any temporal relation with other events (e.g. the introduction melody of a first part of a piece determines in some way the introduction melody - sometimes similar - of the second part of the same piece).

3. An Application in Musical Composition

In this section we present a musical composition system where we have already applied our time representation. This system composes new musical pieces using a case-library with several analysis obtained from expert musicologists. In our case, we have a series of analysis of pieces from a seventeenth century composer.

Our approach consists on using past cases to solve problems (much like in the normal CBR method), but intends to deal with problems that don't really need a usual or copied solution [6,7], so it changes and adapts the past cases in order to get a different new case. The choice of cases for change and adaptation is made using a specific metric function [8] that weights several aspects of the chosen case or piece of case (snippet, [10]), according to the domain's characteristics and user's choice.

In our representation, a music (or case) is composed by a tree-like structure of nodes (case-pieces) and links (part-of links), augmented by a set of causal relation links.
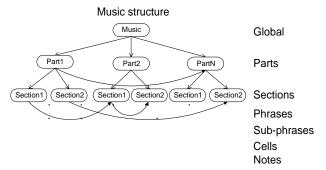

Figure 1 - A typical music structure

When the structure is used to represent an analysis in the case-library, the causal relations are taken as explanations for specific events. When generating a new solution, these relations become "suggestions" and are taken as clues for new parts of music.

For example, a starting motif in a piece may be strongly related to the starting motif of the second half of the referred piece (for instance, it may have the same melody, transposed by an ascending fifth). When we use this starting motif in our generative process, this link indicating that there will be a similar melody in the future, will be taken as a suggestion, to be considered for the new solution. This means that the link may be considered as an idea to work with when composing later parts in the new music.
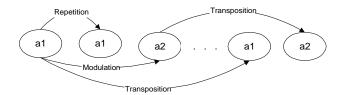
Figure 2 - In this example, we have a typical set of relation links between motifs (a1 and a2). Motif a1 begins and is repeated once. Then the "answer" is a2 in a different tonality (here a modulation occurs). Later, the situation is repeated, with no repetition of the a1 motif.

The generation process is guided by these causal relations in the sense that they give suggestions on which case-pieces (known as case-nodes) to choose on subsequent steps. These suggestions may vary in strength, according to the analyst's point of view (some may be more important than others).

For the propagation of these suggestions on later parts of the music, it is extremely important the use of relative temporal references, since the exact position of the destination of the suggestion may be not yet known when it is propagated.

4. Issues

Applying this generative process in a music domain gives rise to two important issues:

- We must decide how to represent structured temporal objects, considering that every part of a music (e.g. a section, a motif, a bar, a quaver) has a time duration which is defined by its sub-parts. So we can say, for example, that a section has a duration of 4 motives, a motif has a duration of 4 bars, and so on.
- When generating new parts of a music, new relations and suggestions appear, based on the original music pieces. How to adapt these new suggestions to the new music? (e.g.: in the original music, there is a relation between an introductory melody and certain melody in the middle of the final section; in the new music, how should we represent, as a suggestion, this "middle of" notion so to allow a repetition of the melody, even when there is no final section yet?)

5. Solutions

5.1. Time Structured Representation - Syntax and Semantics

The solution we propose combines a tree-like representation with a pseudo-dating scheme to provide an efficient and expressive means to deal with the problem. As a node's position in the tree is represented by an address, we may establish a correspondence between addresses and periods, so that the addresses show the temporal relations that exist between nodes.

The address of a node in level n is represented by $Nn:Nn-1:...:N0$, where each $Ni \in N_0$ (from now on, we will call offsets to the Ns). An offset $L=Ni$, $0 <= i < n$, means that the node with that address has a predecessor in level i of the tree which is the L-th son of its father (with the exception of the node in level 0, which doesn't has ascendants and has always offset 0). The offset $J=Nn$ means that this node is the J-th son of its closer ascendant. Every node propagates its address to its descendants, that is, if the node's address is $Nn:...:N0$, its M-th son's address will be $M:Nn:...:N0$.

This representation explicitly embeds, in its syntax, the position that a node and its ascendants occupy in the tree relatively to the others. Also it implicitly embeds the level to which the node belongs, since the number of offsets gives us this information.

It's worth noting that the tree nodes don't have all the same duration, and in consequence, each address is not committed with a fixed portion of time as is the case of the standard representation of time in digital clocks. We can say that a node has the length of its descendants, and that if it hasn't descendants, it has an intrinsic value (in the last level, the length of the notes that compose it).

Moreover, the dating scheme provides an efficient means to case-node retrieving, as we may use addresses as indexes in the case library.

5.2. Expressiveness

During the process of composition, it is not very important to know the absolute time at which events occur. Composers or writers, for example, are more concerned with temporal relations that exist between the events of a music or a story, as well as with the content relations that exist between them. Similarly, analysts are more concerned with relations such as "The motif a1 belongs to section A that belongs to the first part of the music" or "The first motif of the piece is repeated in the beginning of the second part in the tonality of the dominant".

