



Antescofo le temps-réel en musique

Jean-Louis Giavitto, José Echeveste & Arshia Con
CISEC - Mardi 10 juin 2014, Toulouse



Antescofo

L'interaction en temps-réel en musique

Jean-Louis Giavitto, José Echeveste & Arshia Cont
CISEC - Mardi 10 juin 2014, Toulouse



ircam
Centre
Pompidou



Départements

- R&D (9 teams)
- Production & Diffusion
- Transmission
- Bibliothèque multimedia

> 180 collaborateurs

- ~ 65 chercheurs
- 25 PhD + post-docs
- 20 étudiants en composition
- 15 résidences
- Compositeurs, Artistes, etc.

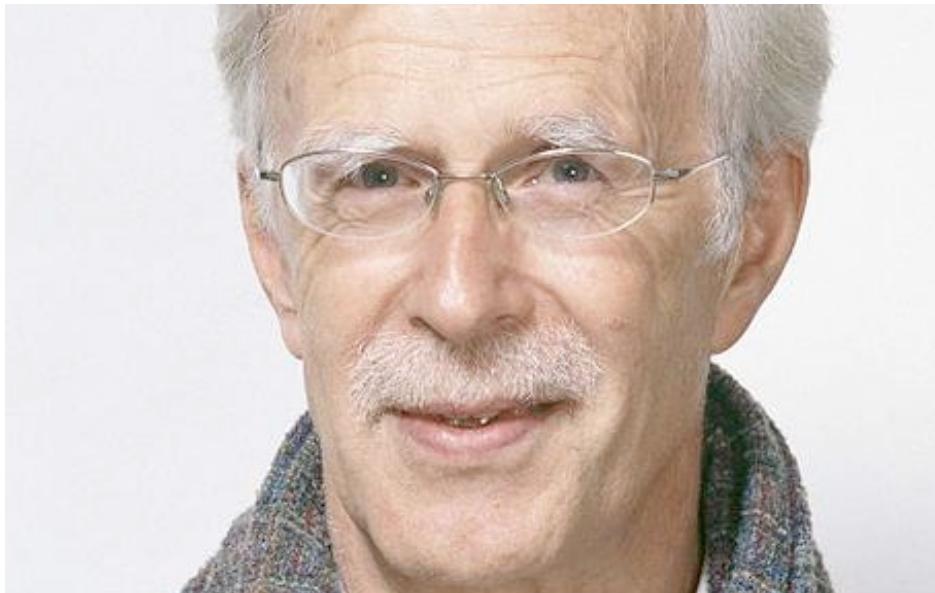


Du laboratoire à la scène

Jonathan Harvey Speakings

pour Orchestra & Live Electronique

G. Nouno, A. Cont: Computer Music Designers



Jonathan Harvey's « Speakings »

- Jonathan Harvey (1939 — 2012)
- Idea: An orchestra that speaks
 - From speech to orchestra and vice versa
 - Enrich orchestral timbre using non-existing speech structures
- Read:
 - G. Nouno, A. Cont, G. Carpentier, and J. Harvey
“Making an Orchestra Speak”, *Sound and Music Computing Conference*, 2009.
Best paper award.
 - Tribune article on Forumnet with Max and OpenMusic patches.
- Precedence: Modest Mussorgsky, Clarence Barlow
- Procedure:
 - Study existing artistic approaches (Modest Mussorgsky, Clarence Barlow, ...)
 - Musical research: Prototype ideas, artistic trial and error, choose pertinent approaches.
 - Development & Composition
 - Tests and performance considerations
 - Rehearsal and Concert (Royal Albert Hall, Proms Festival, BBC Orch. 2008)

Phase I: Computer Aided Composition

■ Automatic Orchestration

- How to write speech structures for Orchestra?
- The Orchidée project



A detailed handwritten musical score for orchestra, spanning multiple pages. The score includes parts for various instruments such as strings (Violin 1, Violin 2, Viola, Cello), woodwinds (Flute, Clarinet, Bassoon, Trombone, Tuba), brass (Trumpet, Horn), and percussion (Drums, Cymbals). The score features complex rhythmic patterns, dynamic markings like 'f' (fortissimo) and 'ff' (fortississimo), and performance instructions like 'solo' and 'ensemble'. The title 'KK' appears twice in the score.

Phase I: CAC

■ Speech to music transcription



A Game of Chess

1 2 3 4 5

voice

partials

6 7 8

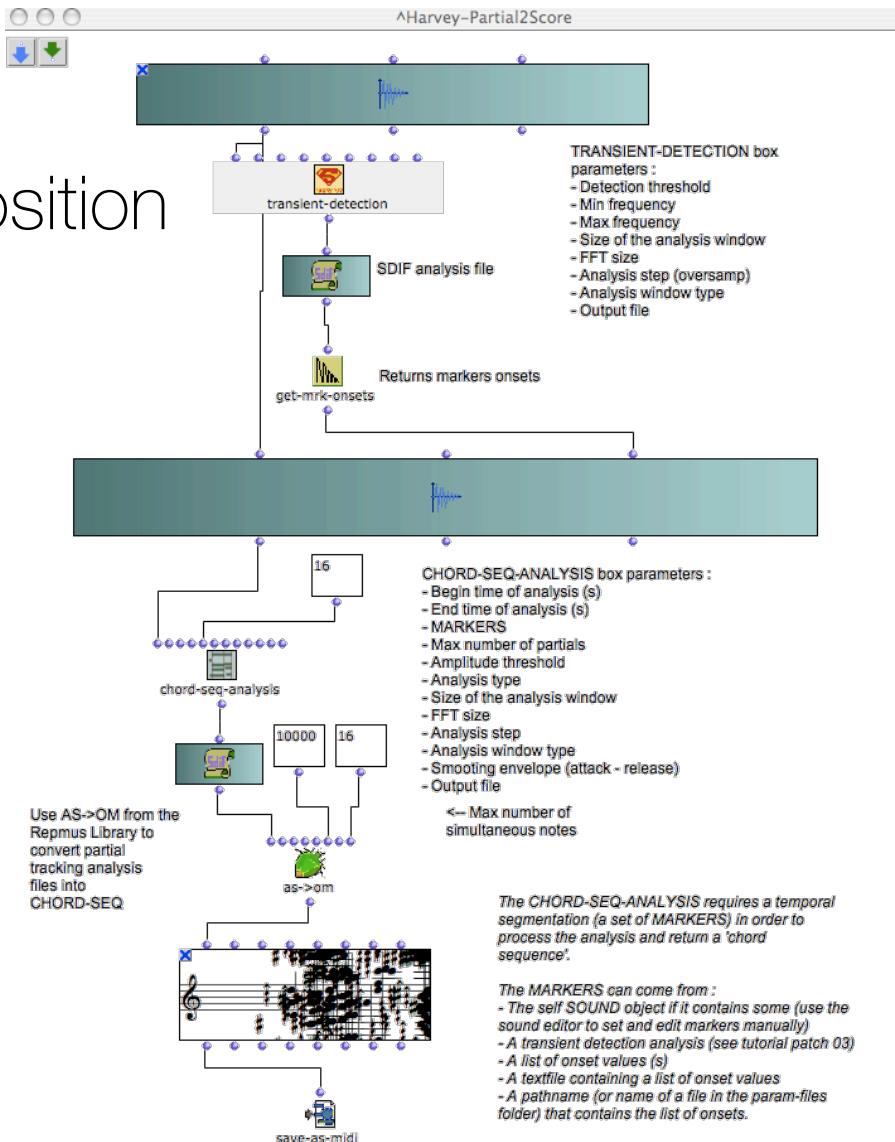
9

10 11



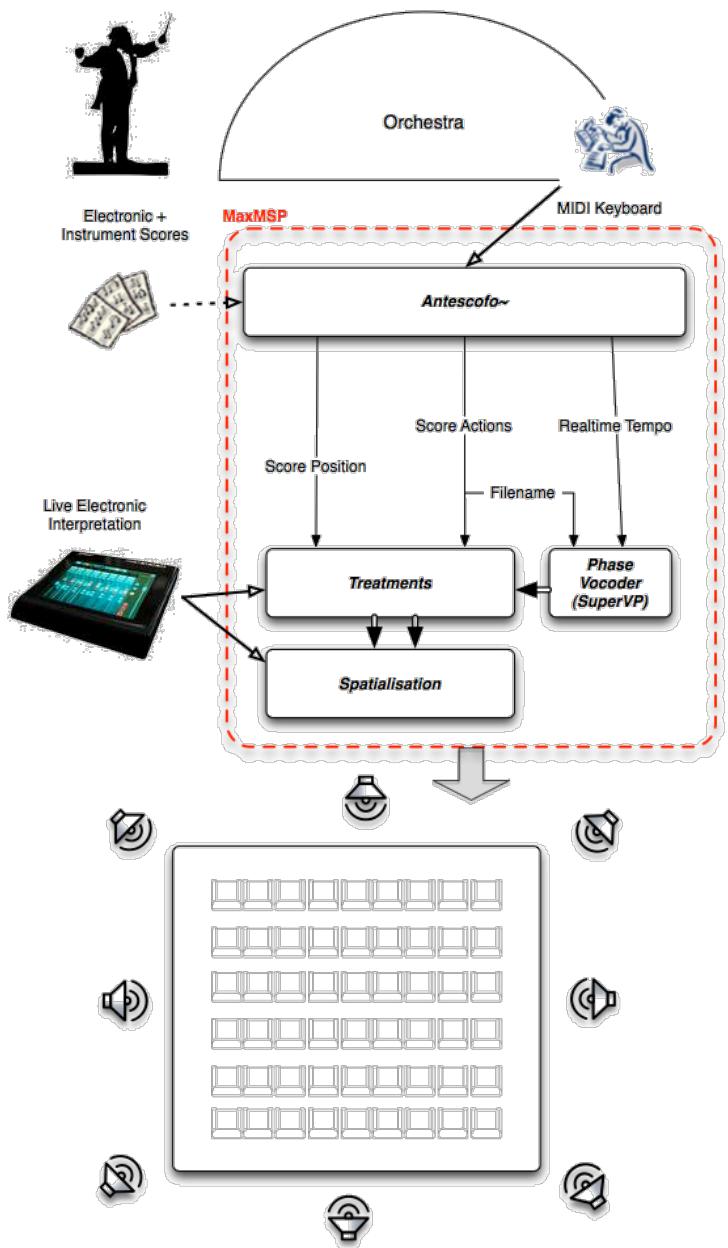
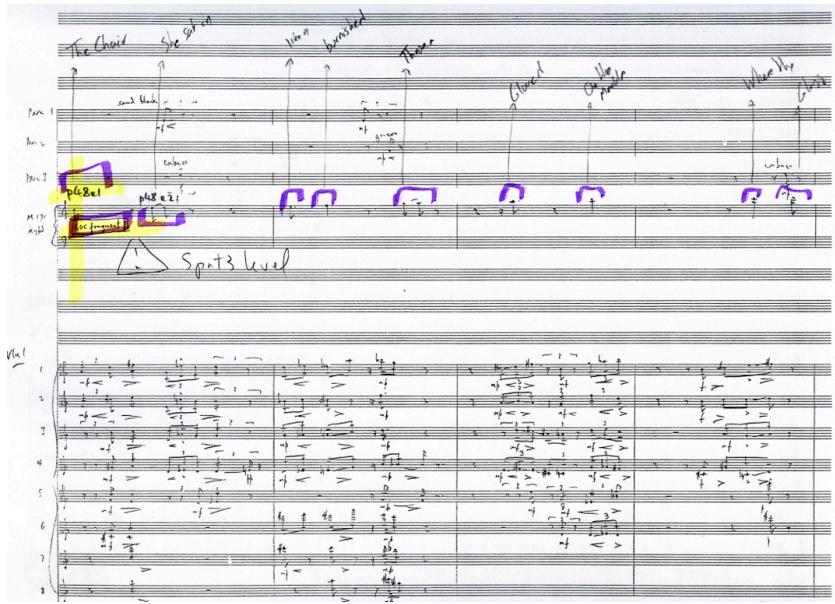
Harvey's « Speakings »

