Antescofo: a score follower

Strategies for synchronization and error handling during automatic accompaniment

José Echeveste - Arshia Cont
Jean-Louis Giavitto - Florent Jacquemard
Introduction

• **Real-time synchronisation** of electronic accompaniment with a musician’s playing

• A **language** is used to compose the **interaction** between a musician and a computer

• Supports **specific time** management and **error** handling

• Formal methods are used **analyze** a score and verify **properties**
Automatic accompaniment adapted to a musician’s interpretation

- http://www.youtube.com/watch?v=iN9MmiSI Bj0
What is score following
Interactive music

Automatic improvisation (FSM)

Sound Processing (CC, Calc)

Sound synthesis (Calc, CC)

Gesture Processing (CC)

Space (Calc, CC)

Physical Models (CC, Calc)

And More!

CC: continuous control
FSM: finite state machine
Calc: calculation
Antescofo Modules

- Recognition algorithm
  - signal to symbol

Results:
- Position
- Tempo

- Interaction specification
- **Generation** of electronic actions:
  - Composition / programming
  - Adaptation to **non-determinism** due to the interpretation
  - Error handling

Information theory

Reactive System
<table>
<thead>
<tr>
<th>Composing musical interaction</th>
<th>Synchronous paradigm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td>Program</td>
</tr>
<tr>
<td>Instrumental Score</td>
<td>Expected input</td>
</tr>
<tr>
<td>Musician’s playing</td>
<td>Real input</td>
</tr>
<tr>
<td>Accompaniment score</td>
<td>Expected output</td>
</tr>
<tr>
<td>Generated accompaniment</td>
<td>Real output</td>
</tr>
</tbody>
</table>
A language to specify the interaction between the Musician and the Machine

- Recognizable **events**: NOTE, CHORD, TRILL
- **Actions** to execute: elementary actions (messages), **Groups**, Loops
- Durations and delays in **absolute** or **relative** time (/ tempo)
Interactive score example

energico, affilato
\( \frac{3}{4} \)  

Violin

Instrumental score

Electronic score
(discrete actions)

Electronic Score
(continue actions)
Adapt to the interpretation

Current position
What note is played?

Tempo
How fast are notes played?

Errors
What note is omitted?
Synchronization strategies

1. Group or loop **loose**: synchronize with the **estimated tempo**
2. Group or loop **tight**: synchronize with the **events**

---

**Diagram Description**

- **Absolut Times (seconds)**
- **Events played by the musician**
- **Beats of the estimated tempo**

- **Group loose**
  - **action-loose 1**
  - **action-loose 2**
  - **action-loose 3**
  - **action-loose 4**

- **Group tight**
  - **action-tight 1**
  - **action-tight 2**
  - **action-tight 3**
  - **action-tight 4**
Handling errors

- An error is located when the system recognizes an event without recognizing the previous event(s).

- An action associated to an unrecognized event:
  1. is not executed if it is local
  2. is executed if it is global.

Synchronization and error handling strategies are orthogonal.
Demo
Formal Semantics of Antescofo

• **Reactive** system
• **Distributed** system
• **Structured** system: hierarchical groups et loops
• **Timed** system (relative time)

A **formalization** for:
  • Defining trace semantics
  • Analyzing possible executions
  • Verifying properties
Abstract syntax of the augmented score

\[
\begin{align*}
\text{score} & : = \varepsilon \mid \text{event score} \mid (d \ \text{group}) \ \text{score} \mid (d \ \text{loop}) \ \text{score} \\
\text{event} & : = e_i \\
\text{group} & : = \text{Group synchro error} \ (d \ \text{action})^+ \\
\text{loop} & : = \text{Loop synchro error} \ p \ (d \ \text{action})^+ \\
\text{action} & : = a \mid \text{group} \mid \text{loop}
\end{align*}
\]

\[
\begin{align*}
\text{synchro} & = \text{loose} \mid \text{tight} \\
\text{error} & = \text{global} \mid \text{local}
\end{align*}
\]
Event and action dating