Our time representation for structured temporal objects, applied to music composition, focus precisely on the aspects of structure and hierarchy that exist between the components of a musical piece, giving also information on the relative position they occupy inside it. This is in obvious contrast with Balaban's [4] representation of time in music domains, where, although we can establish all temporal relations between events, their position inside the musical piece is represented in an

absolute way, requiring some additional effort to establish those relations.

More precisely, we can show that our representation can express by its own, all the temporal relations between periods defined by Allen [2, 3]. As pointed by him, pseudo-dates have the advantage of requiring little computation in the comparison of two periods. In fact, with this representation, we almost don't need to make network search to establish the temporal relations between all the nodes. As an example, if a case has a node A, with address 1:0:1:0, and a node B, with address 0:1:0, such that holds(A, α) and holds(B, β) we can easily verify that period α is during period β, and is met by the period γ such that holds(C, γ) and C's address is 0:0:1:0. Also, we may verify that γ starts β.

5.3. Pragmatism

The structured representation we propose offers a direct means to represent the hierarchical relations of a typical music structure like the one presented in Figure 1. As we are going to see, it also offers a convenient way to represent the causal links (suggestions) of the music structure.

As these links represent relations between pairs of nodes, we may define each link as an attribute of the origin's node and associate to it the address of the destination node.

Considering the nature of the paradigm (CBR) we are using in our system, it is important to represent the addresses used by the suggestions in a sufficiently adaptable way so that it is possible to include them in new cases having a different structure from that where they were originally used. This allows us to treat musical events (in all aspects: melody, harmony, rhythm, etc.) in many different ways, increasing the creativity potential of the system. Besides, we intend to represent diagonal suggestions (e.g., the link from node 0:0:0 to node 0:0:1:0 in figure 3).

To make this possible, instead of using absolute addresses, we label each suggestion with a relative address, with two components:

- a horizontal component, consisting in the offsets which represent the relative horizontal displacement of the destination node from the origin node;
- a vertical component, consisting in the offset which represent the relative vertical displacement of the destination node from the origin node.

With this representation for addresses we may establish links (or suggestions) in all directions, and get the required adaptability, since it allows the same suggestion to be used in different situations.
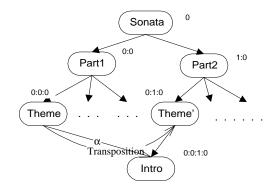

Figure 3

For example, in fig. 3, we see a very common relation in the baroque music: the introduction theme is repeated, transposed, in the second part. With our representation, the nodes' addresses are 0:0:0 (the origin - the introduction on the first part) and 0:1:0 (the destination: the introduction theme on the second part). To establish a link between them, rather than referring the complete addresses, we only need to use an H component, in this case, 0:1, which may be seen as beggining-of:1 or as 0:following, or even (generalizing) as beginning-of: following. (this means that the destination address is "the beginning of the following part"). The rest of the address isn't necessary at all, because it is similar in both nodes.

As another example, in other domain, suppose that we want to express something like "I'll be back in the beginning of next month". Within this representation, this would correspond to a relation labeled "beginning of: following" to link the present decision to its future effect (of course this depends on the assumed granularity).

5.4. Why is this useful?

In the act of generating a music, the less the choice is constrained, the more creative it probably is. Of course, there must be some coherence guidance in the process, otherwise strange, unpleasant and incoherent music is created. As said before, this guidance is given by the suggestions, which must be sufficiently non restrictive in order to give some freedom.

How does this temporal representation scheme help in this problem?

First of all, it introduces some structural relativism, i.e., each suggestion label must have only coordinates relative to the structure. For example, in fig.3, the label α corresponds to (H=0:following; V=0). The rest of the structure is dependent on the context (suppose "theme" was placed in a part2 of a new music, with address "0:1:0"; this link would point to "0:0:2:0").

Second, it is independent of the objects' duration, i.e., a link from a to b (corresponding to a musical duration of

N beats), can be applied to a new situation of, say a' to b', with a duration of $M \neq N$.

Finally, since this representation agrees with analyst's and composer's own perception of musical structure, it simplifies the act of applying directly their ideas.

