Iscore: A system for writing interaction

Antoine Allombert
LaBRI/Université de Bordeaux/IRCAM
Laboratoire Bordelais de Recherche en Informatique
351, cours de la Libération
33405 Talence, France
allomber@labri.fr

Myriam Desainte-Catherine
LaBRI/Université de Bordeaux
myriam@labri.fr

Gérard Assayag
IRCAM
Institut de Recherche et Coordination Acoustique Musique
1, place Igor Stravinsky
75004 Paris, France
assayag@ircam.fr

ABSTRACT

In this article, we present the development of research carried out to design a system of interactive scores for composition and execution, based on temporal constraints called *Iscore*. This system has been designed in order to allow musicians to interpret pieces of electro-acoustic music but since it uses a symbolic representations of the scores, it can bee seen as a system for more generally writing interaction with a temporal approach. Then, numerous other applications could be possible. This system can be used to create interactive multimedia scenarios (for theater shows or museum visits as examples) and also for adapting musical pieces or interactive multimedia documents for mutli-player applications or players with limited ability.

Categories and Subject Descriptors

P.1 [**Entertainment, Art and Technology**]

General Terms

Algorithms

Keywords

Interaction, Temporal constraints, Music scores, Multimedia scenarios

1. INTRODUCTION

While composing an interactive musical piece or more generally, while writing an interactive multimedia scenario, one must often create musical parts or semantic elements before bindings them to interactive events or computing programs. But on the one hand, existing systems for writing music offer limited real-time interaction, and on the other hand, programming languages used for synthesizing artistic elements like Max/MSP for music, don't provide sophisticated tools for composition. We assume that a new kind of system is needed for composing or writing which can mix both of the approaches: providing an efficient environment of composition for building the parts of the work as well as programming tools for specifying interaction computation. In this paper we propose such a system that we are developing. *Iscore* (for "interactive score) is a computer-assisted composition tool that allows composers to build musical parts and structures of their pieces, binding them with temporal logical relations and adding discrete interactive events to control triggering some structures. While this system is still a work in progress, we have an implementation of it that allows us to test our choices and concepts.

2. BACKGROUND

2.1 Static Scores

Our system ensues from a previous study carried out at the SCRIME on a system of computer assisted composition based of temporal constraints called *Boxes* [4]. We present a score written with this system in figure 1. In this environment, the musical elements of the piece are represented by boxes and the composer can put them on a sheet in which the horizontal axis represents the time line while the vertical one has no specific significance. *Boxes* is close from the environment of the *Maquettes* of *OpenMusic* [3] a computer-assisted composition program developed by the *Representations Musicales* team of the IRCAM.

The main characteristics of the *Boxes* system are:

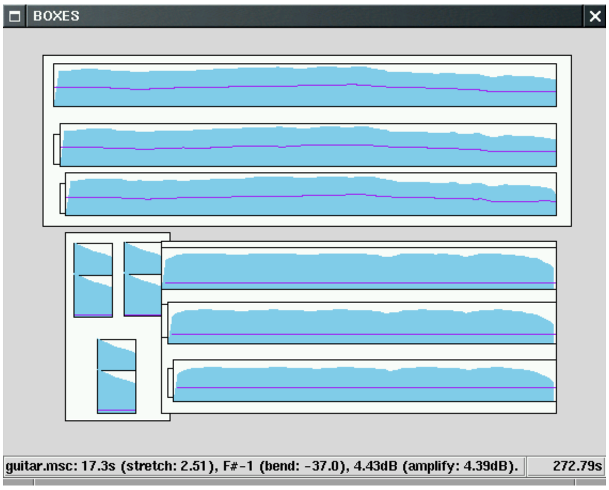
- a hierarchical model
- the use of temporal constraints

The composer can include some boxes into a box and therefore define structures of different levels. Then he can gather some notes into high-level symbolic structures such as chords or voices. Secondly, the composer can bind boxes with temporal constraints. These binary constraints are the Allen relations [1] presented in the figure 2.