- Computer Assisted Composition
 - Speech to music transcription
 - OpenMusic + SuperVP



Phase II: Live electronics

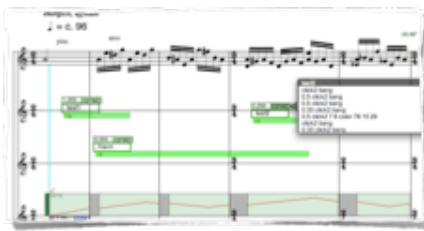
- Impose speech structures on orchestral sound
 - Real-time transformation
 - Synchronizing electronics with conductor/orchestra



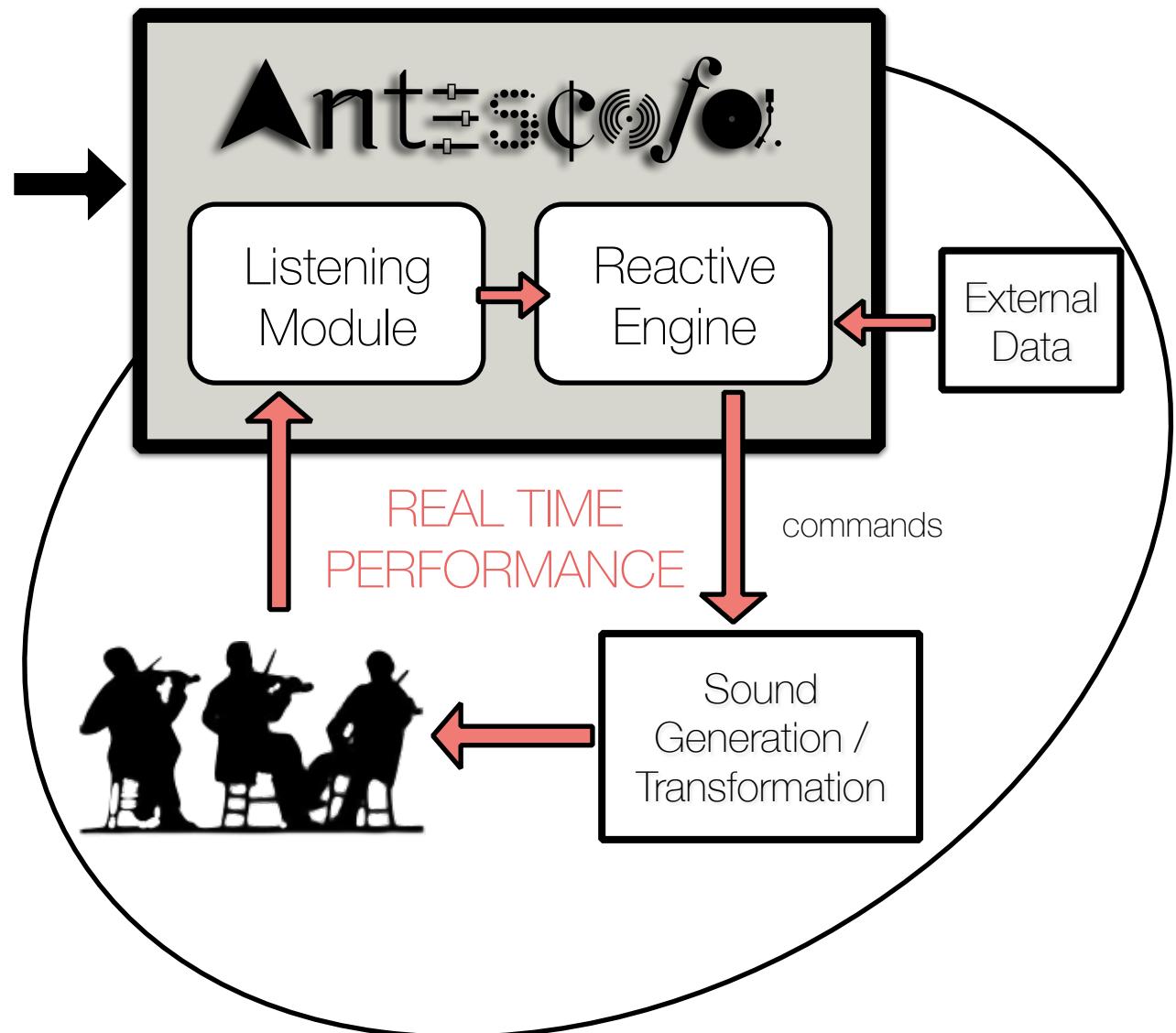
Speakings

- An excerpt
 - Live electronics
 - e.g. sound synthesis synchronized with orchestral event + tempo
 - Premier from August 2008 by BBC Scottish Orchestra in London (Royal Albert Hall)





COMPOSITION
TIME

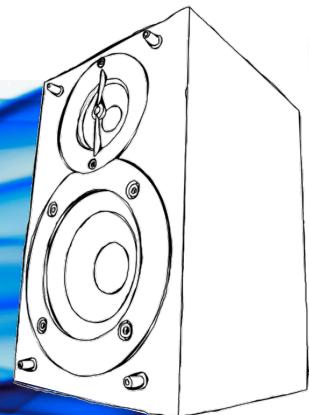


De la perception à l'action, du signal au symbole, du percept au concept et retour

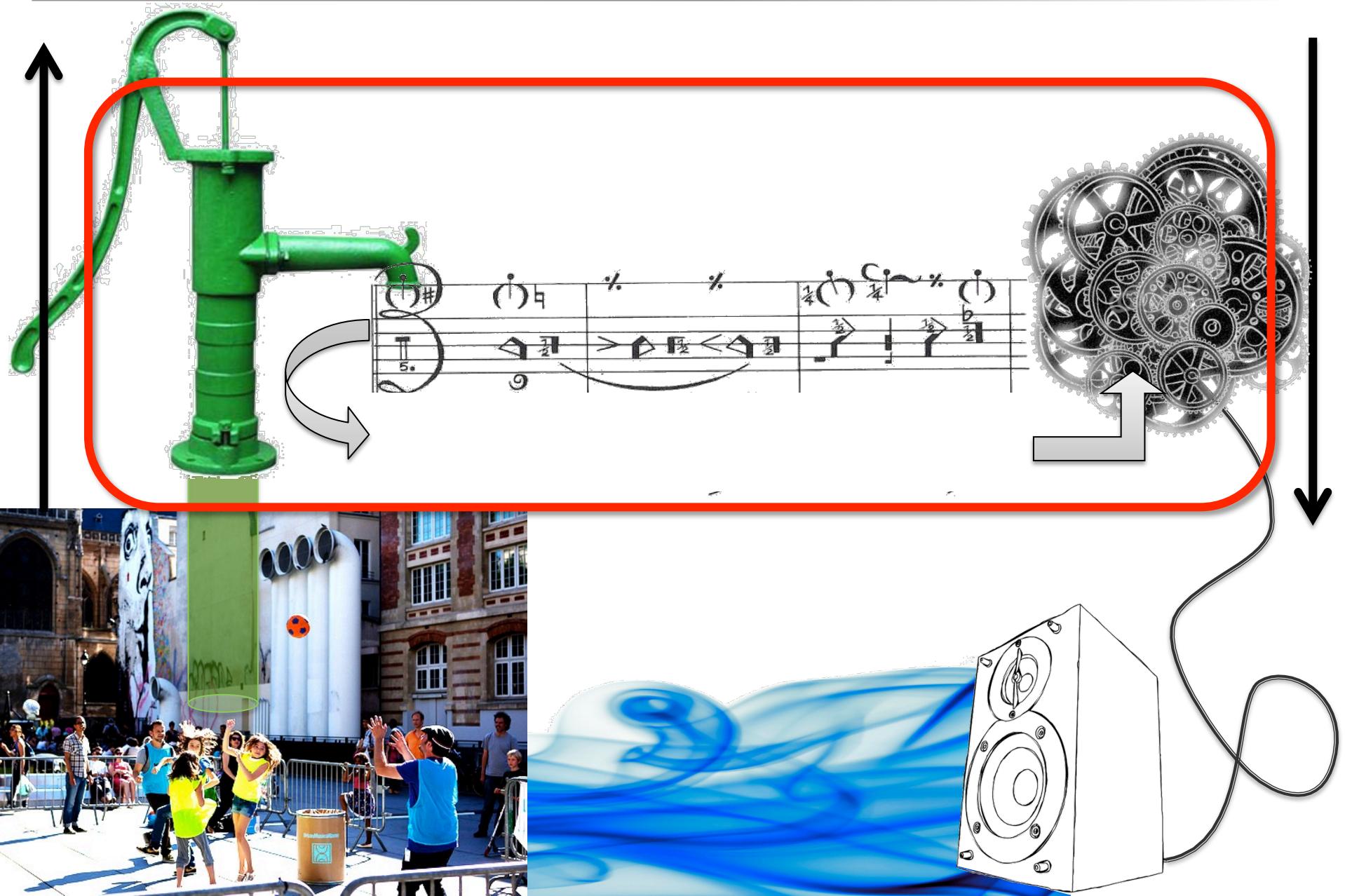
géometrie
de
l'information

Antescofo!

