Augmented Music Scores

D. Fober, C. Daudin, Y. Orlarey, S. Letz

Grame
Centre national de création musicale
Lyon - France

April 2010

GRAME CENTRE NATIONAL DE CREATION MUSICALE

Sommaire

The Interlude Project.

New digital paradigms for the expressive gestural exploration and interaction with music contents.

Application domains:

- professional (pedagogy, interactive music...)
- general public (musical games...)

Partners:

- Ircam, Grame
- VoxLer, Dafact
- NoDesign, Atelier les Feuillantines

The Interlude Project.

New digital paradigms for the expressive gestural exploration and interaction with music contents.

Application domains:

- professional (pedagogy, interactive music...)
- general public (musical games...)

Partners:

- Ircam, Grame
- VoxLer, Dafact
- NoDesign, Atelier les Feuillantines

The Interlude Project.

New digital paradigms for the expressive gestural exploration and interaction with music contents.

Application domains:

- professional (pedagogy, interactive music...)
- general public (musical games...)

Partners:

- Ircam, Grame
- VoxLer, Dafact
- NoDesign, Atelier les Feuillantines

Interaction with symbolic content.

Augmented Music Score

- An *augmented music score* is a score that connects a symbolic music object to different representations of its performance.
- The music score is to be taken in a broad sense, as a graphic object representing a temporal object.
- The performance corresponds to a specific sound or gesture instance of the score.

Interaction with symbolic content.

Augmented Music Score

- An *augmented music score* is a score that connects a symbolic music object to different representations of its performance.
- The music score is to be taken in a broad sense, as a graphic object representing a temporal object.
- The performance corresponds to a specific sound or gesture instance of the score.

Interaction with symbolic content.

Augmented Music Score

- An *augmented music score* is a score that connects a symbolic music object to different representations of its performance.
- The music score is to be taken in a broad sense, as a graphic object representing a temporal object.
- The performance corresponds to a specific sound or gesture instance of the score.

Interaction with symbolic content.

Augmented Music Score

- An *augmented music score* is a score that connects a symbolic music object to different representations of its performance.
- The music score is to be taken in a broad sense, as a graphic object representing a temporal object.
- The performance corresponds to a specific sound or gesture instance of the score.

Problematics

The core of the augmented music score

- score extension to arbitrary music objects
- expression of relations between graphic and time spaces
- performance representation (gestural, sound)

Sommaire

First class music objects

All the score components:

- have a graphic dimension,
- have a time dimension,
- can be addressed both in the graphic and time domains,
- maintain relations between time and graphic space,
- can be synchronized in the time and graphic space.

Components

Graphic resources typology.

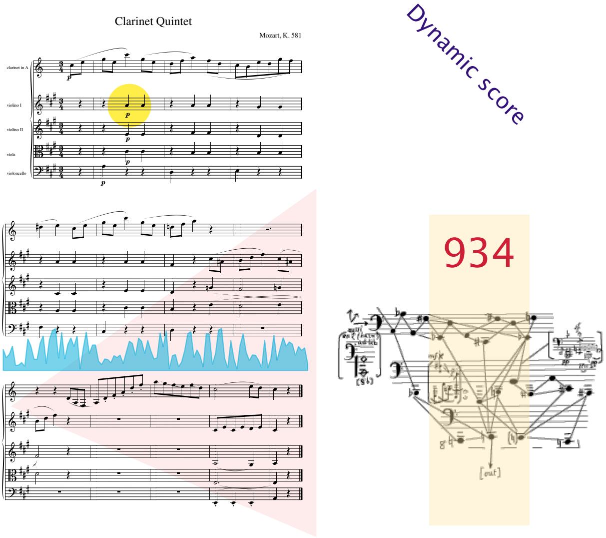
- Music scores
  GMN (Guido Music Notation format) or MusicXML format
- Textual elements
- Graphic bitmaps (jpg, gif, tiff, png, ...)
- Vectorial graphic (rectangles, ellipses, ...)
- Sound and gesture graphic representations

Available parameters

Common parameters

- position (x, y, z)
- scale
- rotation
- color
- date
- duration
- visibility

Example


Clarinet Quintet
Mozart, K. 581
clarinet in A
violino I
violino II
viola
violoncello
Dynamic score
934
[out]

Implementation

- as a C++ shared library.
- as an application: an augmented score viewer.
- multi-platform [MacOS X, Linux, Windows].
- based on the Qt framework.
- based on the Guido engine and the libMusicXML library.
- supports the OSC protocol [oscpack].

