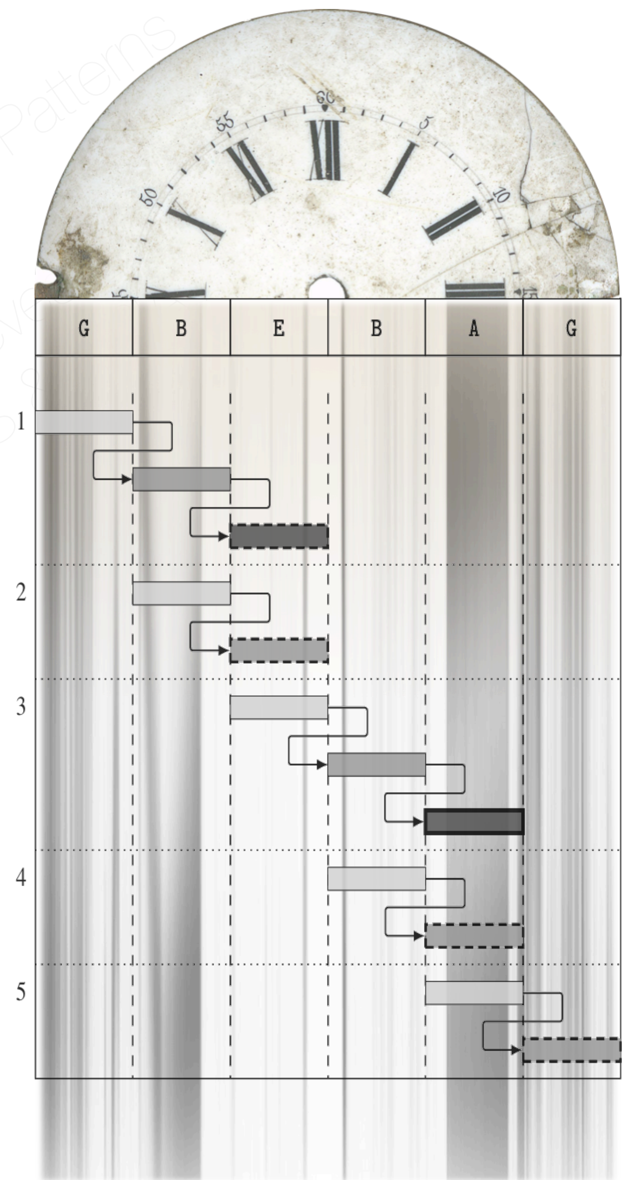
PPDP, Canterbury September 2014

Real-Time Matching of Antescofo Temporal Pattern

Jean-Louis Giavitto
José Echeveste

ircam Centre Pompidou · cnrs · inria · UPMC PARIS UNIVERSITAS



Real-Time Matching of Antescofo Temporal Patterns PPDP 8 September 2014 Jean-Louis Giavitto & José Echeveste IRCAM, UPMC, CNRS UMR STMS 9912

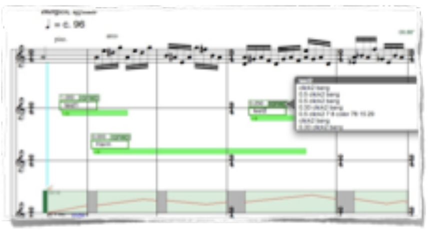
Outline

1. The application domain
   score following and mixed music
2. The augmented score: a domain specific language
   time as a first class entity of the language,
   not a side-effect of the computation nor a resource
3. Real-time temporal patterns
   extended subset on RE on sequences *in time* (not memory!)
4. Sketch of their semantics
   beware of causality
5. Implementation by translation
   the efficient translation of the before operator
6. Conclusions & perspectives

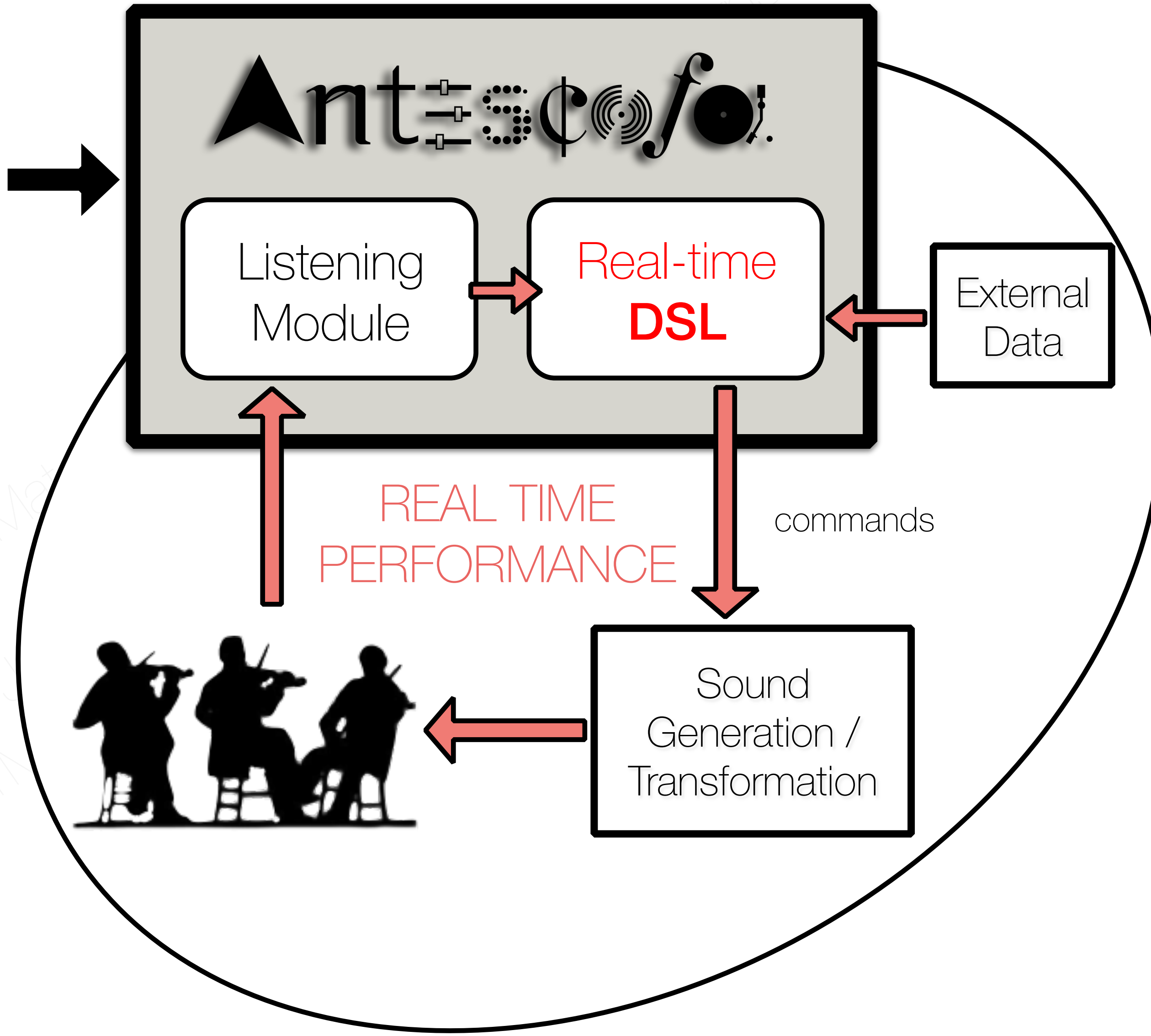
Automatic Accompaniment using *Antescofo*

Left Hand Concerto, Ravel. *Pianist*: Jacques Comby
Orchestra: recording Orchestre de Paris modulated by *Antescofo* in real time (Ircam 2014).