langages
synchrones
et systèmes réactifs

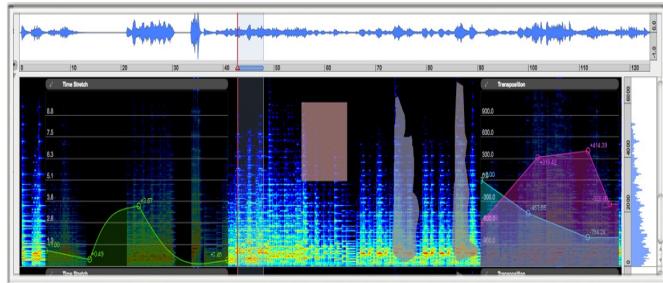


Human in the loop

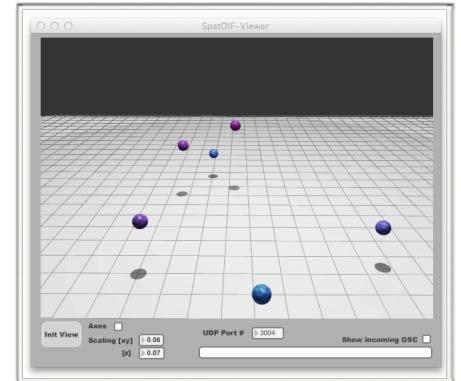




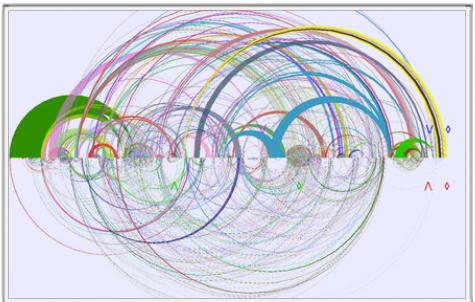
Gesture



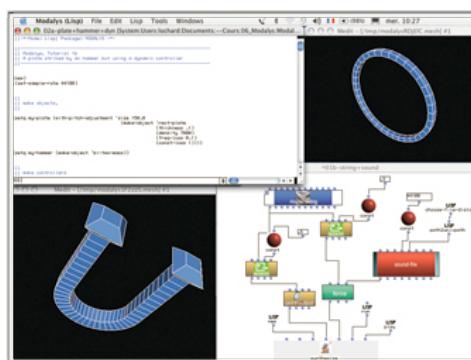
Audio Analysis/Synthesis



Spatialization



Improvisation

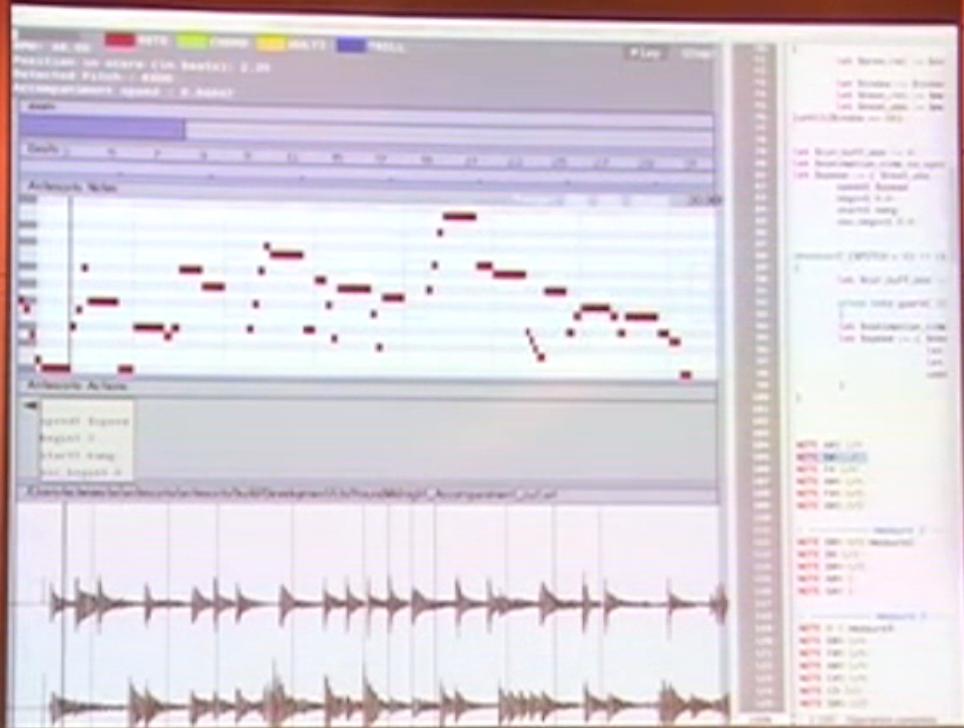


Physical Models

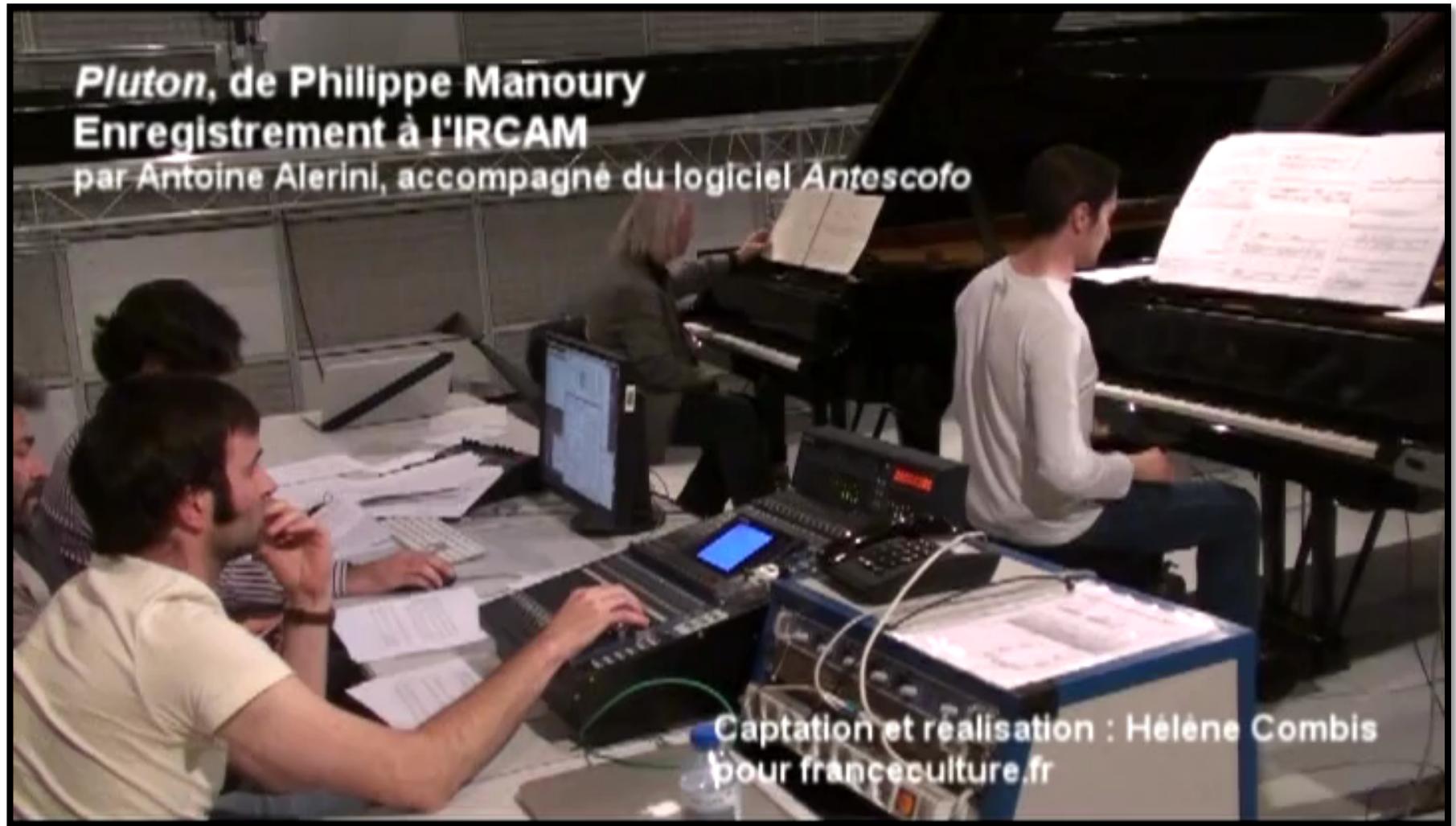


MIDI / Control

More than 40 Creations
New York Philharmonics, Chicago Symphony, Los Angeles Philharmonics, Berlin Philharmonics, BBC Orchestra...



Application : Musique mixte



Julia Blondeau, compositrice

- Quelques enjeux, quelques questions

- Un réel rapport interprète / électronique
- De nouvelles possibilités d'écriture de l'électronique
- Passer d'avancées quantitatives à qualitatives ?
- Dépasser la « simple » question de la synchronisation ?
- Une nouvelle manière d'aborder l'électronique grâce au temps-réel ?
- Une écriture plus fine des temps...



L'écart entre la partition et l'interprétation

1 = ca 66

(PLUNGER POTE)

HUM

PLAY

ETC

2

3

4

5

6

7

8

9

10

11

12

BEGIN WITH EFFORT TO REPEAT EXACTLY / CAPITALIZE ON SLIGHT IRREGULARITIES

BEGINNING HERE, MAKE PROGRESSIVELY MORE RADICAL DEVIATIONS FROM BASIC FORM

L'écart entre la partition et l'interprétation



- événement
- durée

Time-triggered versus event-driven architecture

[H. Kopetz, 1991]

1. event-triggered systems
activities triggered by the occurrence of a significant event

2. time triggered systems
activities triggered on point of a predefined temporal grid

Time triggered

- TT1: A system-wide time reference exists, with good-enough precision and accuracy. We shall refer to this time reference as the global clock. For single-processor systems the global clock can be the CPU clock itself.
- TT2: The execution duration of code driven by interrupts other than the timers is negligible. In other words, for timing analysis purposes, code execution is only triggered by timers synchronized on the global clock.
- TT3: System inputs are only read/sampled at timer triggering points.

Plan

1. Comprendre le temps de l'ordinateur
 1. un *temps stratifié* : rapprocher programme et partition
 2. la *matière du temps* : simultanéité, succession ↗ durée
 3. des *formes du temps* : cyclique, linéaire, DAG et ramifié
 4. retour sur la durée
2. Comprendre le temps du musicien
l'écart entre la partition et l'interprétation
3. Coordonner le temps de l'ordinateur et celui du musicien : *Antescofo*

Vivre dans le même temps

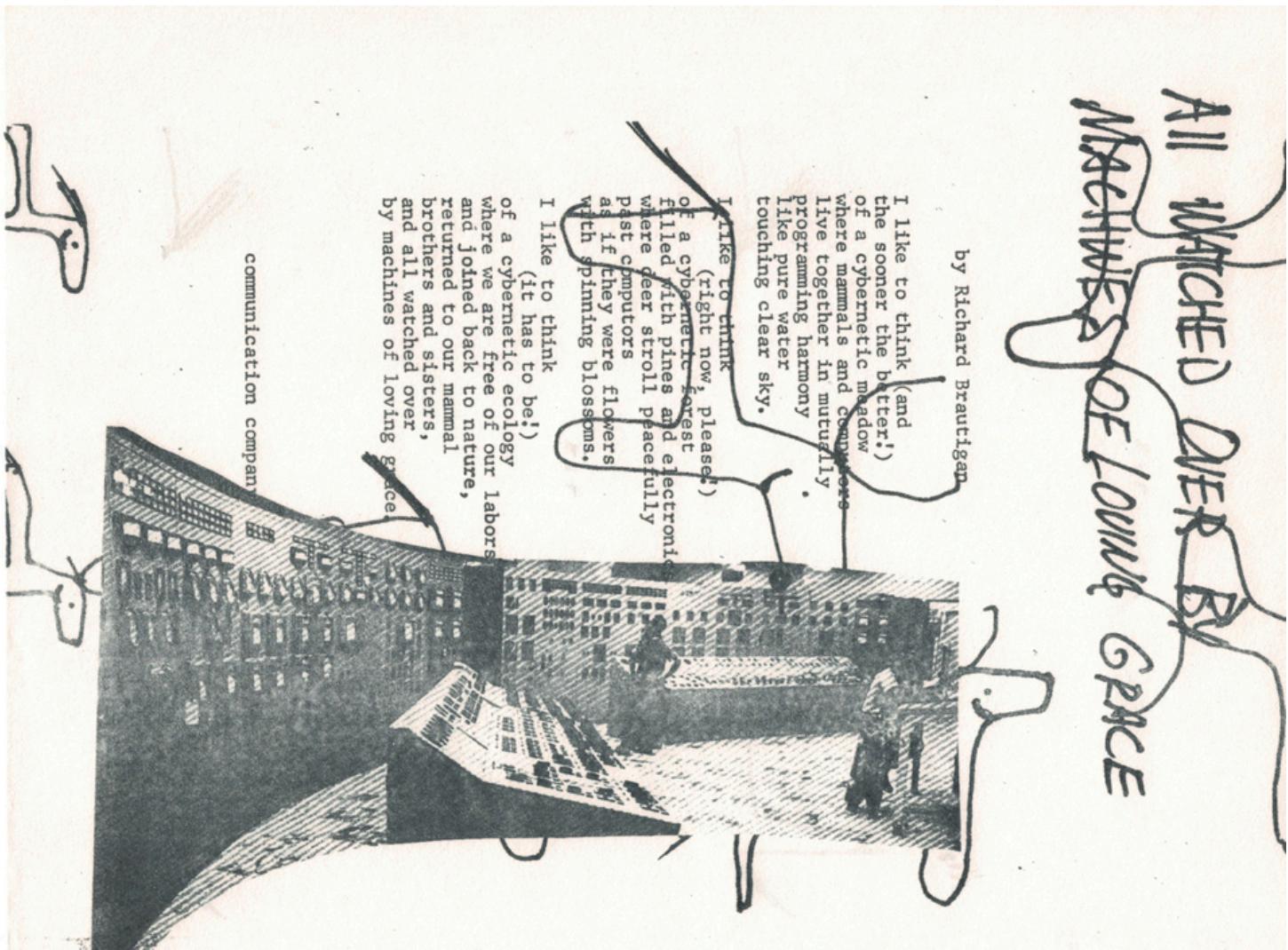
All WATCHED OVER By MACHINES OF LOVING GRACE

by Richard Brautigan

I like to think (and
the sooner the better!)
of a cybernetic meadow
where mammals and computers
live together in mutually
programming harmony
like pure water
touching clear sky.