Sommaire

Time segments

- A *time segment* is defined as an interval $i = [t_0, t_1[$ such as $t_0 \leqslant t_1$.
- $i = [t_0, t_1[$ is said empty when $t_0 = t_1$.
  We will use $\oslash$ to denote empty intervals.
- Time segments intersection is the largest interval such as:

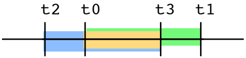
$$\forall i_m,\ \forall i_n,\ i_m \cap i_n := \{j \mid j \in i_m \ \wedge\ j \in i_n\}$$

Time segments

- A *time segment* is defined as an interval $i = [t_0, t_1[$ such as $t_0 \leqslant t_1$.
- $i = [t_0, t_1[$ is said empty when $t_0 = t_1$.
  We will use $\oslash$ to denote empty intervals.
- Time segments intersection is the largest interval such as:

$$\forall i_m,\ \forall i_n,\ i_m \cap i_n := \{ j \mid j \in i_m \ \wedge \ j \in i_n \}$$

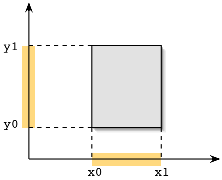
Time segments



- A *time segment* is defined as an interval $i = [t_0, t_1[$ such as $t_0 \leqslant t_1$.
- $i = [t_0, t_1[$ is said empty when $t_0 = t_1$.
  We will use $\oslash$ to denote empty intervals.
- Time segments intersection is the largest interval such as:

$$\forall i_m,\ \forall i_n,\ i_m \cap i_n := \{j \mid j \in i_m \ \wedge\ j \in i_n\}$$

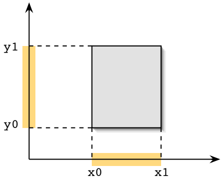
Graphic segments



- A *graphic segment g* is defined as a rectangle given by two intervals $g = (x, y)$ where x is an interval on the x-axis and y, on the y-axis.
- $g = \{x, y\}$ is said empty when $x = \emptyset$ or $y = \emptyset$
- Intersection $\cap$ between graphic segments:

$$\forall g_m = \{x_m, y_m\},\ \forall g_n = \{x_n, y_n\},\ g_m \cap g_n = \{x_m \cap x_n, y_m \cap y_n\}$$

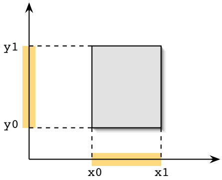
Graphic segments

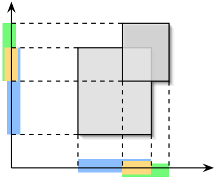


- A *graphic segment g* is defined as a rectangle given by two intervals $g = (x, y)$ where x is an interval on the x-axis and y, on the y-axis.
- $g = \{x, y\}$ is said empty when $x = \emptyset$ or $y = \emptyset$
- Intersection $\cap$ between graphic segments:

$$\forall g_m = \{x_m, y_m\},\ \forall g_n = \{x_n, y_n\},\ g_m \cap g_n = \{x_m \cap x_n, y_m \cap y_n\}$$

Graphic segments





- A *graphic segment g* is defined as a rectangle given by two intervals $g = (x, y)$ where x is an interval on the x-axis and y, on the y-axis.
- $g = \{x, y\}$ is said empty when $x = \emptyset$ or $y = \emptyset$
- Intersection $\cap$ between graphic segments:

$$\forall g_m = \{x_m, y_m\},\ \forall g_n = \{x_n, y_n\},\ g_m \cap g_n = \{x_m \cap x_n, y_m \cap y_n\}$$