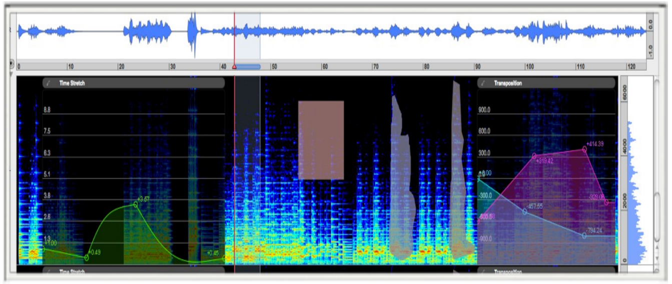
ircam Centre Pompidou



COMPOSITION TIME

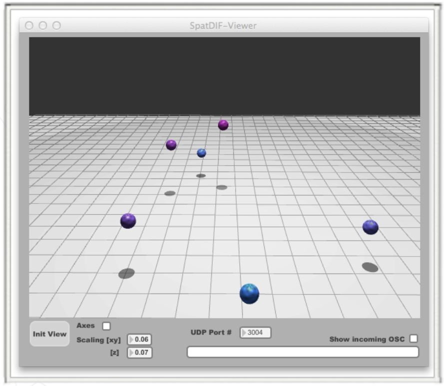


ircam Centre Pompidou



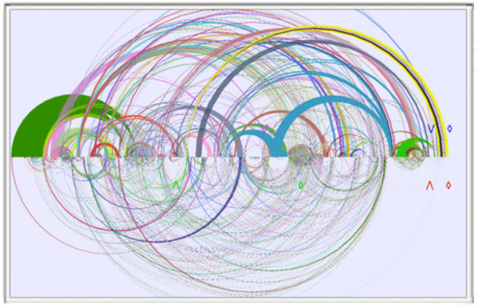
Audio Analysis/Synthesis

Gesture

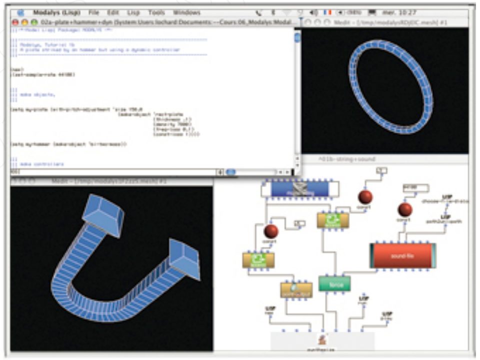


Spatialization

Antescofo



Improvisation

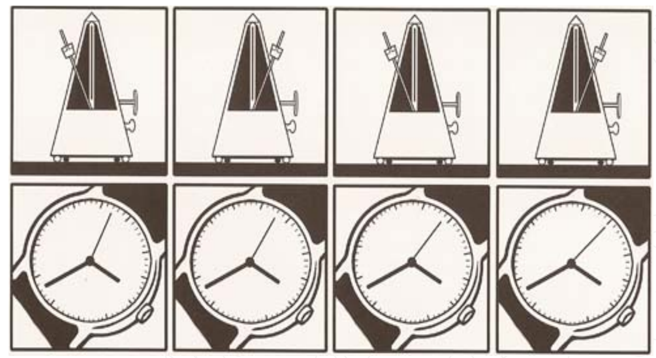


Physical Models

MIDI / Control

More than 40 Creations

New York Philharmonics, Chicago Symphony, Los Angeles Philharmonics, Berlin Philharmonics, BBC Orchestra...



AN *ANTESCOFO* GLIMPSE

Antescofo domain specific language

- handling multiple temporal references: *event* + *tempo*
  - external (*e.g.* musicien)
  - computed
  - physical (wall clock)
- tempo: the « flow » of a time reference
- duration: delays and groups lifespan (relative to a time reference)
- dynamicity:
  - process: creation, call, destruction, with their own time frame, as high-order values
  - computed delays
  - computed tempii
- augmented score as the expected (complex) temporal scenario
- performance: implementation of the temporal scenario including deviation
- synchronization & error handling w.r.t. the temporal scenario

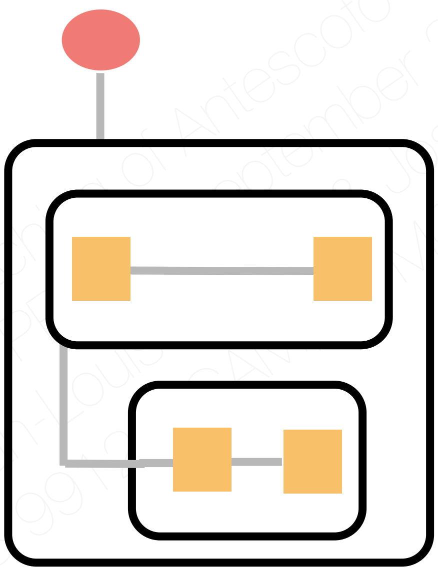
Syntax

events: `NOTE 60 2.0`

atomic actions:

```
$v := @sin($x)
superVP ($v+3)
```

compound actions:



```
group
{
        print hello
        print beautiful
    2.0 print world
}
```

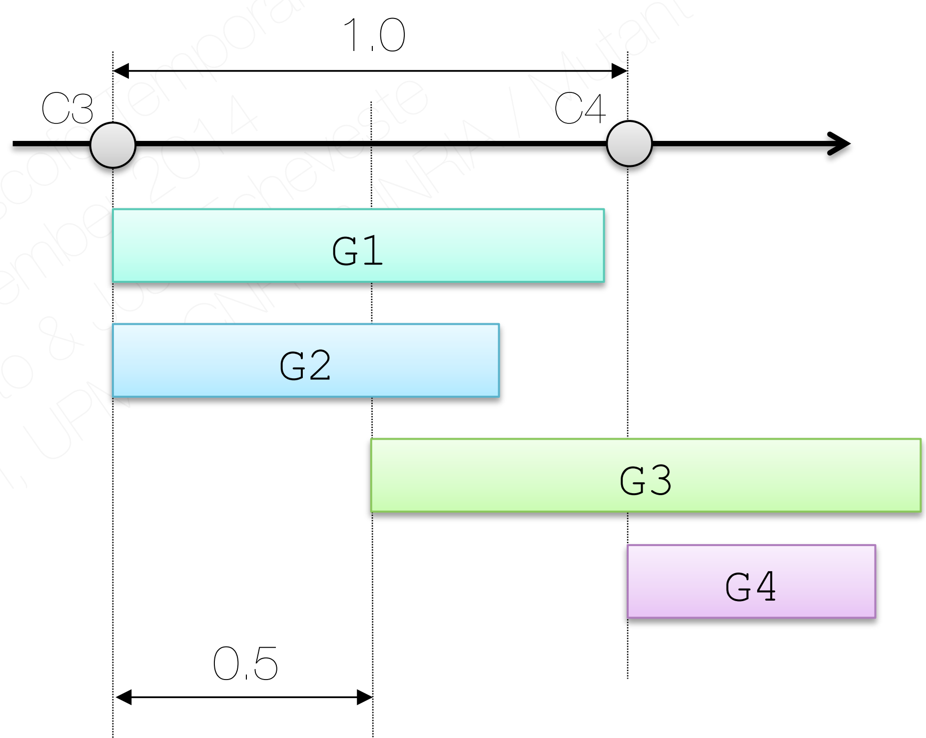
```
curve @grain 0.1s
      @action draw $x $y
{
 $x,$y {    { 0.3,   1.2 }
         4s { 0.9,   2.4 }
       }
}
```

```
loop  3.0
{
  print "loop"
} during [6#]
```

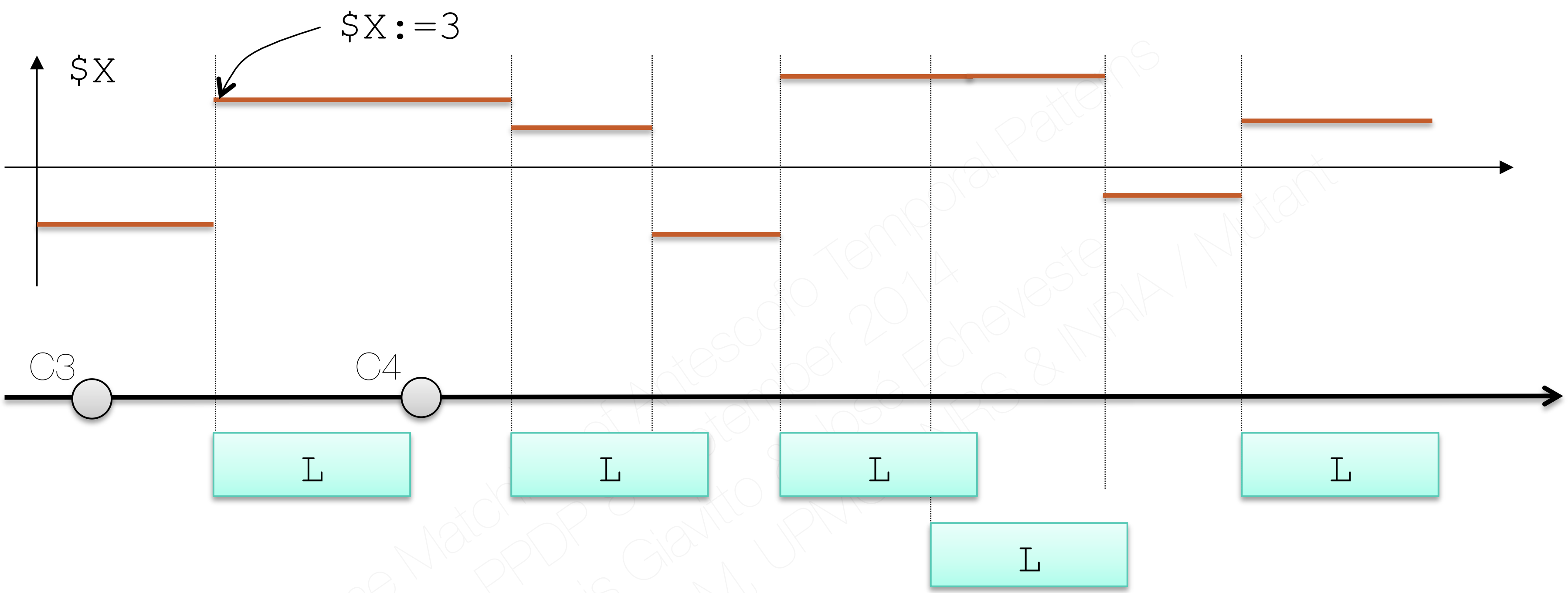
```
whenever ( $y > 3.0 )
{
   print $y "greather than 3"
}
```