As an example, suppose that, while composing a new music, the system decides to place the introduction theme of figure 3 (address 0:0:0) in a completely different place (for instance, 2:3:4). With the H and V coordinates, it is possible to "drag" the related links to the new position. The resulting relation is: "2:3:4 is repeated, transposed, in 0:4:4"[1]

This time representation itself brings new ways of generating solutions and increases the potential creativity of the new pieces of music.
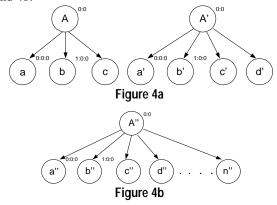
A problem of complexity now arises. What is the complexity of this representation? How large is the space of objects with which we deal? First of all, we assume that the granularity is limited to a small number of levels (at least in music analysis domain this is a natural assumption). In our first implementation, we decided to have no more than 6 levels in the hierarchy, giving addresses of the form N5:N4:N3:N2:N1:N0 in the last level. In musical composition, each node subdivides (normally) into no more than 8 components (for the first levels, this number is much lower, usually less than 4). This gives an idea of the space dimension we deal with.

Another important point to pay attention relates the role of this representation in case-piece evaluation. The question is: how can we compare case-pieces in order to choose good solutions? For the kind of tasks we deal with, it is vital to evaluate features like internal context (e.g., notes in a phrase), external context (hierarchy, structure) and temporal position. In our system, this evaluation is made according to a similarity metric developed by Macedo et al[8],where our temporal representation assumes two main roles: first, it simplifies the access to the external context; second, a node address is an important attribute to take into account when comparing nodes. The latter part demands a more detailed explanation.

In a similarity metric, we want to measure how close two objects are, in a space defined by their attributes. In what matters to temporal position, this problem is not simple. Two objects may be very close in time, and yet have little relation at all (even temporally speaking). For example, we may have three objects a, b, and c, with positions, respectively, 3:2:1:0, 5:2:1:0 and 3:2:0:0). Here, although a and b are closer to each other, object c is

probably related in some way o a, since it's also the fourth phrase (3:*:*:*) of a third section (2:*:*)[2].

Also, the relative position of a node within its context is important. For example, consider the cases in Figures 4a and 4b.

Figure 4a

Figure 4b

If we want to know which of the nodes b' and b'' is the closest to b, the first thing to note is that they all have the same address (1:0:0). Evaluating the context, we can see that b' is probably closer to b than b'' is, since their contexts are structurally more alike. To make this comparison, we only need to consider addresses of the form '*:0:0', counting, for each case, how many elements match.

6. An example

Let's consider that we have a case library with the two cases of Figure 5. The nodes labeled a_i, b_i $i \in N$, may be seen as sets of different ideas (e.g. musical motifs, ideas in a story,…) for the generation of a new case.

---

1 For the case of H coordinate being 0:following. If it was 0:1, then nothing happened, since it would be pointing backwards!

---

[2] Note that addresses start with a 0 (0:2:0:0 is the first phrase of the third section). The asterisk(*) means "don't care".
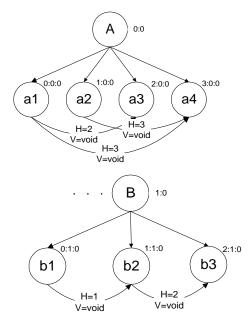
Figure 5 - Case 1 and Case 2

Figure 6 - New case

Each idea belongs to a section and a part (of a music or a story, for example). This relation is indicated by the straight arrows that don't have an H and V label.

Each of these structural relations originates a new level, and consequently, adds a new offset. For example B (address 1:0) of case 2, originates 3 structural relations (with b1, 0:1:0; b2, 1:1:0; b3, 2:1:0). So, an introduction theme of the second part of the first movement would have 0:1:0 for address. This kind of time reference is extremely useful for musical composition.

The causal relation links are represented by the arrows labeled with H and V values. Each H and V corresponds to the Horizontal and Vertical[3] component of each relation. For example, a1 of address 0:0:0 (the beginning of everything) is related to a3 (suppose it's a repetition of the original idea). The horizontal component of this relation is 2, so, to convert the first address in the second one, all we must do is simply to replace the first "0" by "2".

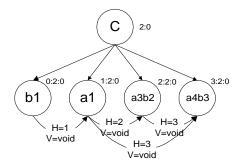Suppose the system has created a new solution, with a C part (Figure 6).

For the beginning, "b1" of B was chosen by applying its attached suggestion (the original relation to "b2", let's call it α) in C. To succeed with "b1", and respecting the α link, the system considered interesting to choose "a1" (and all of its relations). In doing so, the system is generating a new idea based on the combination of A and B. Now, the "a1" links, with the help of the H and V attributes, propagate to 2:2:0 and 3:2:0, which maintains the coherence of the original idea. In the rest of the generation process, the system has chosen to join "a3" to "b2", thus originating new different links that initially belonged to "a3" and "b2". The true power of this is that now it is possible to propagate relations (intuitions, suggestions…) to later ideas in order to transform them (now, it's possible to generate an idea "a4b3" based on a combination of the "repetition" of "a3b2" and the "inversion" of "a1").

Notice that the original relations of each node have been maintained (now used as suggestions).