Segment generalization

- A n-dimensional segment is defined as a set of n intervals $s^n = \{i_1, ..., i_n\}$ where i_j is an interval on the dimension j.
- A segment s^n is said empty when $\exists i \in s^n \mid i = \emptyset$
- Intersection between segments is defined as the set of their intervals intersection:

$$s_1^n \cap s_2^n = (i_1 \cap j_1, ..., i_n \cap j_n)$$

where $s_1^n = (i_1, ..., i_n)$ et $s_2^n = (j_1, ..., j_n)$

Segment generalization

- A n-dimensional segment is defined as a set of n intervals $s^n = \{i_1, ..., i_n\}$ where i_j is an interval on the dimension j.
- A segment s^n is said empty when $\exists i \in s^n \mid i = \emptyset$
- Intersection between segments is defined as the set of their intervals intersection:

$$s_1^n \cap s_2^n = (i_1 \cap j_1, ..., i_n \cap j_n)$$

where $s_1^n = (i_1, ..., i_n)$ et $s_2^n = (j_1, ..., j_n)$

Segment generalization

- A n-dimensional segment is defined as a set of n intervals $s^n = \{i_1, ..., i_n\}$ where i_j is an interval on the dimension j.
- A segment s^n is said empty when $\exists i \in s^n \mid i = \emptyset$
- Intersection between segments is defined as the set of their intervals intersection:

$$s_1^n \cap s_2^n = (i_1 \cap j_1, ..., i_n \cap j_n)$$

where $s_1^n = (i_1, ..., i_n)$ et $s_2^n = (j_1, ..., j_n)$

Segmentations

- A n dimensions resource R is *segment-able* when it can be defined by a segment S^n of dimension n.
- The segmentation of a resource R is the set of segments $Seg(R) = \{s_1^n, ...s_i^n\}$ such as:

$$\begin{array}{lll} \forall i,j \in Seg(R) & i \cap j = \emptyset & \text{segments are disjoints} \\ \forall i \in Seg(R) & i \cap S^n = i & \text{all segments are included in R} \end{array}$$

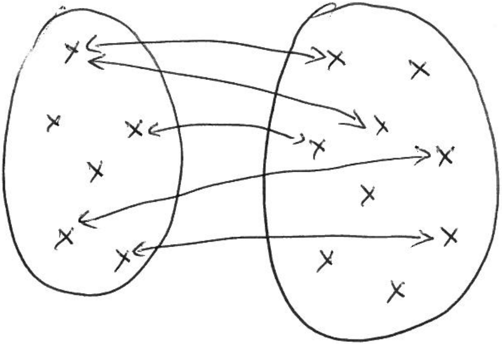
Segmentations

- A n dimensions resource R is *segment-able* when it can be defined by a segment S^n of dimension n.
- The segmentation of a resource R is the set of segments $Seg(R) = \{s_1^n, ...s_i^n\}$ such as:

$$\forall i, j \in Seg(R) \quad i \cap j = \emptyset \qquad \text{segments are disjoints}$$
$$\forall i \in Seg(R) \quad i \cap S^n = i \qquad \text{all segments are included in R}$$

Mapping (1)

A *mapping* is a relation between 2 segmentations.



Mapping (2)

- For a mapping $M \subseteq Seg(R_1) \times Seg(R_2)$ the function:

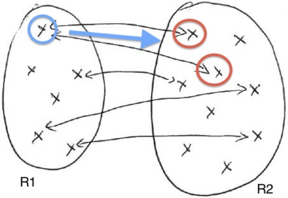
$$M^+(i) = \{i' \in Seg(R_2) \mid (i, i') \in M\}$$

gives the set of segments from R_2 associated to the segment i from R_1.