Group

```
Note C3 1.0
    Group G1
    {
      G1
    }
    Group G2
    {
      G2
    }
    0.5 Group G3
    {
      G3
    }
    0.5 Group G4
    {
      G4
    }
Note C4 1.5
…
```



Whenever




```
Note C3  2/3
    Whenever ($X > 0)
    {
        L
    }
Note C4 1.5
…
```

Real-Time Matching of Antescofo Temporal Patterns
PPDP September 2014
Jean-Louis Giavitto & José Echeveste
UMR STMS 9912 IRCAM, UPMC & INRIA / Mutant

Expressions

- ## Values

  `int, float, bool, string, symbol…`
  `tab, map, continuous symbolic curve…`
  `functions, processes… (first-order values)`

- ## Operators and predefined functions

  `@sin(), @exp(), (…? … : …), @random(), @score()...`

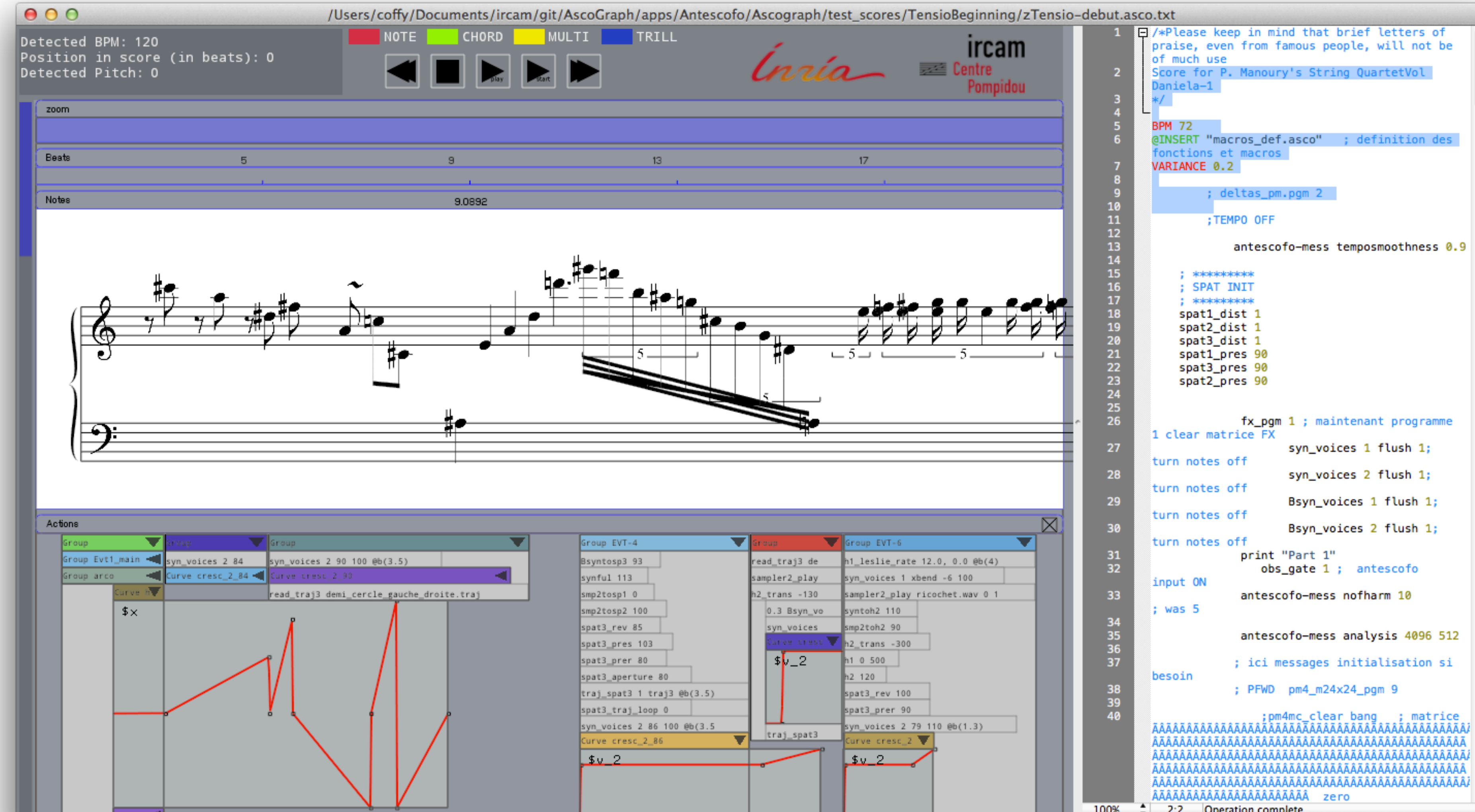
- ## Imperative Variables

  - system variables: `$RT_TEMPO $NOW $RNOW $TEMPO $PITCH`, *etc.*
  - history
    `[3#]:$x`
    `[3]:$x`
    `[3s]:$x`

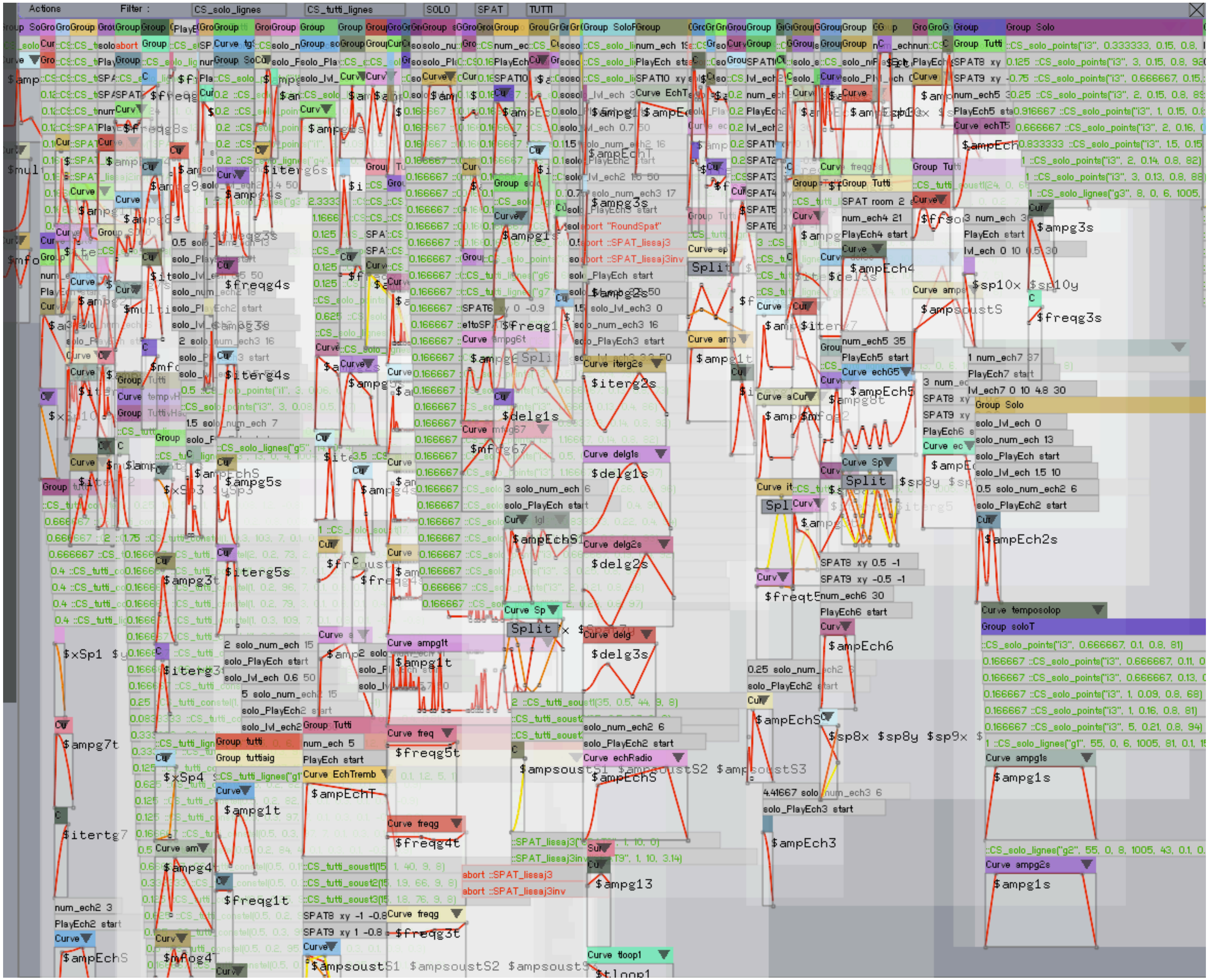
$v	*undef*	43	52	53	49
timestamps in beats	0.0	1.0	2.5	4.0	5.5
timestamps in sec	0.0	2.3	4.2	5.9	7.5