As we can see, the relations describe the possible temporal configurations of two elements with non-zero duration.

With these constraints, the composer can design a temporal organization of his piece that the system will maintain during the composition stage. Once Allen relations had been

DIMEA'08, September 10-12, 2008, Athens, Greece.



guitar.msc: 17.3s (stretch: 2.51), F#-1 (bend: -37.0), 4.43dB (amplify: 4.39dB). 272.79s

Figure 1: An example of a score composed with *Boxes*

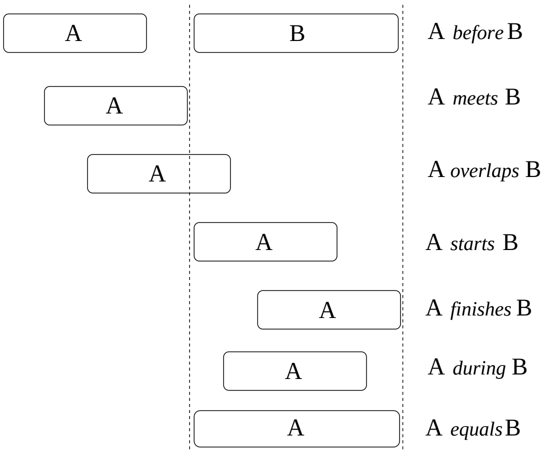


Figure 2: The Allen relations

introduced in the score, when the composer changes the position of a box, the system will compute new values for the dates of the beginning and the end of the other notes such that the new dates respect the Allen relations and the position of the box modified by the composer. It allows the composer to define a global organization of his piece and after that, refine the position of each box without changing the organization of his piece. It's important to notice that this system emphasizes on "writing the time" during the composition phase since the composer can first define the temporal relations between the structures of his piece before adjusting their positions. Then he can really focus on the meaning he want to give to the temporal building of his piece.

Finally, *Boxes* uses a spectral model for synthesizing the sounds. This system has been designed for the composition of static musical pieces. This means that after the composition stage, the date of the beginning of each note is computed, the sounds are synthesized and none will be able to change anything of it during the playing stage. We decided to extend this approach to allow the creation of interactive scores.

2.2 Interactive Scores

The evolution of techniques and music during the XX^{st} century and especially the invention of the "electro-acoustic" music which is composed of any kind of sounds and not any more traditional notes, led the composers to create "on tape" pieces that one can not play except by broadcasting them through listening installations. In this context the composition process means creating the sound material and then temporally organize it. So, the pieces are recorded on tapes (magnetic ones in the 50's and digital ones today) that are broadcasted to the audience during the concerts. In this situation, the possibilities for a musician to interpret a piece goes through spatialization on the listening system. Then this type of pieces does not benefit from the interpretation of the musicians like the traditional music does.

We looked for a system that allows composer to create interactive scores and musicians and performers to interpret them.

To define what is interpretation and the possible ways to deal with it, we base our research on the work of Jean Haury [5] who identifies four ways for interpretation:

- dynamic variations (the possibility to continuously modify the volume of the notes during the performance)
- accentuation (temporary volume variations)
- phrasing (modifying the binding of the notes)
- agogic modifications (the possibility to change the date of beginning and end of the notes)

In this study, we only focus on the agogic modifications. Thus we are not interested in the contents of the note and we will consider them as abstract processes that compute output streams (mostly audio streams in our case but not necessary).

Jean Haury also notices that the musicians can access the possibilities of interaction through control points placed in the piece and that he calls "interaction points". A pause is a good example of interaction point in the instrumental music since the musician or the conductor can choose the duration of a pause. We consider for the moment that these points are discrete and that they control triggering or releasing the notes. Precisely, we use the word *events* to mean the beginnings and end of the notes. An interaction point is associated with an event, making it interactive.