- and the reverse function:

$$M^-(i') = \{i \in Seg(R_1) \mid (i, i') \in M\}$$

gives the set of segments from R_1 associated to the segment i' from R_2.



Mapping (2)

- For a mapping $M \subseteq Seg(R_1) \times Seg(R_2)$ the function:

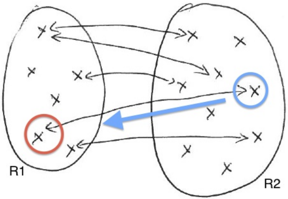
$$M^{+}(i) = \{i' \in Seg(R_2) \mid (i, i') \in M\}$$

gives the set of segments from R_2 associated to the segment i from R_1.

- and the reverse function:

$$M^{-}(i') = \{i \in Seg(R_1) \mid (i, i') \in M\}$$

gives the set of segments from R_1 associated to the segment i' from R_2.



Mapping (3)

- These functions are defined for a set of segments as the union of each segment mapping:

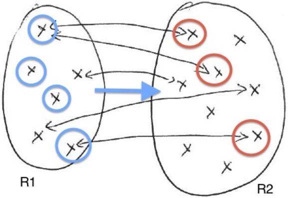
$$M^+(\{i_1, ...i_n\}) = M^+(i_1) \cup M^+(i_2)... \cup M^+(i_n)$$

- Mappings composition:
let $M_1 \subseteq Seg(R_1) \times Seg(R_2)$
and $M_2 \subseteq Seg(R_2) \times Seg(R_3)$

$$(M_1 \circ M_2)^+(i) = M_2^+(M_1^+(i))$$

- i.e. the relation:

$$M_1 \circ M_2 \subseteq Seg(R_1) \times Seg(R_3)$$



Mapping (3)

- These functions are defined for a set of segments as the union of each segment mapping:

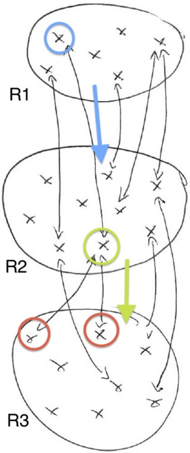
$$M^{+}(\{i_1, ...i_n\}) = M^{+}(i_1) \cup M^{+}(i_2) ... \cup M^{+}(i_n)$$

- Mappings composition:
let $M_1 \subseteq Seg(R_1) \times Seg(R_2)$
and $M_2 \subseteq Seg(R_2) \times Seg(R_3)$

$$(M_1 \circ M_2)^{+}(i) = M_2^{+}(M_1^{+}(i))$$

- i.e. the relation:

$$M_1 \circ M_2 \subseteq Seg(R_1) \times Seg(R_3)$$



Mapping (3)

- These functions are defined for a set of segments as the union of each segment mapping:

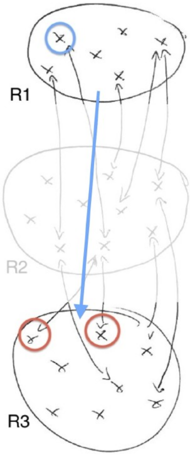
$$M^{+}(\{i_1, ...i_n\}) = M^{+}(i_1) \cup M^{+}(i_2) ... \cup M^{+}(i_n)$$

- Mappings composition:
  let $M_1 \subseteq Seg(R_1) \times Seg(R_2)$
  and $M_2 \subseteq Seg(R_2) \times Seg(R_3)$

$$(M_1 \circ M_2)^{+}(i) = M_2^{+}(M_1^{+}(i))$$

- i.e. the relation:

$$M_1 \circ M_2 \subseteq Seg(R_1) \times Seg(R_3)$$



Relations between graphic and time spaces.