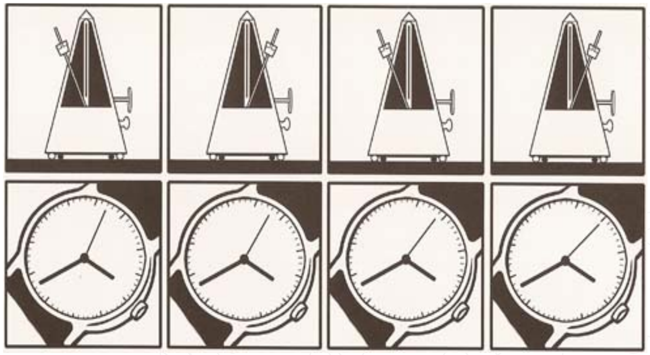
  - `@date([3#]:$x)`
    `@rdate([3#]:$x)`

ircam Centre Pompidou



Example: sound synthesis control in Nachtleben (5', Julia Blondeau)

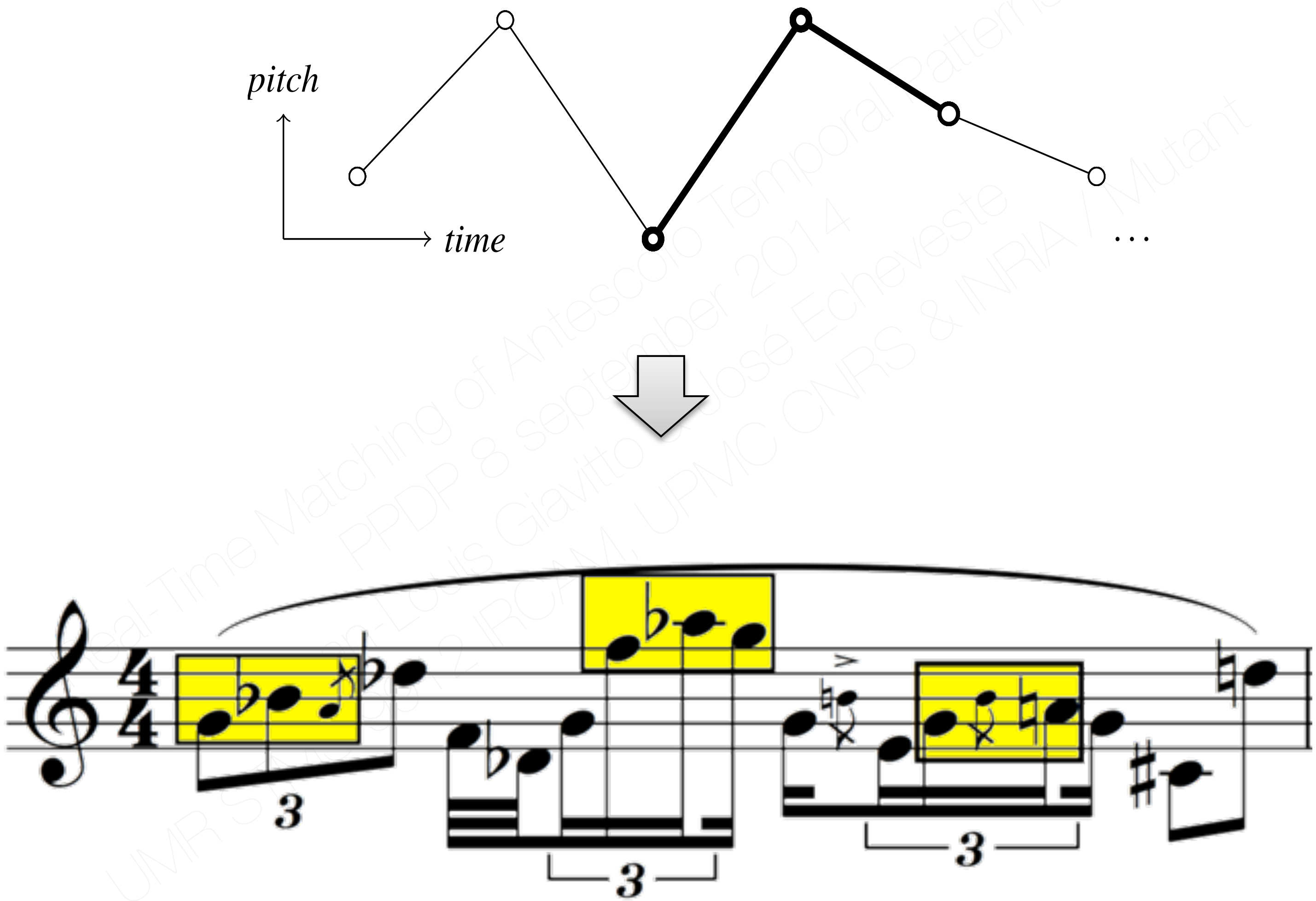




WHY TEMPORAL PATTERNS?

UMR STMS 9912 IRCAM-UPMC CNRS INRIA / Mutant

Neume

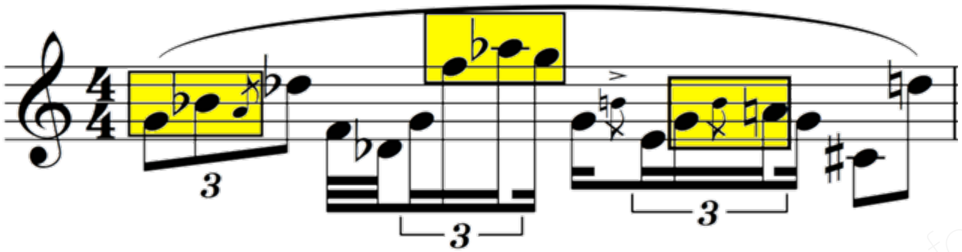


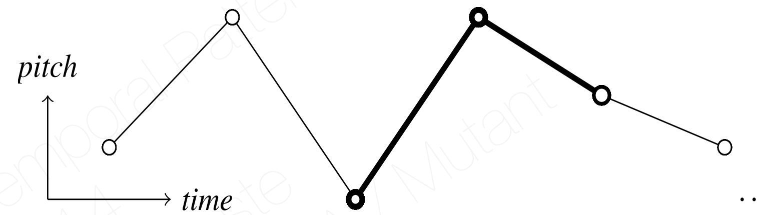
Real-Time Matching of Antescofo Temporal Patterns
PPDP 8 september 2014
Jean-Louis Giavitto José Echeveste
UMR STMS 9912 IRCAM, UPMC CNRS & INRIA / Mutant

LIVE EXAMPLE

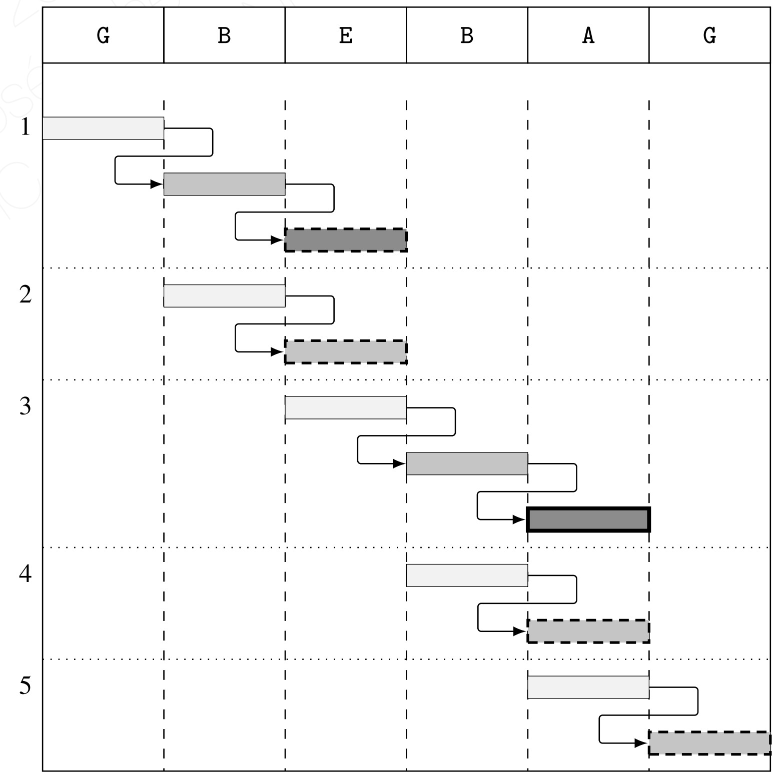
Real-Time Matching of Antescofo Temporal Patterns
PPDP 8 septembre 2014
Jean-Louis Giavitto & José Echeveste
UMR STMS 9912 IRCAM, UPMC CNRS & INRIA / Mutant