In addition, these possibilities of interaction that can modify the piece come with some limits imposed by the composer. For example, in instrumental music, these limits are set trough guidelines from the composer, for example for the volume (*p*,*ff*...) or the tempo (*accelerendo*...). Therefore the performer can benefit from liberties allowed by the composer while respecting limits set by the same person. Since we are only interested in the agogic modifications and then the possibility delaying or anticipating the trigger of some events of the piece and then modifying the beginning and duration of the notes, the composer must be able to define a temporal organization between the notes. Here we can see the coherence with static scores defined in *Boxes* and the possibility to temporally organize the musical structures thanks to the Allen relations.

As a consequence, we extend the use of the Allen relations from the assisted composition of static scores to the setting of a temporal organization in which the performer will express his interpretation of the score. Finally, our model present the following characteristics:

- a representation using hierarchical boxes
- the use of Allen relations
- introduction of interactions points at the beginning and end of the notes
- a generic encoding of the scores which allows one to use the same execution machine for every score and not an "ad hoc" system for each piece.

One thing that is important to remember is the relative independence of the different sides of the composition process: temporal organization, sound processing and interaction design. Therefore a composer can start by building the temporal meaning of his piece with abstract elements, continue by associating the structure with some signal processing computations and finally finish by binding some events with interaction points. Of course one can claim that a real composition process is a subtler mix of these three sides, but to some degree, we can see these steps as independent, and this independence allow us to imagine more general applications of our system.

We present in figure 3 a simple interactive score. In this example, a hierarchic box T_1 contains 4 simple boxes: T_2, T_3, T_4 and T_5, some of them constrained by Allen relations, the beginning of T_4 is interactive through the addition of an interaction point T.

It's important to note that the system we want to develop must support the composition stage and also the execution stage. That means that the same tool will be used by composers and performers with two different way of using it.

For the composition stage, we use similar processes as in *Boxes* by using a constraints satisfaction system to statically compute new values for the event dates each time the composer adds a relation or modifies a place of a box.

For the execution stage, we face a real-time situation and it appeared that a solution based on a constraints satisfaction system could lead to much too long computation times. Then, the computation of the date of an event could take so long that we will not be able to trigger it at the date it must occur. So we searched for a representation of the scores that can be executed by a generic machine without any computation in real-time to maintain the Allen relations. We chose a representation based on Petri nets which are specific structure that can manage concurrent processes that must synchronize at specific moments. In order to perform the execution stage, we turn the score into a Petri net which is run.

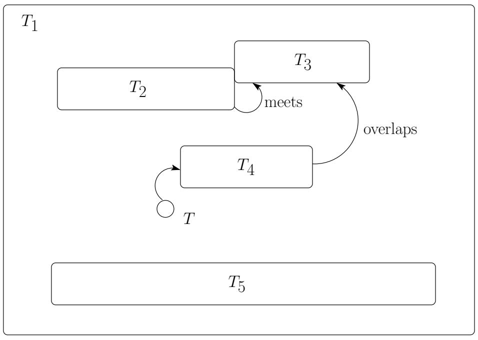


Figure 3: An interactive score example

During this stage, every *static* events (events that are not bound to an interactive point) are triggered automatically by the system while the system will wait for inputs from the performer to trigger the interactive events.

For more details on our theoretic model and our use of Petri nets, refer to [2].

3. IMPLEMENTATION

Two different implementations of our system exist. One takes place in the *Maquettes* environment of *OpenMusic* and the other is a stand-alone version. Here we present the *OpenMusic* version which is the more advanced one. The *Maquettes* environment is close to the *Boxes* one with a white sheet on which the composer can add some musical elements represented by boxes. The horizontal axis represents the time line while the vertical one has no specific meaning. Concerning the composition stage we then only added a constraints satisfaction system based on *Gecode* [6] to maintain the temporal constraints during the editing of the score.

Concerning the performance stage, as we are not interested in the contents of the boxes, we discharge the duties of signal processing to specific applications like Max/MSP or Pure Data. Therefore each box of a score is associated with a process of a third application that triggered when the box starts and is stopped when the box ends. We make *Iscore* communicating with other applications through the Open Sound Protocol (OSC[1]) as shown in the figure 4.