Segmentations and mappings for each component type.

type	segmentations and mappings required
text	*graphic* ↔ **text** ↔ **relative time**
score	*graphic*↔ *wrapped relative time* ↔ *relative time*
image	*graphic* ↔ **pixel** ↔ **relative time**
gr. vectorial	**vectorial**↔ **relative time**
signal	*graphic* ↔ **frame** ↔ **relative time**

Demo

See:

- Max/sync/sync.maxpat
- PureData/sync/sync.pd
- python/example.py
- lisp/example.lisp

INScoreViewer must be running.

Sommaire

The problem...

Previous approach:

Signal(s) (f0, rms...) →
Representation type →
graphic computation

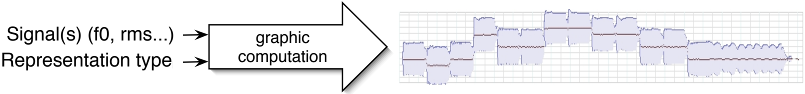
- static signal representation
- non-extensible dynamically

Currently...

- a more general system, covering a large set of representations
- dynamically extensible
- and easy to use...

The problem...

Previous approach:



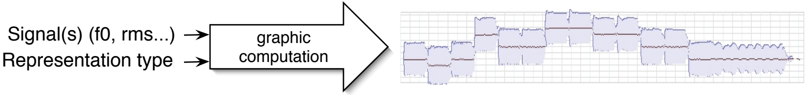
- static signal representation
- non-extensible dynamically

Currently...

- a more general system, covering a large set of representations
- dynamically extensible
- and easy to use...

The problem...

Previous approach:



- static signal representation
- non-extensible dynamically

Currently...

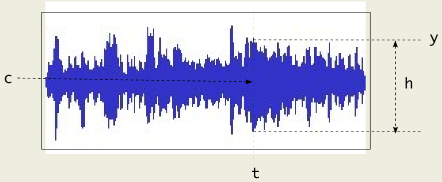
- a more general system, covering a large set of representations
- dynamically extensible
- and easy to use...

Graphic signals

The *graphic of a signal* as a *graphic signal*:

A composite signal made of:

- a *y* signal.
- a thickness signal.
- a color signal.



Graphic signals

Consider a signal S defined as a time function:

$$f(t) : \mathbb{R} \rightarrow \mathbb{R}^3 = (y, h, c) \mid y, h, c \in \mathbb{R}$$

this signal could be directly drawn.
(i.e. without additional computation)

To make simple, we assume that the color space addressed by c has one dimension.

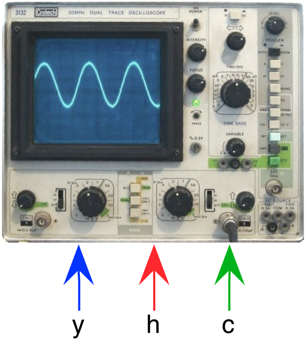
Graphic signals

Consider a signal S defined as a time function:

$$f(t) : \mathbb{R} \rightarrow \mathbb{R}^3 = (y, h, c) \mid y, h, c \in \mathbb{R}$$

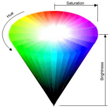
this signal could be directly drawn.
(i.e. without additional computation)

To make simple, we assume that the color space addressed by c has one dimension.



Parallel signals types

- Color signal type:



(HSBA model [hue, saturation, brigthness, transparency])

$$c ::= \overrightarrow{(h, s, b, a)} \mid h, s, b, a \in \mathbb{R}$$

- Graphic signal type:

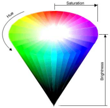
$$g ::= \overrightarrow{(y, th, h, s, b, a)} \mid y, th, h, s, b, a \in \mathbb{R}$$

- Parallel graphic signals type

$$g^n ::= \overrightarrow{g} \mid g \in \mathbb{R}^6$$

Parallel signals types

- Color signal type:



(HSBA model [hue, saturation, brigthness, transparency])

$$c ::= \overrightarrow{(h, s, b, a)} \mid h, s, b, a \in \mathbb{R}$$

- Graphic signal type:

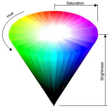
$$g ::= \overrightarrow{(y, th, h, s, b, a)} \mid y, th, h, s, b, a \in \mathbb{R}$$