Matching a temporal pattern






```
whenever ($PITCH) {
  @local $x
  $x := $PITCH
  whenever ($PITCH > $x) {
    @local $y
    $y := $PITCH
    whenever ($PITCH<$y & $PITCH>$x) {
      @local $z
      $z := $PITCH
      a
    } during [1#]
  } during[1#]
}
```



THE PATTERN LANGUAGE: STATE & EVENT

Real-Time Matching of Antescofo Temporal Patterns
PPDP 8 septembre 2014
Jean-Louis Giavitto & José Echeveste
UMR STMS 9912 IRCAM, UPMC CNRS & INRIA / Mutant

Event: checking an instantaneous property

```
Pattern P
{
    @local $x , $y , $z
    Event $PITCH value $x
    Event $PITCH value $y where $x < $y
    Event $PITCH value $z where ($y > $z) & ($z > $x)
}
…
whenever P
{ print  "I just saw a P" }
```

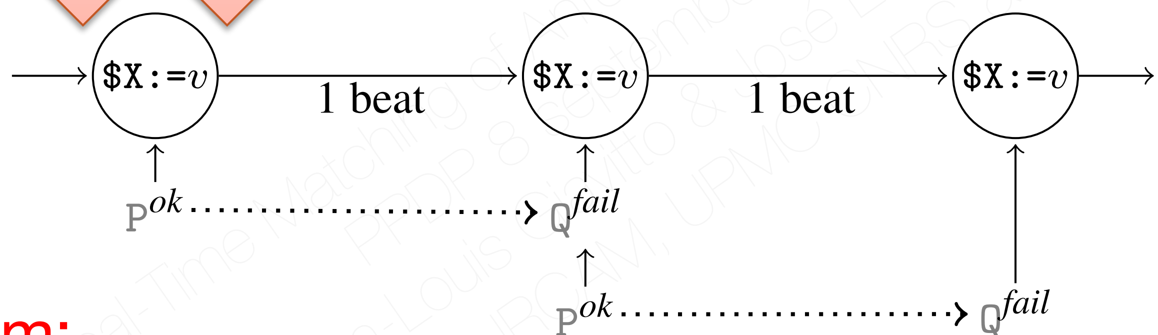
```
@pattern twice
{
  @local $v
  Event $V value $v
  Before [3] Event $V value $v
}
```

Real-Time Matching of Antescofo Temporal Patterns
PPDP 8 september 2014
Jean-Louis Giavitto & José Echeveste
UMR STMS 9912 IRCAM, UPMC CNRS & INRIA / Mutant

State: a property that lasts

variable $X takes the value *v* at least for 2 beats

```
@local    ta      $stop, $w
P:  Eve        CH  value v at $start
Q:  Even      CH  value $w at $stop where ($stop - $start) ≥ 2
```



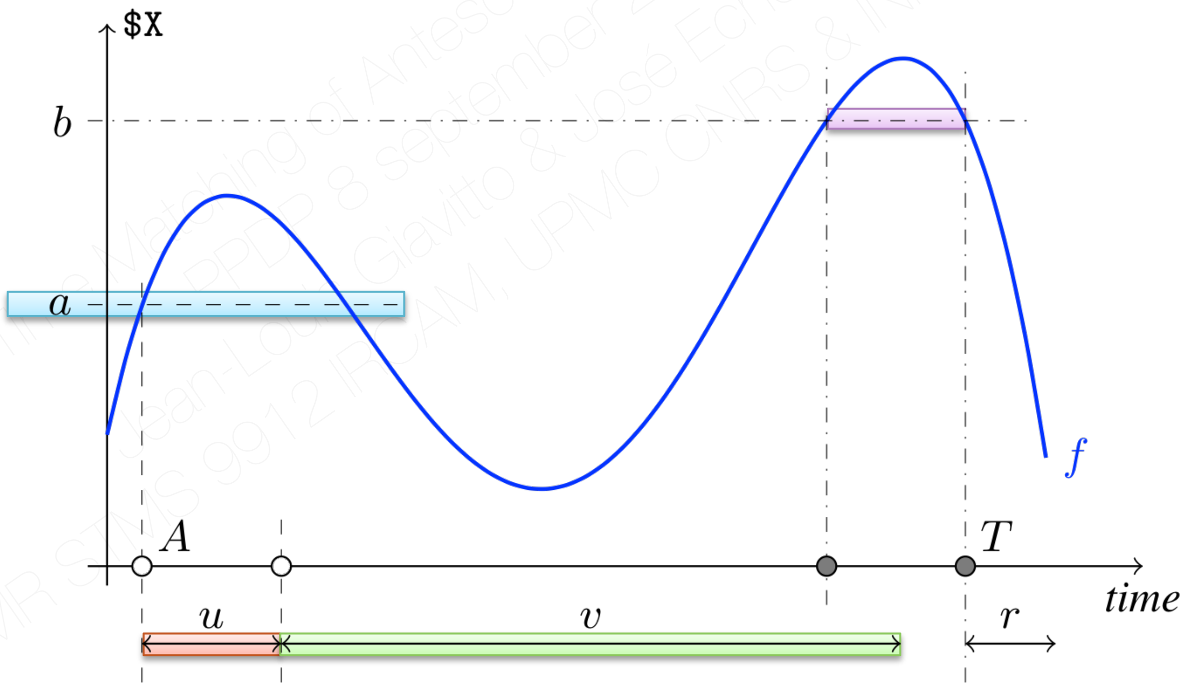
Problem:
they can be an unbounded number of events in the interval

```
State $X where ($X == v) during 2
```

Real-Time Matching of Antescofo Temporal Patterns
PPDP 8 September 2014
Jean-Louis Giavitto & José Echeveste
UMR STMS 9912 IRCAM, CNRS, UPMC & INRIA / Mutant

State: a property that lasts

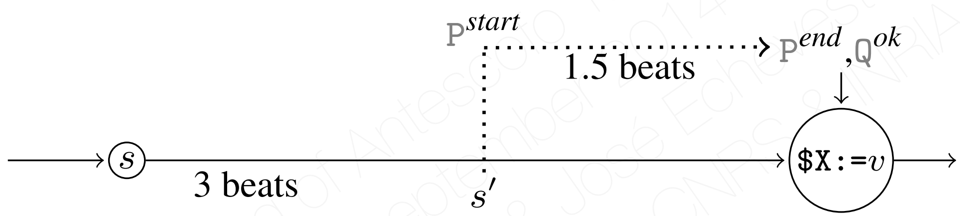
```
State $X during u where $X > a
Before [v]
   State $X where $X > b
```



Real-Time Matching of Antescofo Temporal Patterns PPDP 8 september 2014 Jean-Louis Giavitto & José Echeveste UMR STMS 9912 IRCAM, UPMC CNRS & INRIA / Mutant

Lasting properties do not start everywhere

```
State $X where true during[1.5]
Event $X where ($X == v)
```

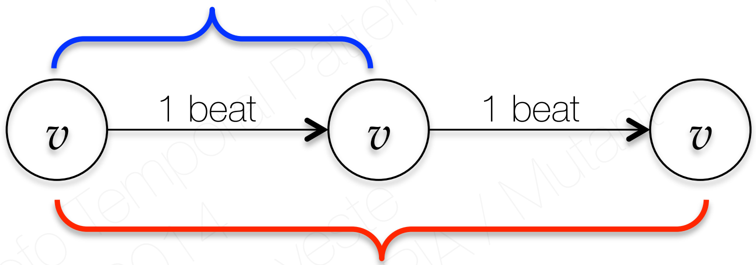


- Continuous time but only a discrete number of events
- The `predicate` feature in FRAN does not apply (the date of the future event is not known yet)
- Implementation require either
  - a sampling of continuous time (and the start of a potential match at each sampled instant)
  - or the access of all past states (*i.e.* an unbounded memory)

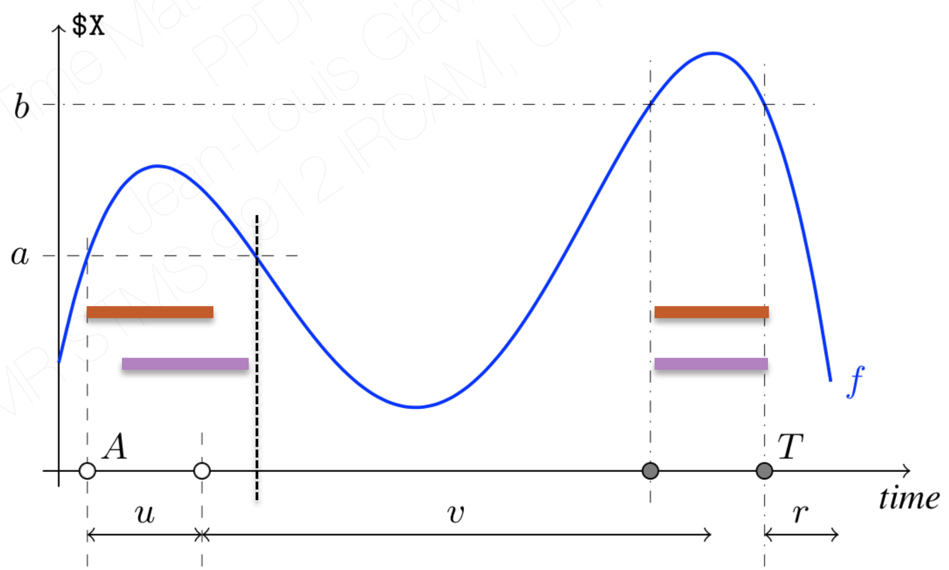
Which match ?