The OSC protocol allows us to send messages associated with specific functions in the receiving applications and also some numerical arguments to be passed to these functions. Each event of a score is then associated with an OSC message, a starting message for the beginnings of boxes and a stopping message for the ends. During the execution, when an event occurs, *Iscore* sends the associated message to the appropriated application. All real-time controls that could be made on the sound processes are done through the applications in charge of the sound processing.

---

[1] http://opensoundcontrol.org

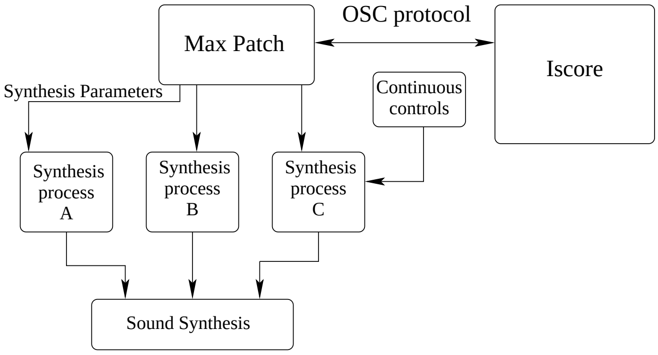


Figure 4: The architecture of the system

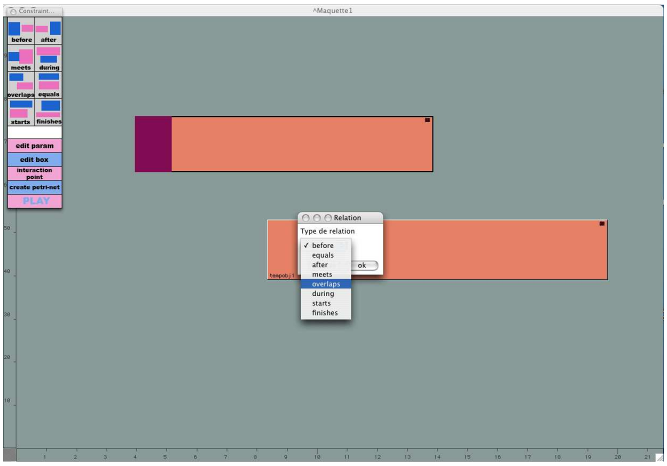


Figure 5: An example of using Iscore to compose an interacitve score

In figure 5, we reproduce a screen-shot of *Iscore* in the *Maquettes* during the edition of a very simple score composed of two boxes bound by a relation chosen by the composer.

3.1 An XML format

The intern backup format in Iscore is an XML format. For the moment it is quite succinct since it contains the strict information that we need to edit the scores. As an example we reproduce here the XML file of the score presented on figure 5.

```
<!DOCTYPE iscore SYSTEM "iscore.dtd">

<score>
 <title>
  Girl
 </title>
 <composer>
  Hooray Henrys
 </composer>
 <date>
  Today
 </date>
<applications>
 <application id="app1">
  <name>
   Pure Data
  </name>
  <ip>
   127.0.0.1
  </ip>
  <port>
   3000
  </port>
 </application>
</applications>

<boxes>
 <box id="bx1" name="tempobj1"
               id-application="app1"
               start-date="8349"
               duration="11333"
               posy="53"
               height="14">
  <start-message>
   /note-2/start
  </start-message>
  <end-message>
   /note-2/stop
  </end-message>
  <interaction-points>
  </interaction-points>
 </box>

 <box id="bx2" name="tempobj"
               id-application="app1"
               start-date="3944"
               duration="9930"
               posy="77"
               height="13">
  <start-message>
   /note-1/start
  </start-message>
  <end-message>
   /note-2/stop
  </end-message>
  <interaction-points>
   <interaction-point event="0">
    <control-message>
     /control
    </control-message>
   </interaction-point>
  </interaction-points>
 </box>
</boxes>

<allen-relations>
 <allen-relation type="overlaps">
  <motherbox>
   bx2
  </motherbox>
  <childbox>
   bx1
  </childbox>
 </allen-relation>
</allen-relations>
```

```
 <osc-receiving-port>
  1999
 </osc-receiving-port>
</score>
```