I like to think
(right now, please!)
of a cybernetic forest
filled with pines and electronics
where deer stroll peacefully
past computers
as if they were flowers
with spinning blossoms.

I like to think
(it has to be!)
of a cybernetic ecology
where we are free of our labors
and joined back to nature,
returned to our mammal
brothers and sisters,
and all watched over
by machines of loving grace



du temps écrit au temps produit
est-ce que le temps informatique est différent du temps musical ?

UN TEMPS STRATIFIÉ

Rapprocher partition et programme ?

Partition	Programme
phase de composition	phase de spécification
phase d'interprétation	phase d'exécution
une écriture / sa dénotation	une expression / son évaluation
un compositeur	un programmeur
un (des) interprète(s)	une (des) machine(s)
...	...

Partition ≠ Programme

- G. Mazzola : un « geste surgelé »
- Adorno: l'état liquide de la musique
- « dégeler n'est pas réchauffer »

Moments du temps et temps logiques

interprétation

partition

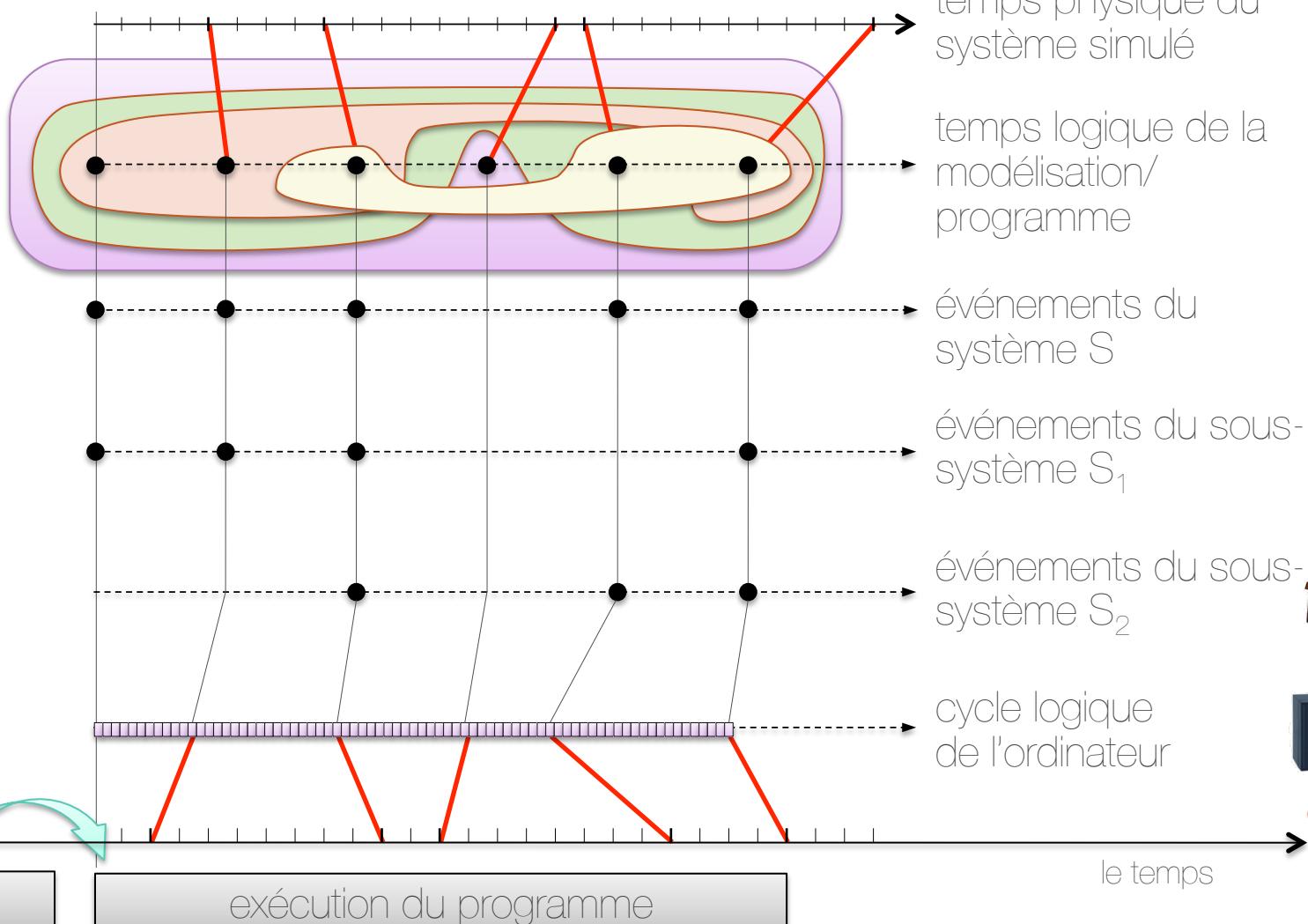
voix...