- Parallel graphic signals type

$$g^n ::= \overrightarrow{g} \mid g \in \mathbb{R}^6$$

Parallel signals types

- Color signal type:



(HSBA model [hue, saturation, brigthness, transparency])

$$c ::= \overrightarrow{(h, s, b, a)} \mid h, s, b, a \in \mathbb{R}$$

- Graphic signal type:

$$g ::= \overrightarrow{(y, th, h, s, b, a)} \mid y, th, h, s, b, a \in \mathbb{R}$$

- Parallel graphic signals type

$$g^n ::= \overrightarrow{g} \mid g \in \mathbb{R}^6$$

Signals parallelization

Let $\mathbb{S}$, the set of signals $s : \mathbb{N} \rightarrow \mathbb{R}$.
We define a *parallel* operation '/' as:

$$s_1 / s_2 / ... / s_n : \mathbb{S} \rightarrow \mathbb{S}^n \mid s_i \in \mathbb{S}$$

Time function of a parallel signal $s^n \in \mathbb{S}^n : \mathbb{N} \rightarrow \mathbb{R}^n$

$$f(t) = (f_0(t), f_1(t), ...f_n(t)) \mid f_i(t) : \mathbb{N} \rightarrow \mathbb{R}$$

Signals parallelization

Let $\mathbb{S}$, the set of signals $s : \mathbb{N} \rightarrow \mathbb{R}$.
We define a *parallel* operation '/' as:

$$s_1/s_2/.../s_n : \mathbb{S} \rightarrow \mathbb{S}^n \mid s_i \in \mathbb{S}$$

Time function of a parallel signal $s^n \in \mathbb{S}^n : \mathbb{N} \rightarrow \mathbb{R}^n$

$$f(t) = (f_0(t), f_1(t), ...f_n(t)) \mid f_i(t) : \mathbb{N} \rightarrow \mathbb{R}$$

Examples

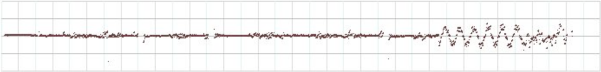
$$g = S_{f0} \ / \ k_t \ / \ k_c$$

S_{f0} : fundamental frequency

k_t : constant thickness signal

k_c : constant color signal

Examples



$$g = S_{f0} - S_{fr} \; / \; k_t \; / \; k_c$$

S_{f0} : fundamental frequency

S_{fr} : reference frequency

k_t : constant thickness signal

k_c : constant color signal

Examples

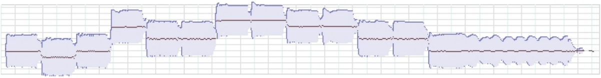
$$g = k_y \;/\; S_{rms} \;/\; k_c$$

S_{rms} : RMS signal

k_y : constant y signal

k_c : constant color signal

Examples



$$g = S_{f0} \ / \ S_{rms} \ / \ k_c$$

S_{rms} : RMS signal

S_{f0} : fundamental frequency

k_c : constant color signal

Examples

$g0 = S_{f0} \;/\; S_{rms0} \;/\; k_c0$

S_{f0} : fundamental frequency

S_{rms0} : f0 RMS values

$g1 = S_{f0} \;/\; S_{rms1} + S_{rms0} \;/\; k_c1$

S_{rms1} : f1 RMS values

$g2 = S_{f0} / \; S_{rms2} + S_{rms1} + S_{rms0} \;/\; k_c2$

S_{rms2} : f2 RMS values

...

$g = g2 \;/\; g1 \;/\; g0$

Demo

See:

- Max/sinus/sinus.maxpat
- PureData/sinus/sinus.pd
- Max/siggraph/siggraph.maxpat
- PureData/siggraph/siggraph.pd

InterludeScoreViewer must be running.

INScore

Interactive Augmented Scores

http://inscore.sourceforge.net/

open source