- Earliest match

```
@pattern TwiceIn3B {
  @local $v
  Event $V value $v
  Before[3] Event $V value $v
}
```

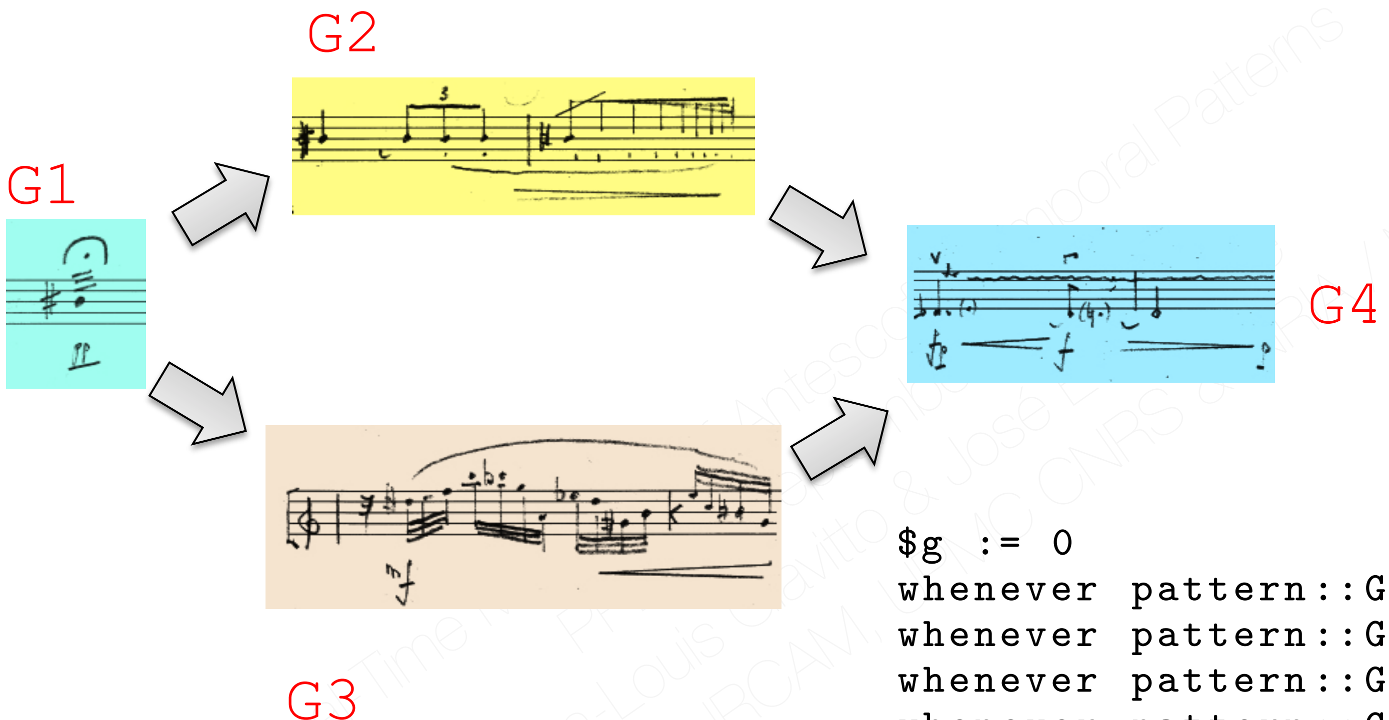


- Refractory period

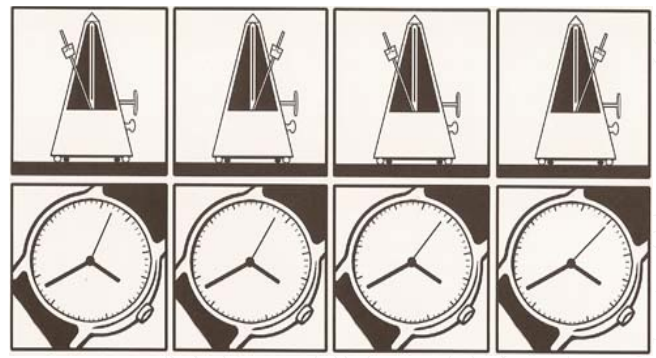


Real-Time Matching of Antescofo Temporal Patterns
PPDP 8 september 2014
Jean-Louis Giavitto & José Echeveste
UMR STMS 9912 IRCAM, UPMC CNRS & INRIA / MuTant

Composing and Chaining Patterns



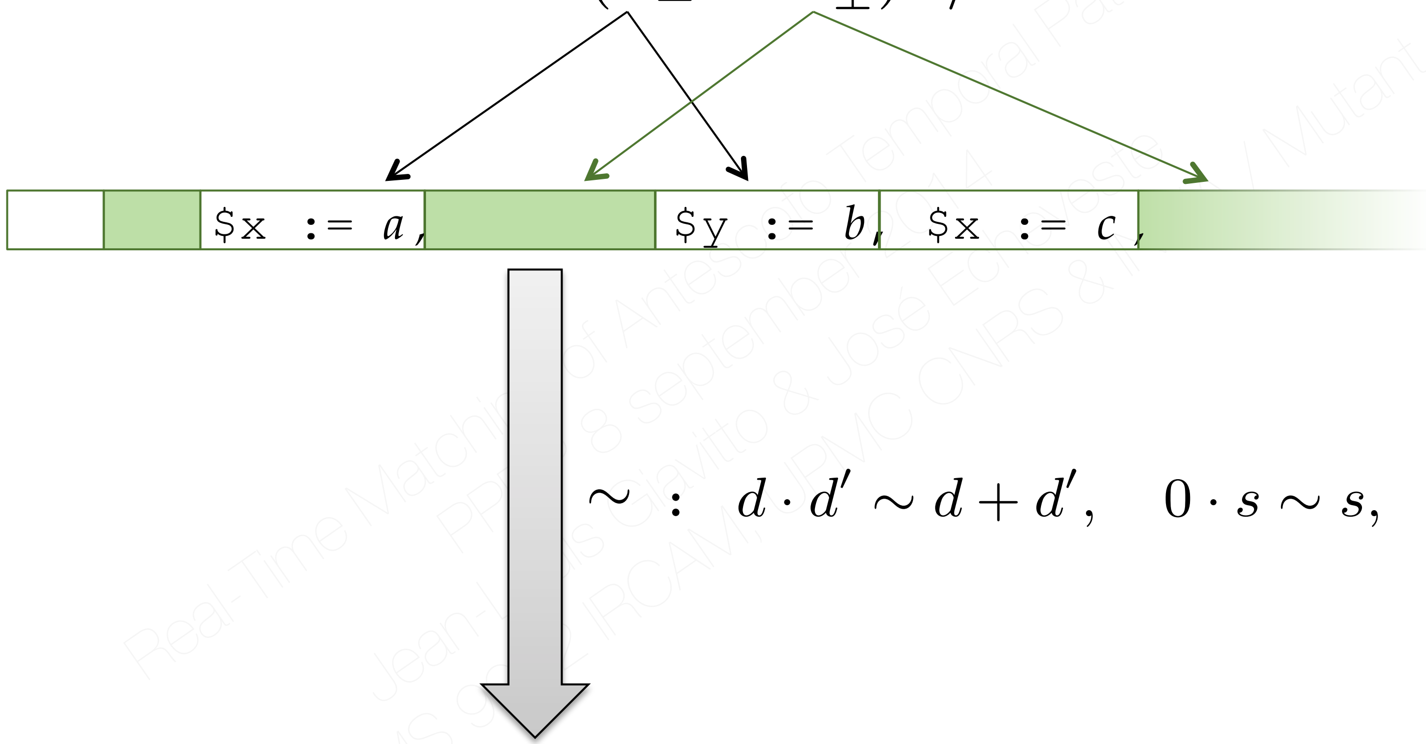

```
$g := 0
whenever pattern::G1 { $g := 1 }
whenever pattern::G2 { $g := 2 }
whenever pattern::G3 { $g := 3 }
whenever pattern::G4 { $g := 4 }

@pattern Gseq {
  Event $g value 1
  Event $g where ($g==2)||($g==3)
  Event $g value 4
}
...
whenever pattern::Gseq { ... }
```