geste
instrumental

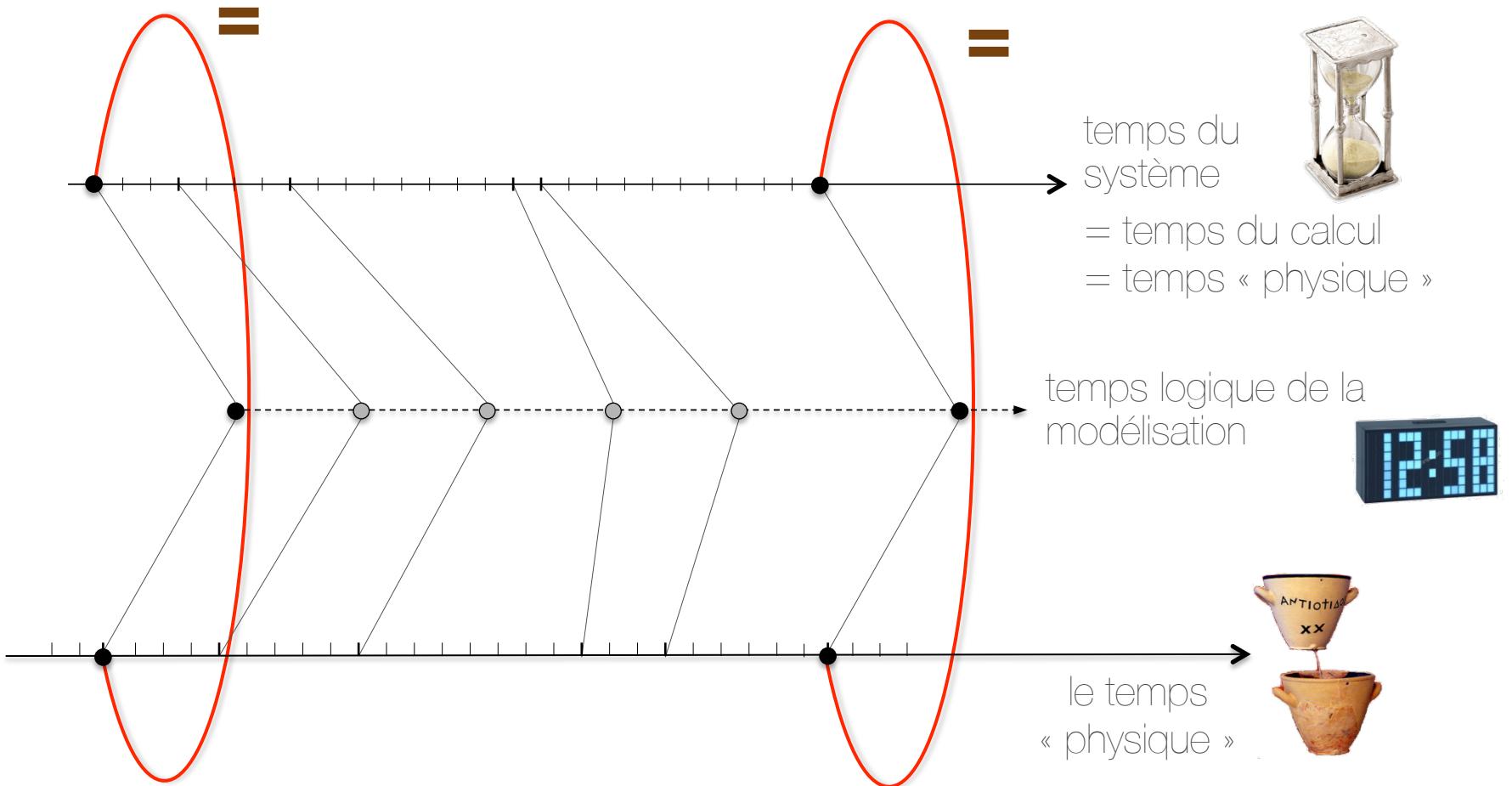
écriture

exécution du programme

le temps

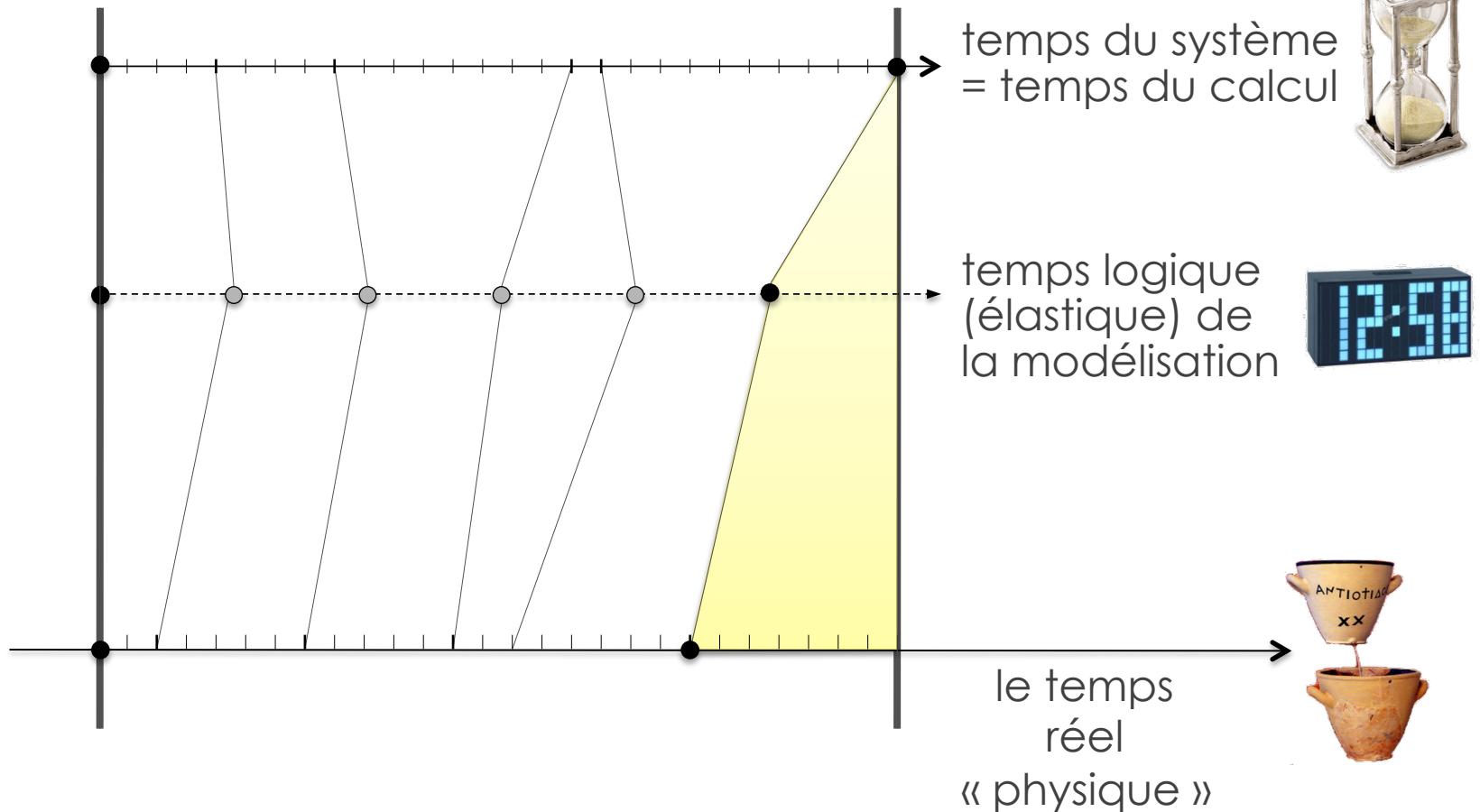


Hors temps et Temps réel

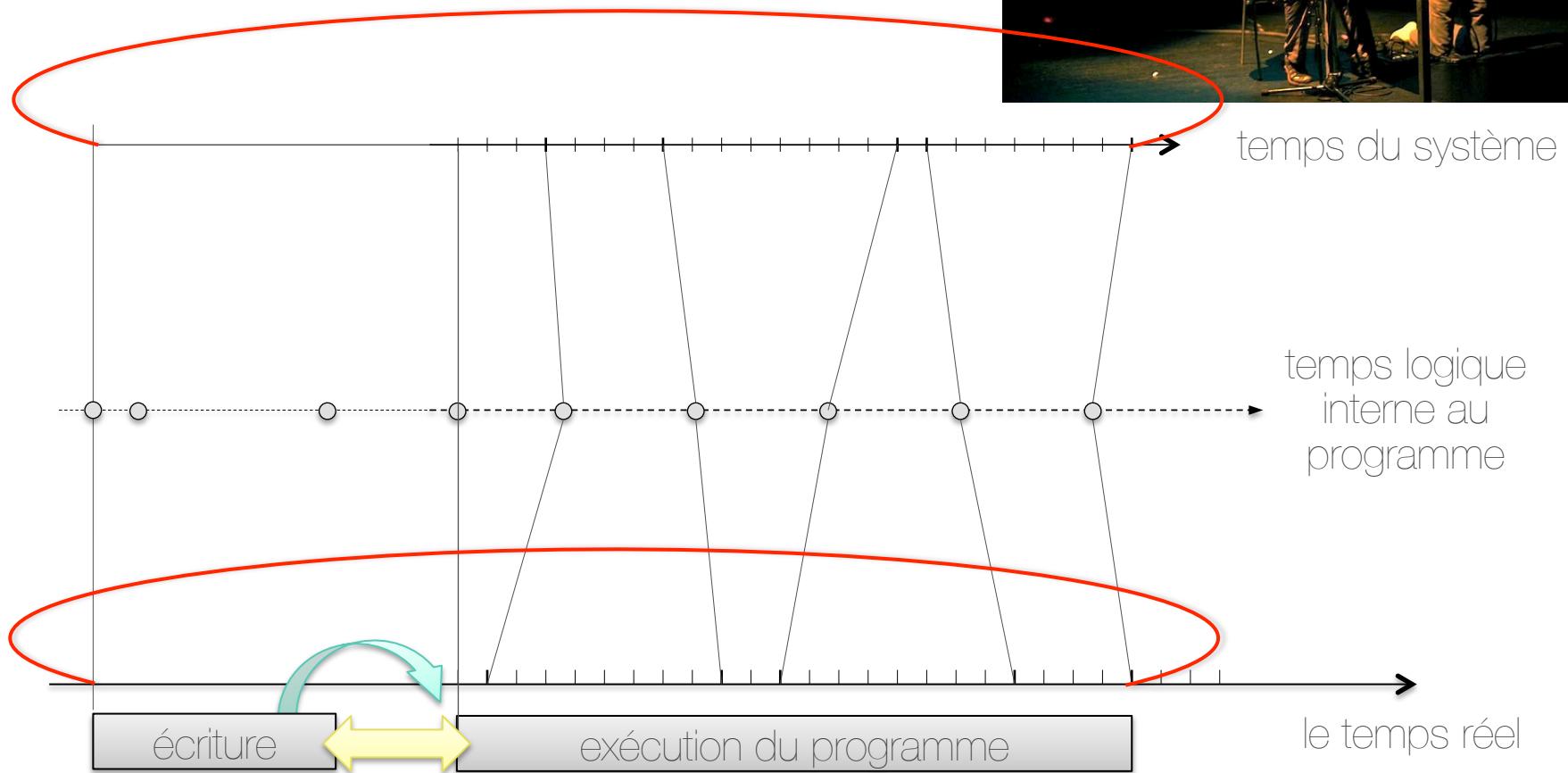


Prédire, à temps (exemple: météo, contrôle)

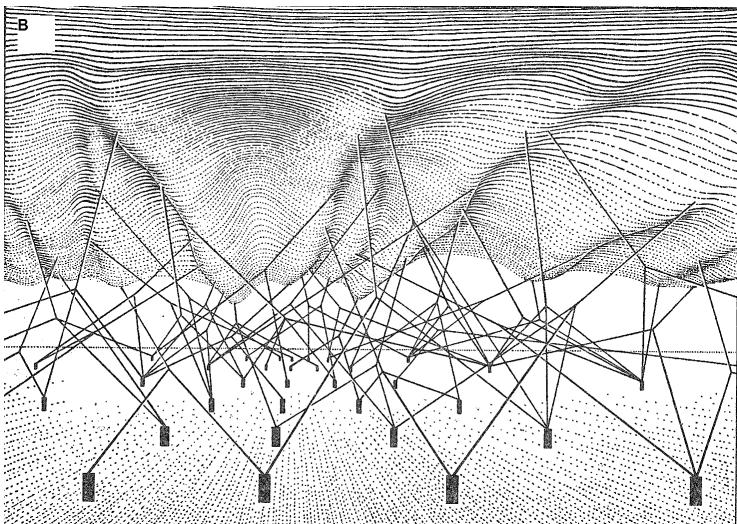
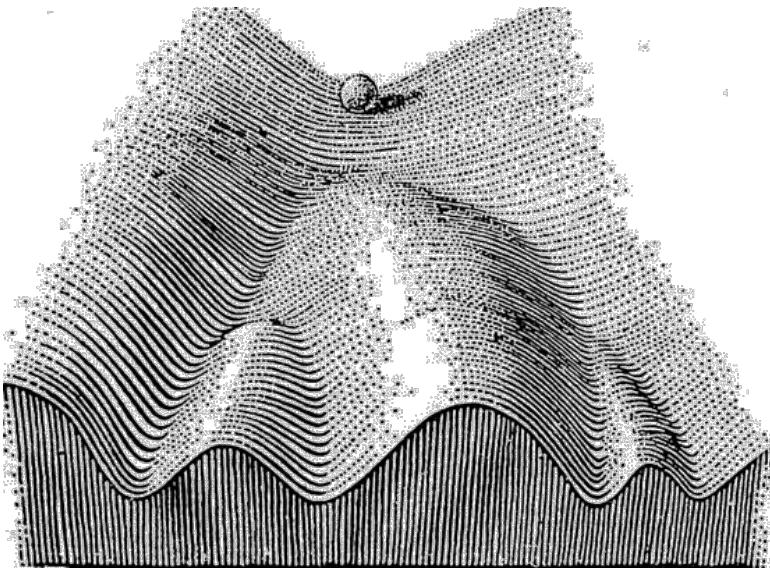
Système temporisé



Live coding...