7. Conclusion

In this paper we have presented a way of representing hierarchically defined temporal objects. This kind of objects is characteristic in tasks like music composition and analysis and story making. In music composition, features like granularity, expressiveness and the ability to reason about temporal and hierarchical relations between objects, are fundamental in choosing an adequate representation of time. The solution we presented here, a kind of "pseudo-date" [1] scheme, allows the needed granularity (simply by adding offsets on the left of the address), the required expressiveness (the set inclusion relations are explicitly declared, allowing an easy derivation of other relations) and as an easy way to relate objects (with the use of causal links).

In a Case Based Reasoning approach, like the one described here, it is very important the versatility of a case, that is, the ability to adapt it to new situations. The versatility of a case depends directly on its representation and, for a generative-type task (like generating musical

---

[3] For the sake of simplicity, we have just used H coordinates. In this example, V is always "void", but its use is similar to the H coordinate. Also, for the sake of simplicity, we have just chosen numbers for these coordinates, instead of abstract references (like "beginning of", "following", etc.)

pieces from a case-library composed of several analysis obtained from expert musical analysts), representing time and relations between objects are fundamental issues.

By now, our system has already generated several different music pieces. It was proved, by experience, that it's possible to generate one different music just by using one musical piece in the library! This is in part, due to our flexible time representation and to the H and V propagation scheme.

Despite the expressiveness demonstrated by this representation for the task in hand, we think it may be improved with the use of even more flexible and abstract ways of referring events (e.g. the use of a suggestion directed to "beginning-of:1:0" instead of two suggestions directed to 0:1:0 and 1:1:0, with 1:0 having more than 2 descendants).

For what concerns to other domains, we think these concepts can be valuable in similar situations, like story creation, planning and Natural Language Processing, design, etc. Instead of

Although we have applied this representation to musical composition, we think that it can be used in other domains where temporal relations between events may be more important than their absolute position (e.g., Design and Planning), and in which one has to deal with incomplete knowledge during the creation process. Moreover, besides representing temporal relations, we think this framework may easily be adapted to represent structural relations among any kind of objects, spatial relations, abstract sets and taxonomies.

Aknowledgements

References

[1] J. F. Allen (1991) - Time and Time Again: The Many Ways to Represent Time. International Journal of Intelligent Systems, Vol. 6, pp. 341- 355.

[2] J. F. Allen (1983) - Maintaining Knowledge about Temporal Intervals. ACM 26(11), pp. 932-843.

[3] F. Allen and P. J. Hayes (1989) - Moments and Points in an interval-based temporal logic. Computational Intelligence, An International Journal, Vol. 5.

[4] M. Balaban (1992) - Musical Structures: Interleaving the Temporal and Hierarchical Aspects in Music. In Understanding Music with IA: Perspectives in Music Cognition, MIT Press, pp. 110 - 138.

[5] Charnassé and B. Stepien (1992) - Hélène Charnassé & Bernard Stepien. Automatic Transcription of German Lute Tablatures: An Artificial Intelligence Application. In Computer Representations and Models in Music, Academic Press limited, pp. 143-170.

[6] L. Macedo, F. Pereira, C. Grilo and A. Cardoso(1996a) - Solving planning problems that require creative solutions using a Hierarchical Case-Based Planning Approach. In Proceedings of the Knowledge Based Computer Systems (KBCS'96), Mumbai, India.

[7] L. Macedo, F. Pereira, C. Grilo and A. Cardoso(1996b) - Towards a Computational Case-Based model for creative planning. In Proceedings of the First European Workshop on Cognitive Modeling (CM'96), Technische Universität Berlin.

[8] Macedo, F. Pereira, C. Grilo and A. Cardoso (1996c) - Plans as structured networks of hierarchically and temporally related case pieces. In Proceedings of the 3rd European Workshop on Case-Based Reasoning (EWCBR'96), Springer-Verlag.

[9] J. Kippen & B. Bel (1992) - Modeling Music with Grammars: Formal Language Representation in the Bol Processor. In Computer Representations and Models in Music, Ac. Press ltd., pp. 207-232.

[10] Kolodner (1993) - Case Based Reasoning. Morgan Kaufman Publishers.

[11] F. Lerdahl and R. Jackendoff (1983) - A Generative Theory of Tonal Music. Cambridge, Mass.: MIT Press.

[12] M. Minsky (1981) - Music, Mind and Meaning. In Machine Models of Music, MIT Press, pp. 327 - 354.

[13] Smaill, G. Wiggins, M. Harris (1993) - Hierarchical Music Representation for Composition and Analysis. Via Web page http://www.music.ed.ac.uk/research/aimusic/AI&Music.html.

[14] Sundberg and Bjorn Lindblom (1993) - Generative Theories in Language and Music Descriptions. In Machine Models of Music, MIT Press, pp. 263 - 286-.