As we can see, all information concerning the boxes and the OSC messages associated with their events, the relations between them and the third applications which the score deals with, are simply encoded. But we would like to extend it by integrating information concerning the piece as part of a larger work for example (an opera for example) as we can find in the MusicXML[2] documents. Our aim is to define a generic format for encoding interactive pieces of non conventional music. This aim is far beyond our studies around *Iscore* because it supposes the ability to encode the processes used to synthesize the sounds and the way the performer will control the interactive points, but we assume that as far as the symbolic representation of the scores is concerned, our format could be an interesting base that must be improved.

4. MULTIMEDIA SCENARIOS

As we said before, the different steps of the composition (building the temporal structures and their organization, programming the signal processes and defining the interaction points) can be seen as independent. Particularly one can program signal processes that do not produce musical content but another types of contents and associate these processes with structures of a score build with *Iscore*. For example, one could write a performance using image synthesis or video processing with *Iscore*. Under these conditions, the word "score" is no longer adapted to the situation and it is more appropriate to talk about "interactive mutlimedia scenarios". We assume that the main approach of *Iscore* which is to emphasize on "writing the time" is still relevant in this more general context. A basic example could be a multimedia presentation involving different type of contents: text, video etc. One can build his presentation with empty structures, put content into these structures and then add some interaction points at strategical moment he thinks the run of his presentation could be modified during execution.

As a consequence of this possible application we contemplate to developing a system for exporting documents in the *Iscore* format into document in SMIL[3] format. This format has been defined to publish multimedia presentations with several interaction possibilities. Of course, this format cannot hold temporal constraints but it could use to broadcast limited simulations of work done with our system. *Iscore* could also be used as a sophisticated editor of SMIL documents.

Of course, we can also imagine using *Iscore* to edit and design much more complex interactive scenarios like video-games or virtual museum visits. We assume that the temporal aspects of such scenarios are very important and that *Iscore* could be a powerful tool to deal with them.

An important improvement of *Iscore* to correctly manage such a type of scenarios is to allow the definition of conditional cases. In fact for the moment, every events of a score that is written will occur during the execution. The performer will be able to delay the events but not choose between two events at a moment of the performance. Conditional cases which are used in some specific pieces of music are strongly required to write some type of scenarios. Since we think about it since the beginning of our study even if it is not yet implemented, conditional possibility can be integrated in our model.

Nevertheless, the introduction of the conditional cases rises up some formal questions about the model of execution. For example, let us suppose that the composer introduces some events that may not occur during the execution and that he synchronizes one of these conditional events with an other event which must occur during the execution. This situation can lead to a problem if the conditional event does not occur while the system is waiting for this event for triggering the one that must be synchronized with the conditional event. A first approach could be to simply prevent the composer from designing this type of situation, but maybe a more deep study could bring up a less restrictive solution.

An other question about the conditional case can be: what happens if several conditions are true at the same time. Note that we envisage the conditional cases as "exploring situations" in which the performer has to choose between several possibilities at a specific time. Since the performer cannot divide himself, he is not supposed to be able to play several conditional parts at the same time.

We think that an other interested challenge will be to design efficient interface to write scenarios with conditions.

4.1 The Virage Project

To experiment such extended applications of our system, we take part of the Virage [4] project. This project supported by the French association for research (ANR[5]), aims to develop an interactive tool for the stage managers of theater shows. Theater and more generally living arts use more and more multimedia contents (video, sound synthesis, effects...) that increases the artistic contents traditionally involved in shows (lights, music...). Therefore a lot of companies use computers to control all of these elements. But there is no relevant tool for writing the temporal organization of these elements that includes interactive possibilities. Interaction is needed to follow the performers actions during the representations. The project look into developing such a tool with two sides: a specific sequencer to write and execute the scenarios and some original interfaces to control this sequencer. We are deeply involved in the creation of the sequencer that will be a specific evolution of *Iscore*. This contribution will make us face specific ways of writing for other contents than music and then will extend our system to take into account the habits of the people of the theater community. For example, during the conception of a show, the person that will build the temporal structures and the one who will program the output process can be different. Typically, the first person will be the director while the second one will be a stage manager. Since we can see these two sides of the conception as independent, we can manage this way to do it.