Waddington : paysage épigénétique

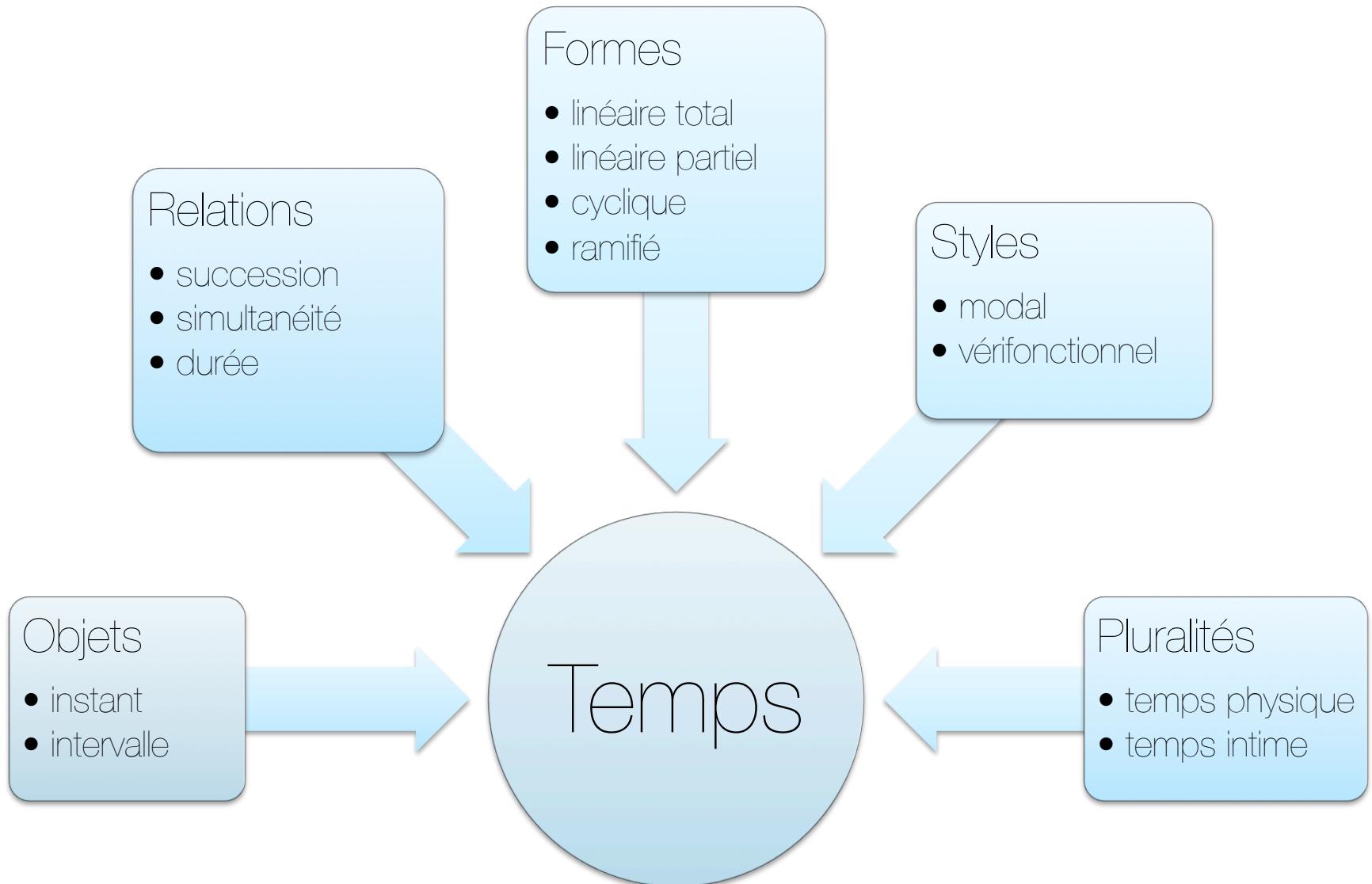


Link Li & Mhairi Towler, 2012

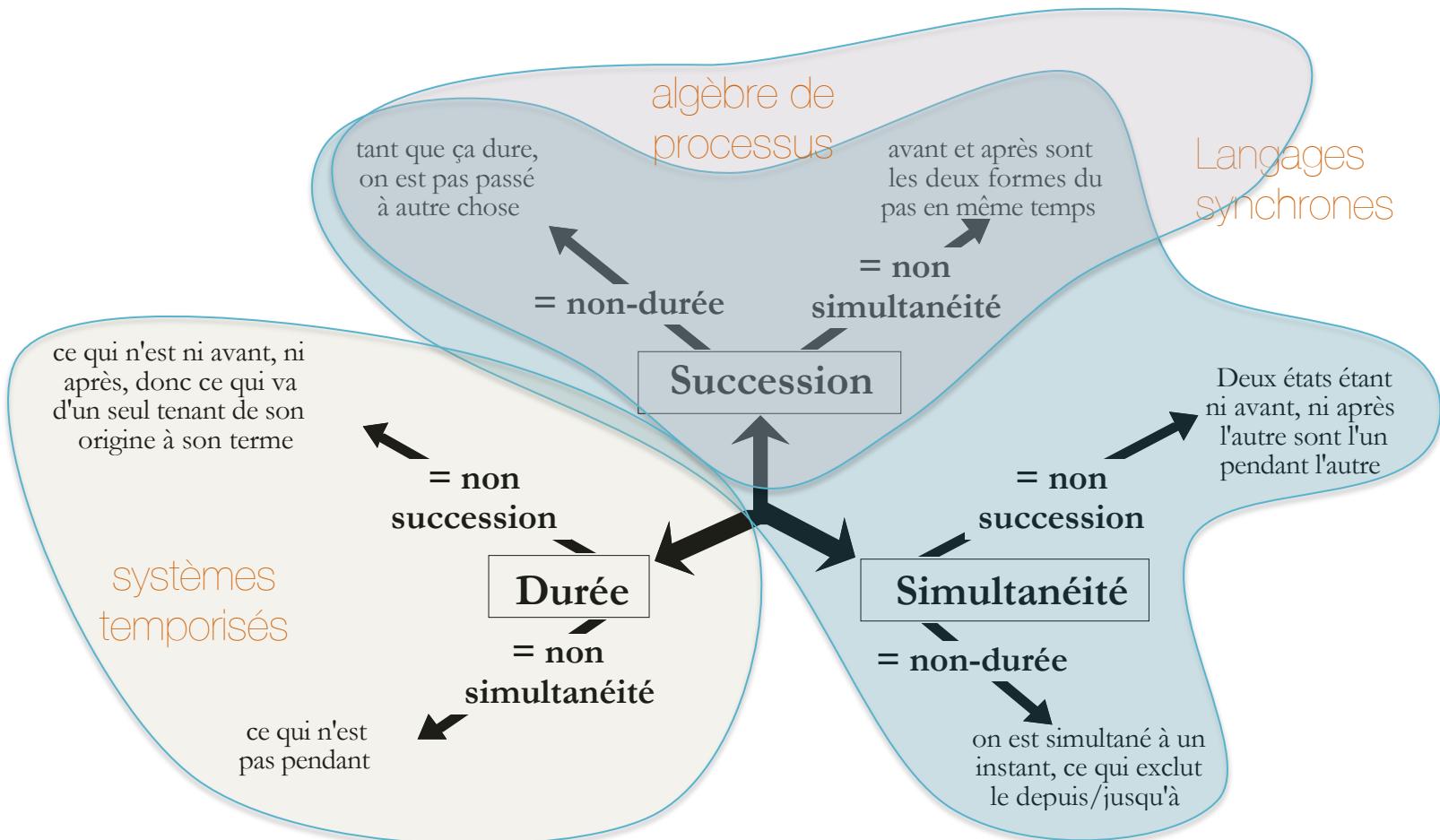
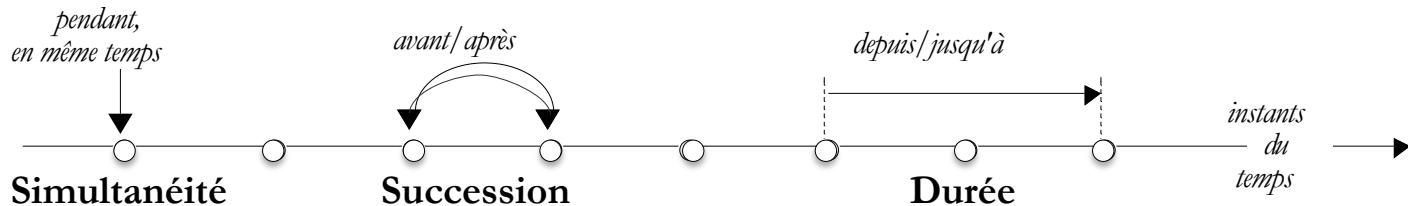


Simultanéité, succession & durée

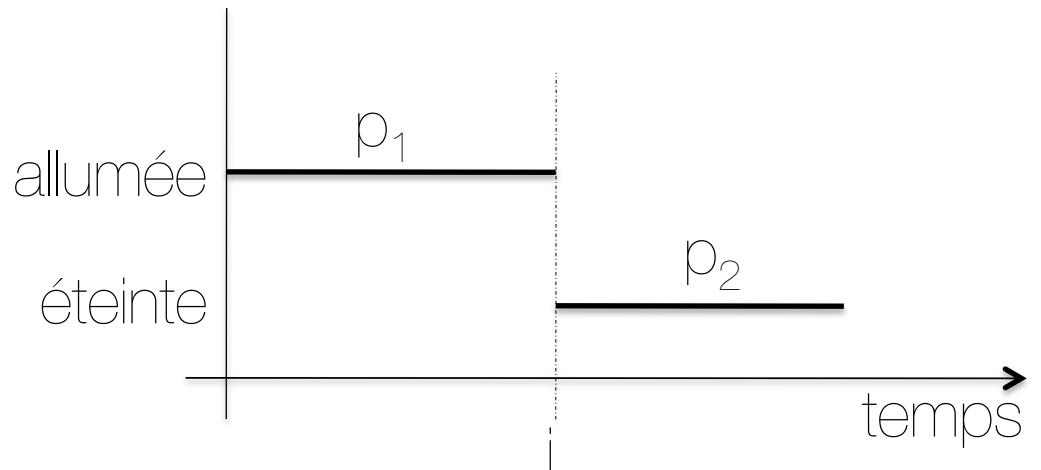
LA MATIÈRE DU TEMPS



Simultanéité, succession & durée



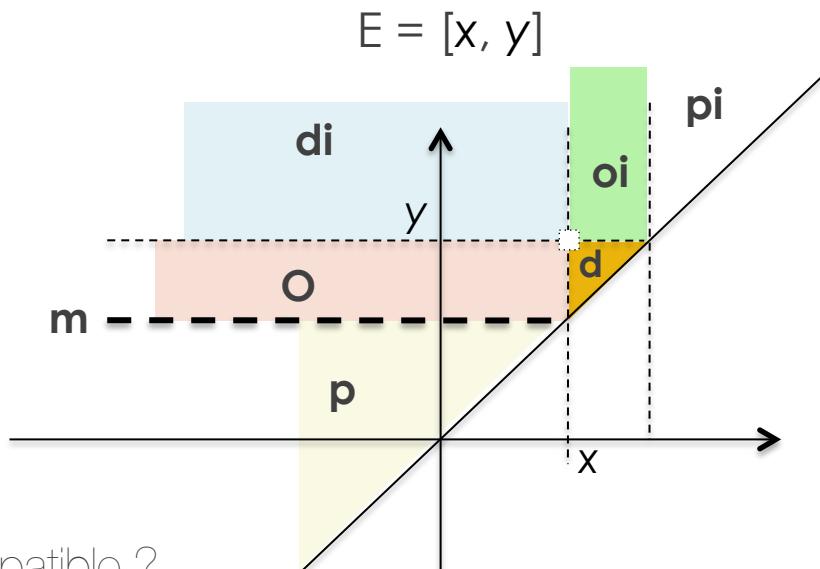
Durées et instants



Les relations de Allen (temps linéaire)



- Contraintes :
 - une séquence E doit se jouer avant A
 - une séquence A après R
- Est-ce possible ? quel scénarios ?
 - E, R, A
 - R, E, A
 - E=R, A
- Quelles relations peut-on rajouter en restant compatible ?

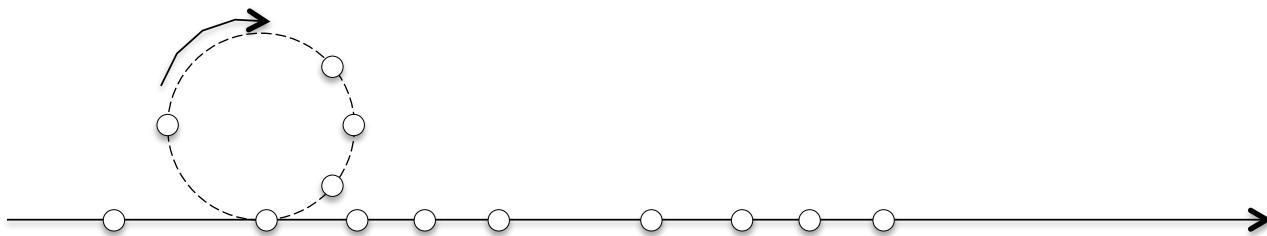


linéaire, cyclique, ramifié

LES FORMES DU TEMPS

Répétition, flot et ramification

- Cyclique

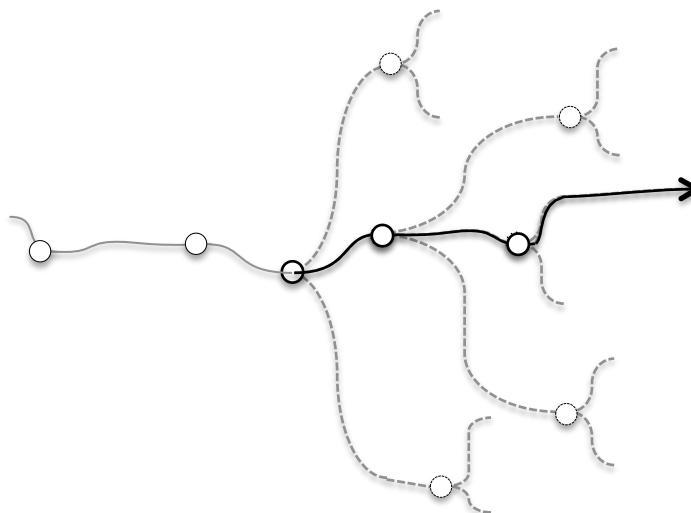


- Linéaire

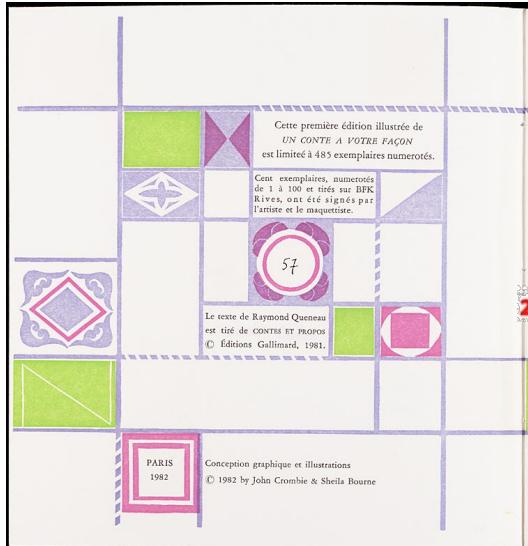
- une relation d'ordre totale (synchrone)
- une relation d'ordre partielle (asynchronisme)
systèmes distribués sans horloge globale, relativité)