TEMPORAL PATTERNS SEMANTICS

Domain: Time-Event Sequences

$$\mathcal{S} = (\mathcal{U}_\perp \cup \mathbb{R}^+_\perp)^* \,/\, \sim$$



$$\sim \; : \quad d \cdot d' \sim d + d', \quad 0 \cdot s \sim s, \quad s \cdot 0 \sim s$$

Time passages are not divisible

One cannot insert new time instants at will

The Semantics Function

$$\mathbf{M} : \mathcal{P} \to \mathcal{E} \to \mathcal{S} \to \mathbb{R}^{+} \cup \{fail\}$$

The input (the future)

The environment: variable → value

The date of the action launched as the result of the match of the pattern

$$\begin{aligned}
\mathcal{P} &::= \varepsilon \mid \mathit{Event} \cdot \mathcal{P} \mid \mathit{State} \cdot \mathcal{P} \\
\mathit{Event} &::= \texttt{Before[}\ \mathit{Dur}\ \texttt{]}\ \texttt{Event}\ \mathcal{I}\ \texttt{at}\ \mathcal{I}\ \texttt{value}\ \mathcal{I}\ \texttt{where}\ \mathit{Exp} \\
\mathit{State} &::= \texttt{Before[}\ \mathit{Dur}\ \texttt{]}\ \texttt{State}\ \mathcal{I}\ \texttt{where}\ \mathit{Exp} \\
&\quad \mid \texttt{Before[}\ \mathit{Dur}\ \texttt{]}\ \texttt{State}\ \mathcal{I}\ \texttt{where}\ \mathit{Exp}\ \texttt{during[}\ \overline{\mathbb{R}^{+}}\ \texttt{]} \\
\mathit{Dur} &::= \overline{\mathbb{R}^{+}} \mid \mathbb{N}\#
\end{aligned}$$

Semantics Equations

$$(1)\quad \mathbf{M}[\![\varepsilon]\!]\rho\, S = \rho(\texttt{\$NOW})$$

$$(2)\quad \mathbf{M}[\![P]\!]\rho\,\epsilon = \mathit{fail}, \qquad P \neq \varepsilon$$

$$(3)\quad \mathbf{M}[\![P_x \cdot Q]\!]\;\rho\;(x' := v \cdot S) = \mathbf{M}[\![P_x \cdot Q]\!]\;\rho[x' := v]\;S \quad \text{where } x \neq x'$$

let $P_x = \texttt{Event}\ x\ \texttt{at}\ y\ \texttt{value}\ z\ \texttt{where}\ e$ **in:**

$$(4)\quad \mathbf{M}[\![\texttt{before[}\,d\,\texttt{]}\ P_x \cdot Q]\!]\;\rho\;(d' \cdot S) = \begin{cases} \mathit{fail}, & \text{if } d \leq d' \\ \mathbf{M}[\![\texttt{before[}\,d - d'\,\texttt{]}\ P_x \cdot Q]\!]\;\rho[\texttt{\$NOW} \mathrel{+}= d']\;S, & \text{if } d > d' \end{cases}$$

$$(5)\quad \mathbf{M}[\![\texttt{before[}0\texttt{\#]}\ P_x \cdot Q]\!]\;\rho\;S = \mathit{fail}$$

$$(6)\quad \mathbf{M}[\![\texttt{before[}\,n\texttt{\#}\,\texttt{]}\ P_x \cdot Q]\!]\;\rho\;(d' \cdot S) = \mathbf{M}[\![\texttt{before[}\,n\texttt{\#}\,\texttt{]}\ P_x \cdot Q]\!]\;\rho\;S$$

$$(7)\quad \mathbf{M}[\![\texttt{before[}\,D\,\texttt{]}\ P_x \cdot Q]\!]\;\rho\;(x := v \cdot S) = \begin{cases} \mathbf{M}[\![P'_x \cdot Q]\!]\;\rho'\;S, & \text{if } \mathbf{E}[\![e]\!]\rho'' = \mathit{false} \\ \min(\mathbf{M}[\![P'_x \cdot Q]\!]\;\rho'\;S,\ \ \mathbf{M}[\![Q]\!]\;\rho''\;S) & \text{if } \mathbf{E}[\![e]\!]\rho'' = \mathit{true} \end{cases}$$

$$\text{where } \rho' = \rho[x := v] \text{ and } \rho'' = \rho'[y := \rho(\texttt{\$NOW}), z := v] \text{ and } P'_x = \begin{cases} \texttt{before[}\,d\,\texttt{]}\ P_x, & \text{if } D = d \\ \texttt{before[}\,(n-1)\texttt{\#}\,\texttt{]}\ P_x, & \text{if } D = n\texttt{\#} \end{cases}$$

let $P_x = \texttt{State}\ x\ \texttt{where}\ e$ **and** $\overline{P}_x \in \{P_x,\ P_x\,\texttt{during[}\,D\,\texttt{]}\}$ **in:**

$$(8)\quad \mathbf{M}[\![\texttt{before[}\,d\,\texttt{]}\ \overline{P}_x \cdot Q]\!]\;\rho\;(d' \cdot S) = \begin{cases} \mathit{fail} & \text{if } d \leq d' \ \wedge\ \mathbf{E}[\![e]\!]\rho = \mathit{false} \\ \mathbf{M}[\![\texttt{before[}\,d - d'\,\texttt{]}\ \overline{P}_x \cdot Q]\!]\;\rho'\;\cdot S & \text{if } d > d' \ \wedge\ \mathbf{E}[\![e]\!]\rho = \mathit{false} \\ \mathbf{M_S}[\![\overline{P}_x \cdot Q]\!]\;\rho\;(d' \cdot S) & \text{if } d \leq d' \ \wedge\ \mathbf{E}[\![e]\!]\rho = \mathit{true} \\ \min\begin{pmatrix} \mathbf{M_S}[\![\overline{P}_x \cdot Q]\!]\;\rho\;(d' \cdot S), \\ \mathbf{M}[\![\texttt{before[}\,d - d'\,\texttt{]}\ \overline{P}_x \cdot Q]\!]\;\rho'\;\cdot S \end{pmatrix} & \text{if } d > d' \ \wedge\ \mathbf{E}[\![e]\!]\rho = \mathit{true} \end{cases}$$

$$\text{where } \rho' = \rho[\texttt{\$NOW} \mathrel{+}= d']$$

$$(9)\quad \mathbf{M}[\![\texttt{before[}\,d\,\texttt{]}\ \overline{P}_x \cdot Q]\!]\;\rho\;(x := v \cdot S) = \begin{cases} \mathbf{M}[\![\texttt{before[}\,d\,\texttt{]}\ \overline{P}_x \cdot Q]\!]\;\rho'\;S & \text{if } \mathbf{E}[\![e]\!]\rho' = \mathit{false} \\ \min\begin{pmatrix} \mathbf{M_S}[\![\overline{P}_x \cdot Q]\!]\;\rho'\;\cdot S, \\ \mathbf{M}[\![\texttt{before[}\,d\,\texttt{]}\ \overline{P}_x \cdot Q]\!]\;\rho'\;\cdot S \end{pmatrix} & \text{if } \mathbf{E}[\![e]\!]\rho' = \mathit{true} \end{cases}$$

$$\text{where } \rho' = \rho[x := v]$$

$$(10)\quad \mathbf{M_S}[\![\overline{P}_x \cdot Q]\!]\;\rho\;\epsilon = \mathit{fail}$$

$$\mathbf{M_S}[\![\overline{P}_x \cdot Q]\!]\;\rho\;(x' := v \cdot S) = \mathbf{M_S}[\![\overline{P}_x \cdot Q]\!]\;\rho[x' := v]\;S \quad \text{where } x \neq x'$$

$$(11)\quad \mathbf{M_S}[\![P_x \cdot Q]\!]\;\rho\;(d' \cdot S) = \mathbf{M_S}[\![P_x \cdot Q]\!]\;\rho[\texttt{\$NOW} := d']\;S$$

$$\mathbf{M_S}[\![P_x \cdot Q]\!]\;\rho\;(x := v \cdot S) = \begin{cases} \mathbf{M_S}[\![P_x \cdot Q]\!]\;\rho[x := v]\;S & \text{if } \mathbf{E}[\![e]\!]\rho[x := v] = \mathit{true} \\ \mathbf{M}[\![Q]\!]\;\rho[x := v]\;S & \text{if } \mathbf{E}[\![e]\!]\rho[x := v] = \mathit{false} \end{cases}$$

$$(12)\quad \mathbf{M_S}[\![P_x\,\texttt{during[}\,d\,\texttt{]}\ \cdot Q]\!]\;\rho\;(d' \cdot S) = \begin{cases} \mathbf{M}[\![Q]\!]\;\rho[\texttt{\$NOW} \mathrel{+}= d]\;(d' - d \cdot S) & \text{if } d \leq d' \\ \mathbf{M_S}[\![P_x\,\texttt{during[}\,d - d'\,\texttt{]}\ \cdot Q]\!]\;\rho[\texttt{\$NOW} \mathrel{+}= d']\;S & \text{if } d > d' \end{cases}$$

$$\mathbf{M_S}[\![P_x\,\texttt{during[}\,d\,\texttt{]}\ \cdot Q]\!]\;\rho\;(x := v \cdot S) = \begin{cases} \mathit{fail} & \text{if } \mathbf{E}[\![e]\!]\rho[x := v] = \mathit{false} \\ \mathbf{M_S}[\![P_x\,\texttt{during[}\,d\,\texttt{]}\ \cdot Q]\!]\;\rho[x := v]\;S & \text{if } \mathbf{E}[\![e]\!]\rho[x := v] = \mathit{true} \end{cases}$$