---

[2] http://www.recordare.com/xml.html

[3] http://www.w3.org/AudioVideo/

[4] http://www.plateforme-virage.org/

[5] http://www.agence-nationale-recherche.fr/

Another side of the *Virage* project is a collaboration with a company that provide individual audio devices for museum visitors. In this situation the question of conditional cases is essential since the geographical approach will take an important place in the writing process (visitor is the room A, then in can step into room B or room C). As a consequence, we are thinking about more geographical interface or even spaciotemporal interface for the writing tool.

We present in figure 6, an example of such a spaciotemporal scenario designed for an imaginary hellenic museum. People are supposed to hold an individual audio device that broadcasts sounds and explanations according to what they are are looking to. The subfigure 6(a) shows what could be the geographical layer of the interface with which one can design the map of the building. We can see four rooms with specific content in each one. In the room A, three statues (Zeus, Hera and Pan) are presented. The designer of the exhibition imposes an order to the visitors for looking to the statues because he wants emphasize on the particular chronological order of the creation of the statues. After visiting the room A, the visitors can choose to step into two rooms. In the room B a movie is showed while in the room C some objects are presented on a table. In the room B, the audio devices will broadcast the audio content associated with the movie and in the room C, a general text about all the objects will be broadcasted. Finally, in the room C four statues are presented with no specific order such as the visitors can switch from one to an other in the order of their choice. Of course, the visitors can step from a room into an adjacent room of their choice since there is no specific order for the whole visit.

In the subfigure 6(b), we present the Petri net associated with the visit. A copy of this net will be run by each audio device used by a visitor. In this net, there is no temporal relations between the rooms but only "neighborhood relations". When a visitor is in the room A, he can step into room B or room C, after visiting one of these two rooms, he can step into room D or go back to room A. When he stands in room D, he can quit the museum or go back to room B or C. We suppose that some specific captors can detect when the visitor walk from a room to an other.

In the subfigure 6(c), we present the temporal scenario associated with the room A. Since the designer wants to write a specific order for the presentation of the statues, he can write a specific scenario for the sounds and texts associated with the statue. The designer can write this order using "*before*" relations between the texts associated with each statue. In addition, he put a sound ambiance that must start as soon as the visitor steps into the room and lasts until he steps out of it. Finally, the designer wants to trigger a sound of transition between the first two statues. This sound will be triggered at some specific moment of the first text and will end at some specific moment of the second one. We suppose that some specific captors can detect when the visitor enters and leaves the aera of a statue. As we can see, the scenario is very close to a musical score and therefore, we can turn it into a Petri net by using our algorithm.

In the subfigure 6(d), we present the Petri net associated with the room D. Since in this room there is no relevant temporal organization over the visit of this room, the Petri net of this room is geographical such as the one associated to the whole museum. As we can see, the place in the middle of the figure represents the state in which the system is when the visitor stands in the room. From this state, he can go watching a statue or another and switching from one to an other. As in the room A, we suppose that some specific captors detect the entering and the leaving of the aera of a statue. One can claim that for this type of geographical net, we could use automata (this remark is also relevant for the net of the museum), but we want to preserve the homogeneity of the model. Since automata can be simulated by Petri nets, we prefer to use Petri nets for all situations.