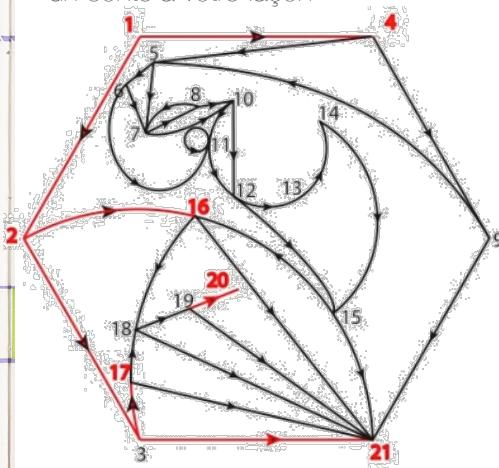
- Ramifié (branching)



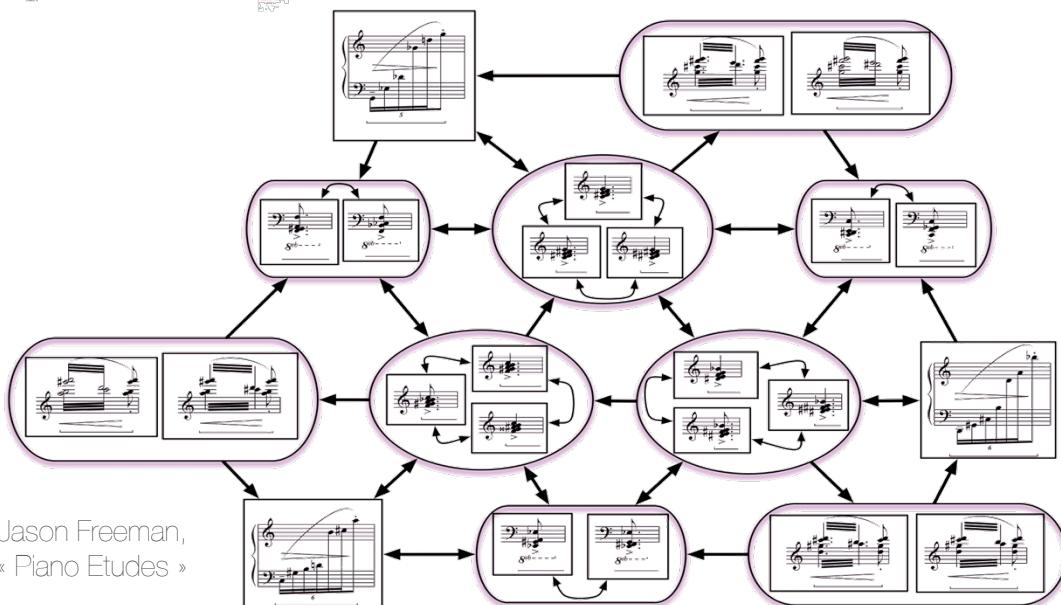
Oeuvres ouvertes extensionnelles



Raymond Queneau
un conte à votre façon



Raymond Queneau
1000 milliards de poèmes



Pluton : une œuvre ouverte intensionnelle

III Séquences d'échantillonnage

commencer ici ① ②

R1 ⑦ ⑧ ⑨

1

R2 ⑩ ⑪ ⑫

R3 ⑬ ⑭ ⑮ ⑯

R4 ⑰ ⑱ ⑲ ⑳

R5 ㉑ ㉒ ㉓ ㉔

R6 ㉕ ㉖ ㉗ ㉘

R7 ㉙ ㉚ ㉛ ㉜

R8 ㉝ ㉞ ㉟ ㉟

R9 ㉟ ㉟ ㉟ ㉟

Séquences fonctionnelles

(permutations)

E M

S

© 1993 Éditions DURAND
Paris, France

D & F 14653

L'ordre d'enchaînement des séquences de cette partie peut être variable. Les séquences **A**, **R**, **E**, **M** et **S** doivent être intercalées et non jouées dans groupe après groupe. Ex :

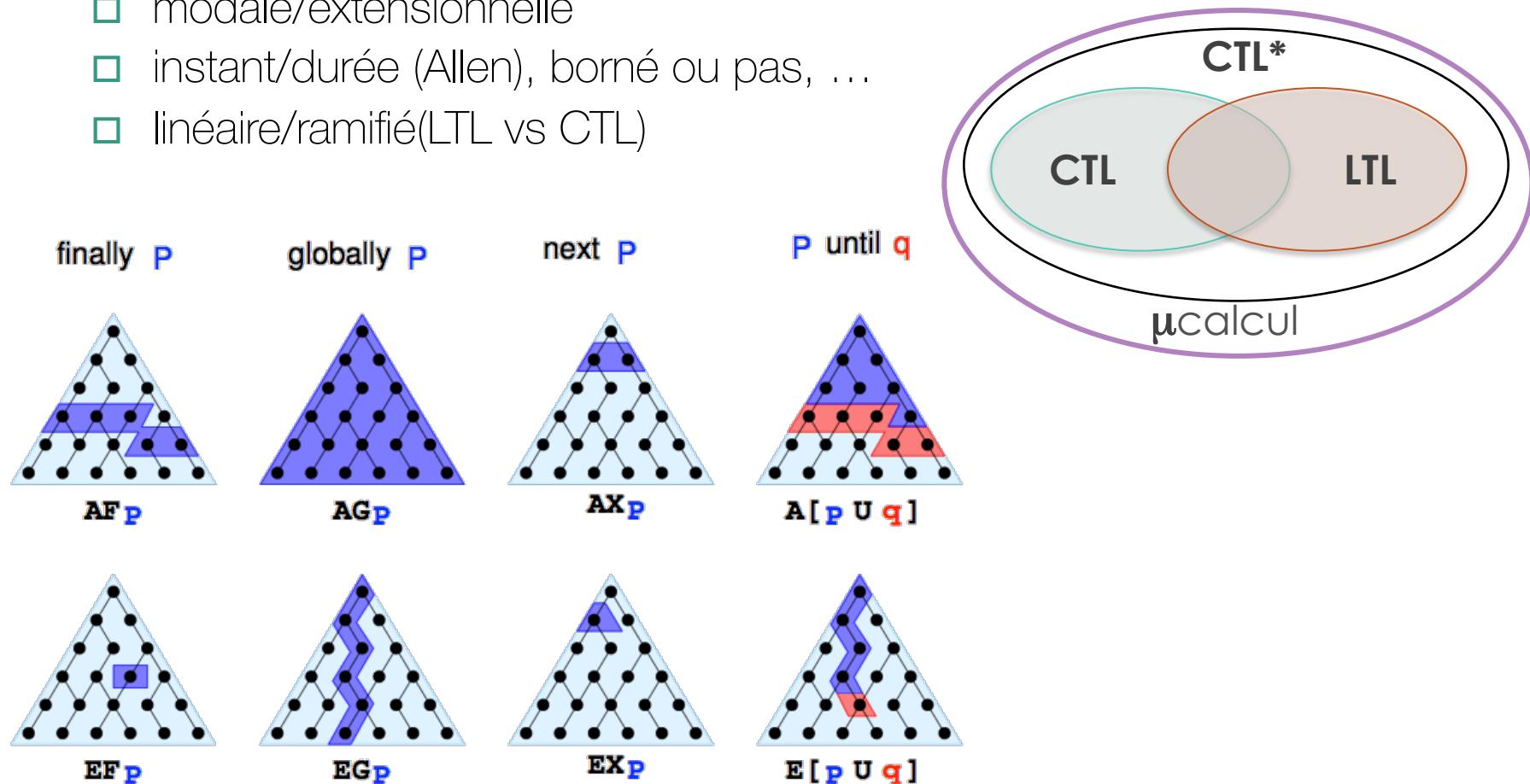


Les séquences **A** et **R** ne peuvent être jouées qu'une seule fois, les séquences **E**, **M** et **S** plusieurs fois. Les séquences **R** (pour Record) provoquent l'échantillonnage des sons du piano dans la 4x. L'ordre d'apparition de ces séquences doit *obligatoirement* être celui inscrit dans la partition (de **R1** à **R9**) et ne peut être modifié.

Les séquences fonctionnelles **E**, **M** et **S** comportent chacune, une note d'ouverture et une note de fermeture. Il est indispensable de jouer cette dernière si l'on désire enchaîner une des séquences **R** ou **A**, mais cette précaution n'est plus obligatoire si l'on enchaîne **E**, **M** ou **S**.
La séquence **E** (pour Échelle) permet de faire une compression de l'ambitus de ce qui sera joué par la 4x

Raisonnements temporels

- Vérifier des propriétés sur la dynamique du système
- Choisir son formalisme suivant l'usage le plus approprié
 - modale/extensionnelle
 - instant/durée (Allen), borné ou pas, ...
 - linéaire/ramifié(LTL vs CTL)

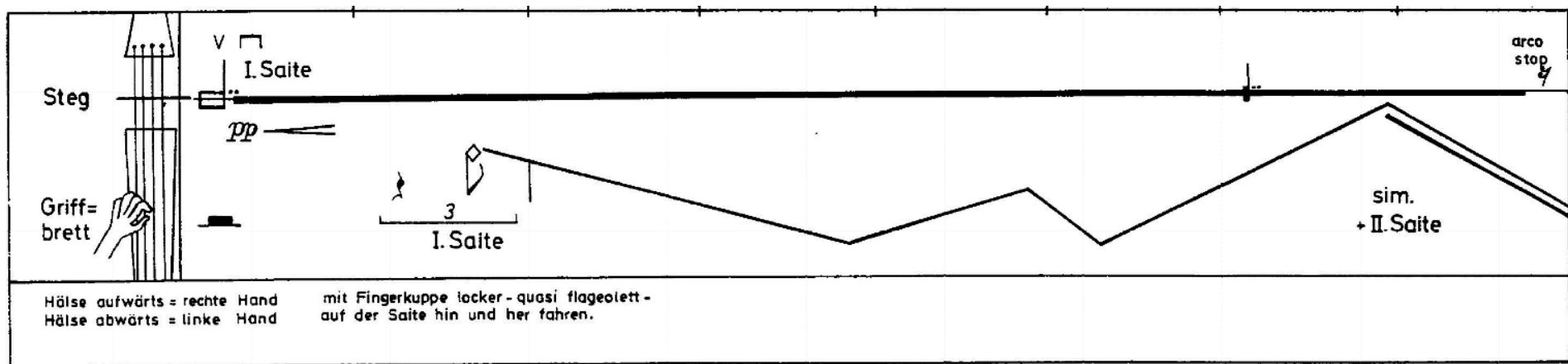


A-series / B-series
Logique Modale / Logique Vérifonctionnelle,
Temps réel / Hors temps

STYLES A & STYLE B

Déclaratif / procédural

- Noter ce que doit faire le musicien
- Noter le résultat



John Mc Taggart (1866 – 1925) : A-propriétés et B-propriétés

Arthur Prior (1914-1969) : logique modale temporelle

- **A-series** : caractérisation d'un événement e comme passé, présent ou futur (*prédicat unaire*).
Le temps est un flux, un passage.
- **B-series** : caractérisation d'un événement e comme avant e' ou après e' (*prédicat binaire*).
Pas de notion de présent, mais un temps « spatialisé », statique
- Affinité entre les propriétés de **type A** et le style **modal** (intensionnel) des logiques temporelles
- Affinité entre les propriétés de type **B** et le style premier ordre (**vérifonctionnel**, extensionnel, explicitement indexé) des logiques temporelles
- On peut retrouver les deux styles dans les langages
 - Le style « modal » est-il adapté à la performance ?
 - Le style « spatial » est-il adapté à la composition ?