The Semantics Equations

$$(1) \quad \mathbf{M}[\![\varepsilon]\!]\,\rho\,S = \rho(\texttt{\$NOW})$$

The empty pattern matches immediately

$$(2) \quad \mathbf{M}[\![P]\!]\,\rho\,\epsilon = \mathit{fail}, \qquad P \neq \varepsilon$$

A non empty pattern fails to match at the end of time
(when there is no more future)

$$(3) \quad \mathbf{M}[\![P_x \cdot Q]\!]\ \rho\ (x' := v \cdot S) = \mathbf{M}[\![P_x \cdot Q]\!]\ \rho[x' := v]\ S \quad \text{where } x \neq x'$$

The occurrence of an event on x has no effect on the pattern matching except the update of the environment

Real Time Matching of Antescofo Temporal Patterns
PPDP 8 September 2014
Jean-Louis Giavitto & José Echeveste
UMR STMS 9912 IRCAM, UPMC CNRS & INRIA / Mutant

The Semantics Equations

$$\textbf{let } P_x = \texttt{Event } x \texttt{ at } y \texttt{ value } z \texttt{ where } e$$

$$(4) \quad \mathbf{M}[\![\, \texttt{Before[} d \texttt{]}\ P_x \cdot Q]\!]\ \rho\ (d' \cdot S) = \begin{cases} \mathit{fail}, & \text{if } d \leq d' \\ \mathbf{M}[\![\, \texttt{Before[} d - d' \texttt{]}\ P_x \cdot Q]\!]\ \rho[\texttt{\$NOW} \mathrel{+}= d']\ S, & \text{if } d > d' \end{cases}$$

if a pattern must match before ***d*** and

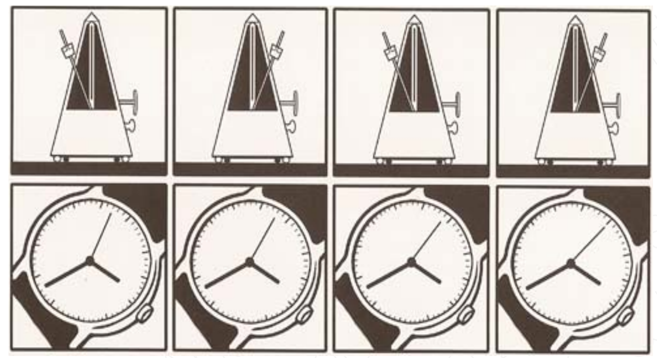
- nothing happens before ***d' > d***, then the matching fails
- if something happens, then the time passage ***d'*** is subtracted from ***d***

$$(7) \quad \mathbf{M}[\![\, \texttt{Before[} D \texttt{]}\ P_x \cdot Q]\!]\ \rho\ (x := v \cdot S) = \begin{cases} \mathbf{M}[\![P'_x \cdot Q]\!]\ \rho'\ S, & \text{if } \mathbf{E}[\![e]\!] \rho'' = \mathit{false} \\ \min(\mathbf{M}[\![P'_x \cdot Q]\!]\ \rho'\ S,\ \ \mathbf{M}[\![Q]\!]\ \rho''\ S) & \text{if } \mathbf{E}[\![e]\!] \rho'' = \mathit{true} \end{cases}$$

$$\text{where } \rho' = \rho[x := v] \text{ and } \rho'' = \rho'[y := \rho(\texttt{\$NOW}), z := v] \text{ and } P'_x = \begin{cases} \texttt{Before[} d \texttt{]}\ P_x, & \text{if } D = d \\ \texttt{Before[} (n-1)\texttt{\#]}\ P_x, & \text{if } D = n\texttt{\#} \end{cases}$$

if something happens it can either

- participate to the match
- or not



IMPLEMENTATION SKETCH

UMR STMS 9912 IRCAM UPMC CNRS INRIA / Mutant

ircam
Centre
Pompidou

Compilation through translation

```
@pattern_def  pattern::Gong
{
   @Local $x, $y, $z, $s2, $s3

   Event $S value $x
   Event $S value $y at $s2
   Event $S value $z at $s3
        where  ($z in $x .. $y)
            &&  ($s3 - $s2) < 2
}

whenever pattern::Gong
{
   print GOTCHA  GONG $s3
}
```

```
WHENEVER  ( $S == $S )
{
     @local $__71_continue, $x

     let $__71_continue := true
     let $x := $S
     WHENEVER  ( $__71_continue && $S == $S )
     {
          @local $s2, $y

          let $s2:= $NOW
          let $y := $S
          WHENEVER  ( $__71_continue && $S == $S )
           {
               @local $s3, $z

               let $s3 := $NOW
               let $z := $S
               if (@between($x, $z, $y) && ($s3-$s2 < 2))
               {
                    let $__71_continue := false
                    print GOTCHA GONG $s3
               }
           } during [1 #]
     } during [1 #]
}
```

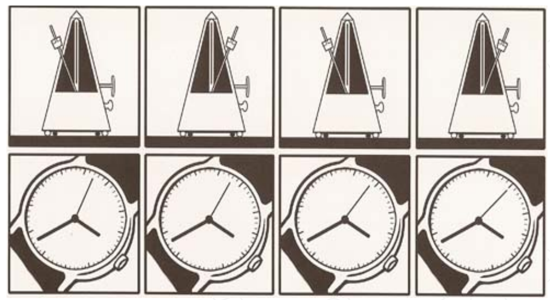
Handling efficiently the *before* operator

@pattern_def pattern::P
{
Event $X
Before [2] Event $X where $X > 0
}

$$(7)\quad \mathbf{M}[\![\texttt{Before[}\,D\,\texttt{]}\ P_x \cdot Q]\!]\ \rho\ (x := v \cdot S) = \begin{cases} \mathbf{M}[\![P'_x \cdot Q]\!]\ \rho'\ S, & \text{if } \mathbf{E}[\![e]\!]\rho'' = \mathit{false} \\ \min(\mathbf{M}[\![P'_x \cdot Q]\!]\ \rho'\ S,\ \ \mathbf{M}[\![Q]\!]\ \rho''\ S) & \text{if } \mathbf{E}[\![e]\!]\rho'' = \mathit{true} \end{cases}$$

$$\text{where } \rho' = \rho[x := v] \text{ and } \rho'' = \rho'[y := \rho(\texttt{\$NOW}), z := v] \text{ and } P'_x = \begin{cases} \texttt{Before[}\,d\,\texttt{]}\ P_x, & \text{if } D = d \\ \texttt{Before[}\,(n-1)\texttt{\#}\,\texttt{]}\ P_x, & \text{if } D = n\texttt{\#} \end{cases}$$

```
WHENEVER ( $X == $X )   {
    @local $__2_continue

    let $__2_continue := true
    WHENEVER  ( $__2_continue && $X == $X )  {
        if ($X > 0) {
            let $__2_continue := false
            print OK
        }
    } during [2]
}
```



CONCLUSION

Real-Time Matching of Antes Patterns
Jean-Louis & José Echeveste
UMR STMS 9912 IRCAM, UPMC CNRS & INRIA / Mutant

Temporal Patterns

- Inspired by "regular expression" but
  - infinite alphabet : event valued in an unbounded alphabet (variables + values)
  - arbitrary predicate
  - state pattern (a causal version of *)
  - causal (no crystal ball)
- First dedicated implementation
  - more efficient
  - less expressive
  - heavy to maintain
- Current implementation by translation : efficient enough for the current applications
- Used in a few concerts

Perspectives

- Concise Semantics
  - do not handle patterns as their own source of event
  - causality is hidden
    but some result may express causality: prefix computation
  - more work needed to delineate what kind of state can be expressed
  - difficult to compare precisely with current work in temporal logic

- Extensions
  - additional operators (NoEvent)
  - extending state properties (if bounded memory)
  - expressing audio signal transformation (in spectral domain ?)
  - more efficient translation (using variable's history)
  - more demanding applications (*e.g.*, probabilistic matching)

Antescofo

http://repmus.ircam.fr/antescofo

http://forumnet.ircam.fr/user-groups/antescofo