Finally, this example shows how we can extend our system for designing spatiotemporal scenarios. The particularity of the "real-life" scenarios such as the museum example is that we cannot physically constraint the moving of the visitors and as a consequence, we have to allow any type of moving between the pieces and the different aeras of the pieces. In this situation the designer cannot totally benefit from mixing temporal constraints with the geography. Of course, we can imagine that some people who work for the museum will insure specific moving of the visitors or that some locking systems will automatically block the access to a room but you cannot instantaneously eject a visitor from a room after a delay has passed for example. More complex mixes between temporal and geographical constraints can be designed for virtual scenarios such as video-games.

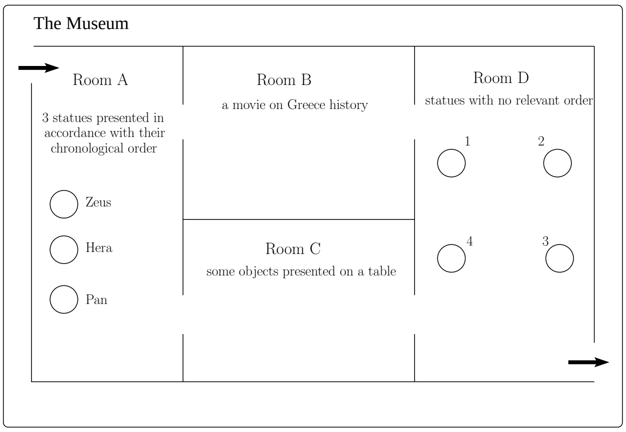
5. ADAPTIVE APPLICATIONS

In the previous section, we presented the benefits we can take from the independence between the temporal structures of a score or a scenario and the processes associated to them. Now, we will exhibit the benefit we can take from the independence between the temporal structures and the interactive points that we can bind to them.

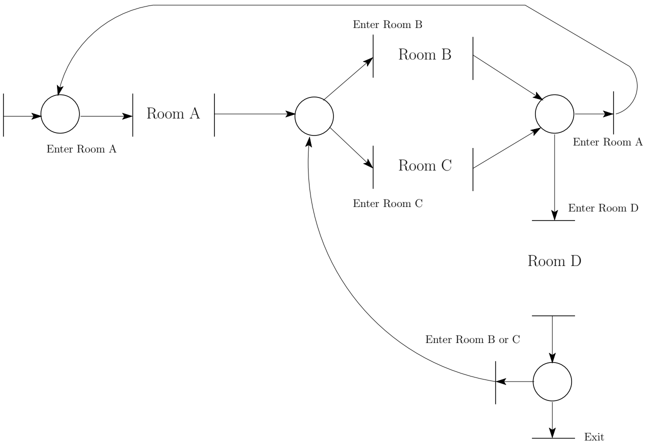
After a score is composed, one can modify the interaction points by adding or removing some of them without modifying the temporal structure of the piece. Thus one will preserve the content and the meaning of the piece but change the way of interacting with it. An interesting application of this is to adapt a piece to the abilities of a musician or a performer. The more skills he has, the more interaction points he will add to control a lot of events. An example could be adapting a piece for a virtuoso or a beginner. There are some pedagogical interests in proposing to a beginner to perform a "limited" version of a piece. He will be able to feel the experience of interpreting the piece without being able to directly control every event of it. Then he can also progressively increase the number of interaction points during his learning process. Therefore a musician can discharge the duties of the control of events to the system by turning these events from an interactive status to a static one.

We can also adapt pieces to be played by handicapped performers with limited skills. In the same way as for a beginner, the system can behaves as a help to perform pieces.

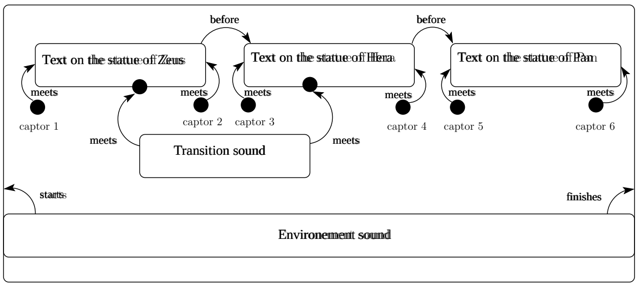
Another dimension of this possibility of modifying the way of interacting with a score is changing the number of performers. Adding and removing interaction points allows us