Set of events and actions symbolically dated

Ordered set ideal case

Ordered set real case

events <t'_i e_i>

actions <t'_a a>

events <t''_i e_i>

actions <t''_a a>

evt_evali

act_evali

act_evalr

datation

date

SCORE
goup loop
e_i a_i
Symbolic dating

Set of events and actions symbolically dated

events $<t_i, e_i>$

actions $<t_j a_j da_j E_j>$

datation
date

goup loop
$e_i$
$a_i$

SCORE
Symbolic dating

\[\text{datation}[\text{score}] = \text{datation}_0[\text{score}]\]
\[\text{datation}_i[\epsilon] = \emptyset\]
\[\text{datation}_i[e_i \text{ score}] = \{< t_i, e_i >\} \cup \text{datation}_i[\text{score}]\]
\[\text{datation}_i[(d \text{ group}) \text{ score}] = \text{date}_{(t, \phi)}[d \text{ group}] \cup \text{datation}_i[\text{score}]\]
\[\text{datation}_i[(d \text{ loop}) \text{ score}] = \text{date}_{(t, \phi)}[d \text{ loop}] \cup \text{datation}_i[\text{score}]\]
Symbolic dating  
(example for a tight Group)

\[
date_{(t_i,E)}[d_G \text{ Group tight error}_G (d_1 \text{ action}_1)...(d_k \text{ action}_k)] = \\
\begin{cases}
= date_{(t_i,E+<e_i,\text{error}_G,0>)}[d' \text{ action}_1] \\
\bigcup date_{(t_i,E)}[0 \text{ Group tight error}_G (d_2 \text{ action}_2)...(d_k \text{ action}_k)]  & \text{si } (t_i + d_G + d_1) < t_{i+1} \\
= date_{(t_{i+1},E)}[0 \text{ Group tight error}_G (d_1 \text{ action}_1)...(d_k \text{ action}_k)]  & \text{sinon}
\end{cases}
\]

avec \( d' = d_G + d_1 \)

et \( d'' = d_G + d_1 + d_2 \)

et \( d''' = (t_{i+1} - t_i) - d_1 - d_G \)
Absolute dating
(ideal execution)
Absolute dating of an elementary action  
(ideal execution)

act_evali(<t_i,a,d,Error>) =

\[
\begin{cases} 
\text{evt_evali}(t_i) + d \times T_i & \text{if } si(t_i + d) < t_{i+1} \\
act_avali(<t_{i+1},a,(d-(t_{i+1} - t_i)),Error>) & \text{otherwise}
\end{cases}
\]
Absolute dating
(real execution)

Set of events and actions symbolically dated

- events $<t_i, e_i>$
- actions $<t_j a_j da_j E_j>$

Ordered set real case

- events $<t''_i, e_i>$
- actions $<t'_a a>$

Act_evalr

$<t'_j \text{Tempo}>'$
Absolute dating of an elementary action 
(real execution)

\[
\text{act\_evalr}(< t, a, d_a, Error >) =
\begin{cases}
\bot & \text{si } \exists < e, \text{local}, d > \in Error \text{ et } e \in \varepsilon_{Error} \\
\text{time\_evalr}(t_x, d) \text{ avec } t < t_x \text{ sinon si } \exists < e, \text{global}, d > \in Error \text{ et } e \in \varepsilon_{Error} \\
\text{time\_evalr}(t, d_a) & \text{sinon}
\end{cases}
\]

avec \( \text{time\_evalr}(t_i, d) = \)
\[
\begin{cases}
t'_i + d \times T'_i & \text{si } t'_i + d \times T'_i < t'_j \text{ avec } t'_i < t'_j \\
\text{time\_evalr}(t_j, d - \frac{t'_j-t'_i}{T'_i}) & \text{sinon}
\end{cases}
\]
Conclusion

• **Antescofo**: Language adopted by the composers
  – Writing of time
  – Writing of interaction

• **Adaptation to a musician’s interpretation**
  – Tempo (relative time)
  – Errors

• **Perspectives**
  – Augment the score language (variables)
  – Analyze the score
  – Use formal validation techniques to verify certain properties
  – Go to open scores
  – Graphical syntax