The Museum
Room A
3 statues presented in accordance with their chronological order
Zeus
Hera
Pan
Room B
a movie on Greece history
Room C
some objects presented on a table
Room D
statues with no relevant order
1
2
4
3

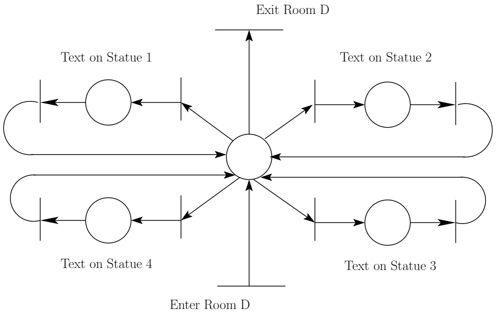
(a) The map of the museum



(b) The Petri net associated with the museum



(c) The multimedia scenario of the room A



(d) The Petri net associated with the room D

Figure 6: An example of a museographic scenario

to increase or decrease the number of performers who can simultaneously interact with the score. These two ways of modification present interesting possibilities. For example, a performer involved in a band or an orchestra will be able to rehearse alone in removing all of the interaction points concerning other performers. In this situation the system will behave as a play back system by following his actions and choices for triggering the events normally controlled by the other members of the band. On the other hand, increasing the number of performers can lead to interesting situation by turning the representation into a "contest game" for controlling musical contents in limited quantity.

To allow these types of applications on existing musical pieces and not only on specifically designed pieces, we are developing a way of importing traditional pieces into *Iscore*. This will be achieve thanks to a transformation system from MusicXML and MIDI format into the *Iscore* XML format.

These considerations on changing the way of interaction with a score can of course be applied with multimedia contents and could lead to multimedia scenarios with an undefined numbers of performers or narrators. In a video-game context, this can allow the modification of the number of players.

6. FUTURE WORK

One way for improving *Iscore* is allow the composer to refine the Allen relation with more precise constraints. Concretely, we want the composer to be able to define a specific propriety for any interval between two events of the score. Three status are possible:

- rigid: the duration of the interval that is written on the score must be maintained during the performance
- semi-rigid: the value of the duration must stay in $[Val_{min}, Val_{max}]$
- supple: the value of the duration can be anything in $[0, \infty]$

Of course these constraints are relevant in case of the presence of interaction points which can modify their duration during the execution. It appeared that such quantitative constraints cannot be easily included in the model using only Petri nets so we are working on adding a limited constraints satisfaction system to hold these specific constraints.

An other way of improvement is to deal with continuous controls and not only discrete one.

7. CONCLUSION

We have presented a system for composing interactive scores of musical pieces based on temporal constraints. This system has two sides: a composition side and an execution side. For both sides several applications can be imagine like changing the contents of the scores to write interactive multimedia scenarios or modifying the way of interaction with a score to adapt it to a number of performers and to the abilities of these performers.

8. REFERENCES

[1] J. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[2] A. Allombert, G. Assayag, and M. Dessainte-Catherine. A system of interactive scores based on petri nets. In *Pr. of the 4th Sound and Music Computing Conference (2007), Lefkada, Greece*, July 2007.

[3] G. Assayag, C. Rueda, M. Laurson, C. Agon, and O. Delerue. Computer assisted composition at ircam : From patchwork to openmusic. *Computer Music Journal*, 23(3), 1999.

[4] A. Beurivé. Un logiciel de composition musicale combinant un modèle spectral, des structures hiérarchiques et des contraintes. In *Journées d'Informatique Musicale, JIM 2000*, 2000.

[5] J. Haury. La grammaire de l'exécution musicale au clavier et le mouvement des touches. In *Manuscrit*.

[6] C. Schulte and G. Tack. Views and iterators for generic constraint implementations. In *Pr. of the Fifth International Colloqium on Implementation of Constraint and Logic Programming Systems, CICLOPS05*, 2005.