3 DEC

Sketch — ~~7 Nov 2012~~

SCORE in C

CHRISTOPHER TRAPANI

[X] markers to be placed by Antescofo [X] possible convergence points

Exemple:

Une esquisse de Christopher Trapani inspirée par les canons rythmiques de Colon Nancarrow

1 -> A 3/5

ca. $\text{♩} = 108$

Clarinet in B \flat

gliss. 'pizz.' — [1B]

1B -> A point d'entrée défini

(A) [2] — Cl. sub. 3 — (B) [3] —

3 -> D 7/11

Cl. sfz f pp — (C) [4] — ppp —

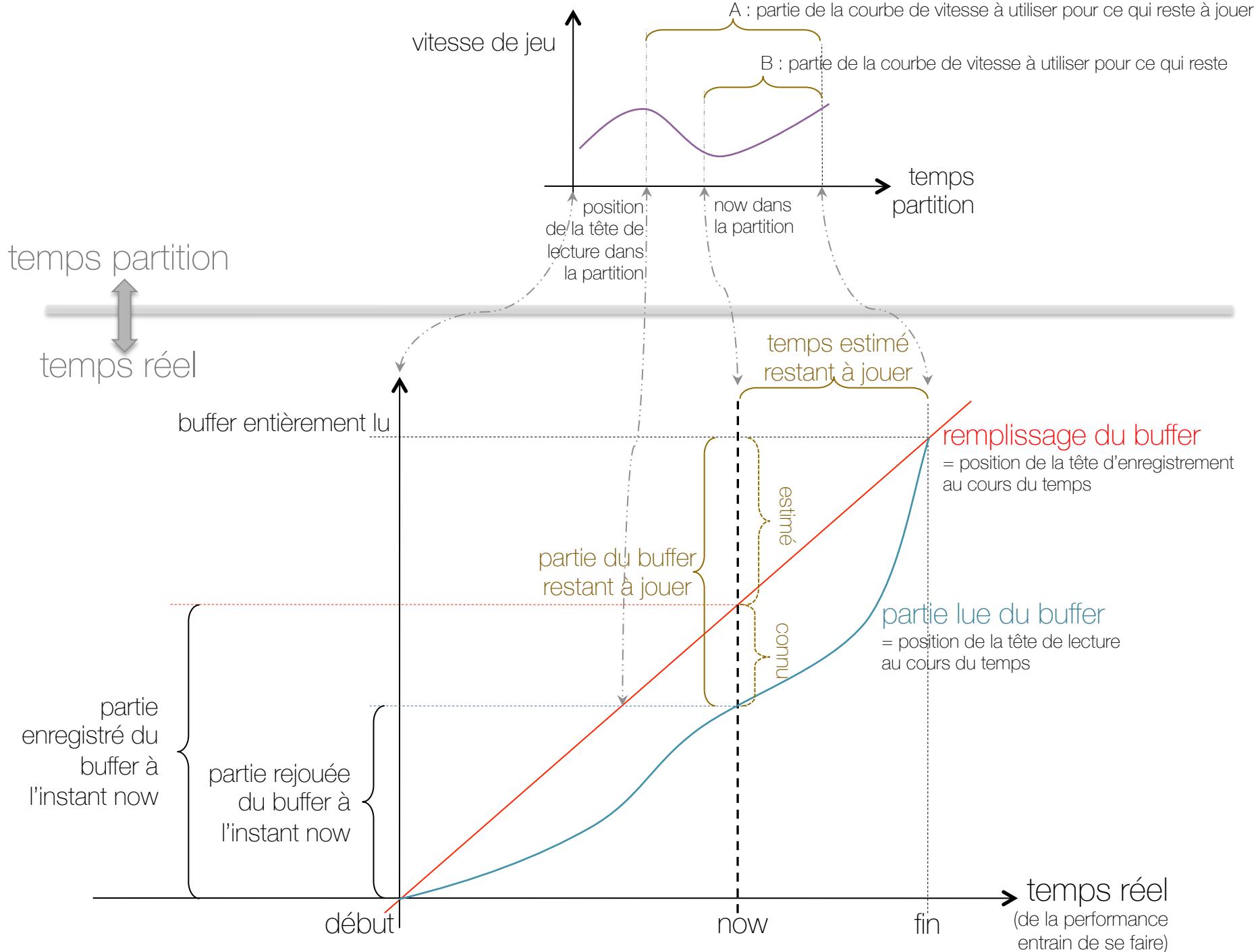
2 -> B point d'entrée défini

(D) [5] — Cl. sfz mf pp —

6 — (E) [6] — tongue-slap 3 —

7 — Cl. sub. mf — p > pp — mf — pp —

19 —



Sketch — ~~7 Nov 2012~~ 3 DEC

SCORE in C

CHRISTOPHER TRAPANI

[X] markers to be placed by Antescofo [X] possible convergence points

1 > A 3/5

ca. $\text{♩} = 108$

Clarinet in B \flat

gliss. 'pizz.'

1B

A [2]

Cl. sub. 3 p 7 mf mp pp sfz f

B [3]

1B > A point d'entrée défini

C [4] **3 > D** 7/11

Cl. sfz f pp 6 7 ppp

D [5]

Cl. sfz mf pp 3 n pp p 5

E [6]

Cl. sfz mf pp 3 tongue-slap 6 6 5 sub. mf > p > pp mf pp

2 > B point d'entrée défini

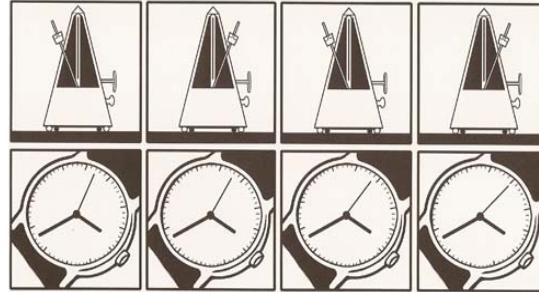
1B -> A point d'entrée défini

gloss.

7

Cl. p 3 pp 7 p 3 pp p 5 pp n

This musical score for Clarinet in B-flat is annotated with numerous markings from the Antescofo system. It includes dynamic instructions like 'f', 'sfz f', 'pp', 'mf', 'p', and 'n'. Performance techniques such as 'gliss.', 'pizz.', 'tongue-slap', and 'sub.' are indicated. The score is divided into sections labeled 1 through 7. Convergence points are marked with boxes and arrows, with some being defined entry points (A, B, C, D, E) and others being possible convergence points. The score is set in common time, with specific tempo markings like 'ca. ♩ = 108' and time signatures like '7/11'.



une durée à la Bergson

PLURALITÉS DU TEMPS

Le temps intérieur pour Zimmermann

Nous appellerons durée temporelle effective la portion de temps que nécessite une œuvre musicale pour son exécution. Cette durée n'est cependant pas dans son extension une grandeur constante dans le sens où elle resterait la même à chaque exécution. Autrement dit, les conditions toujours changeantes de l'exécution musicale, même si elles sont la plupart du temps d'ordre minimal, occasionnent des durées d'exécution variables pour une même composition, alors qu'au contraire les proportions de toutes les relations métriques, rythmiques et donc temporelles demeurent inchangées à l'intérieur de la durée temporelle (effective) qui, elle, varie. De cette manière, le temps inhérent à une composition sera doublement organisé : d'une part par le choix d'une certaine unité de temps (effective) qui a la fonction de tempo musical, d'autre part, par le choix d'une certaine unité de temps (intérieure) qui ordonne les rapports entre Intervalle et Temps. *Les unités de temps, intérieures aussi bien qu'effectives, sont définies par la conscience intérieure du temps musical à laquelle, en ce sens, nous devons attribuer la fonction régulatrice de l'expérience et de la perception du temps en musique.*

Deux temps ?



durée



événement



Une seconde par seconde

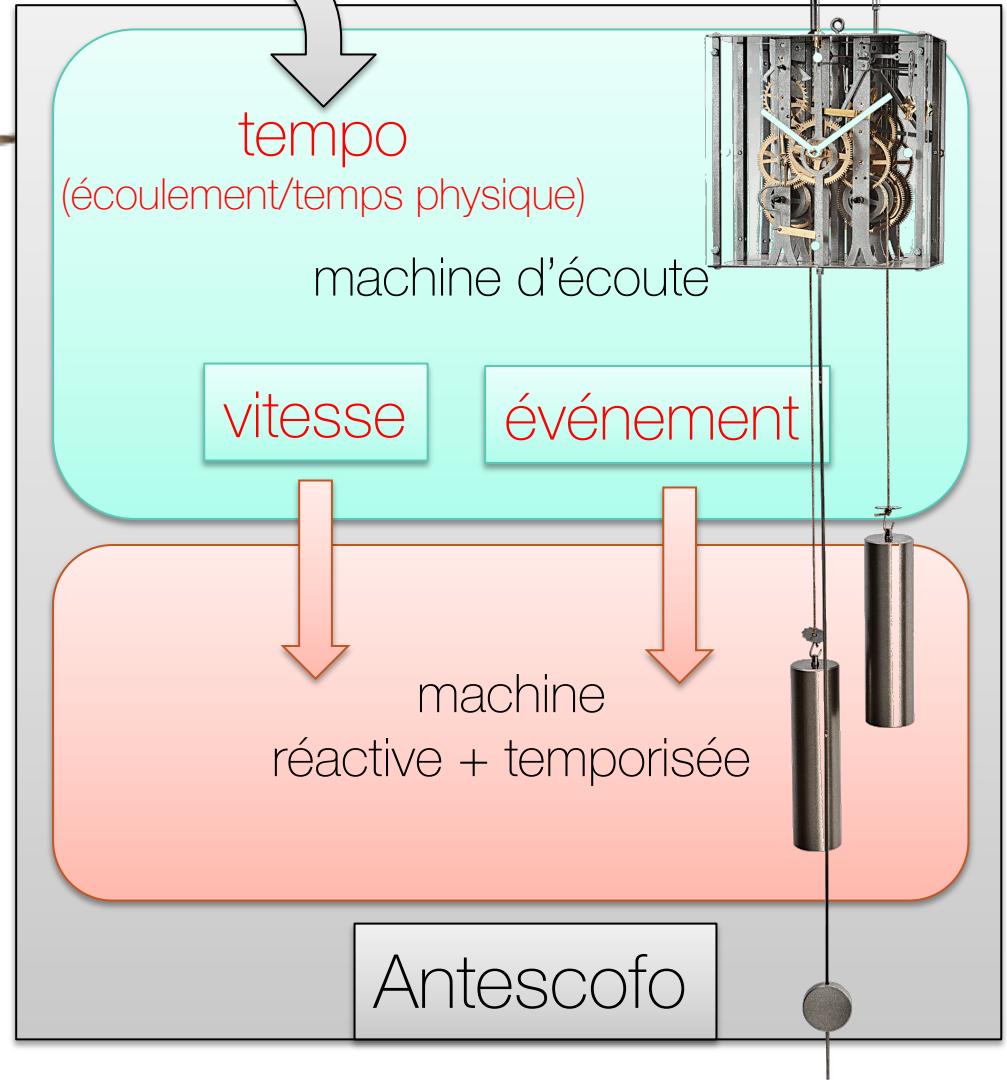
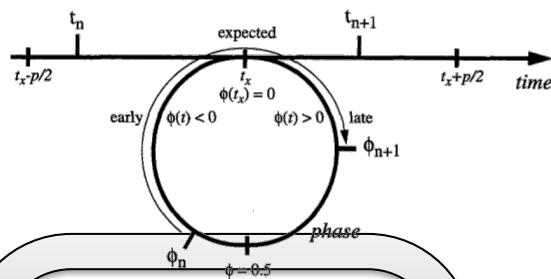


Subordonner le temps objectif au temps subjectif *au lieu du contraire*

- La durée n'est pas réductible aux instants
 - diviser une durée par deux
 - accelerando
 - le phrasé musical (ex. rubato)
- la partition est écrite / tempo, pas au temps physique
- le "taux de change" (varie dans le temps et)
n'est connu qu'après
- parce que cela permet une interaction musicale
entre instrumentistes et machine

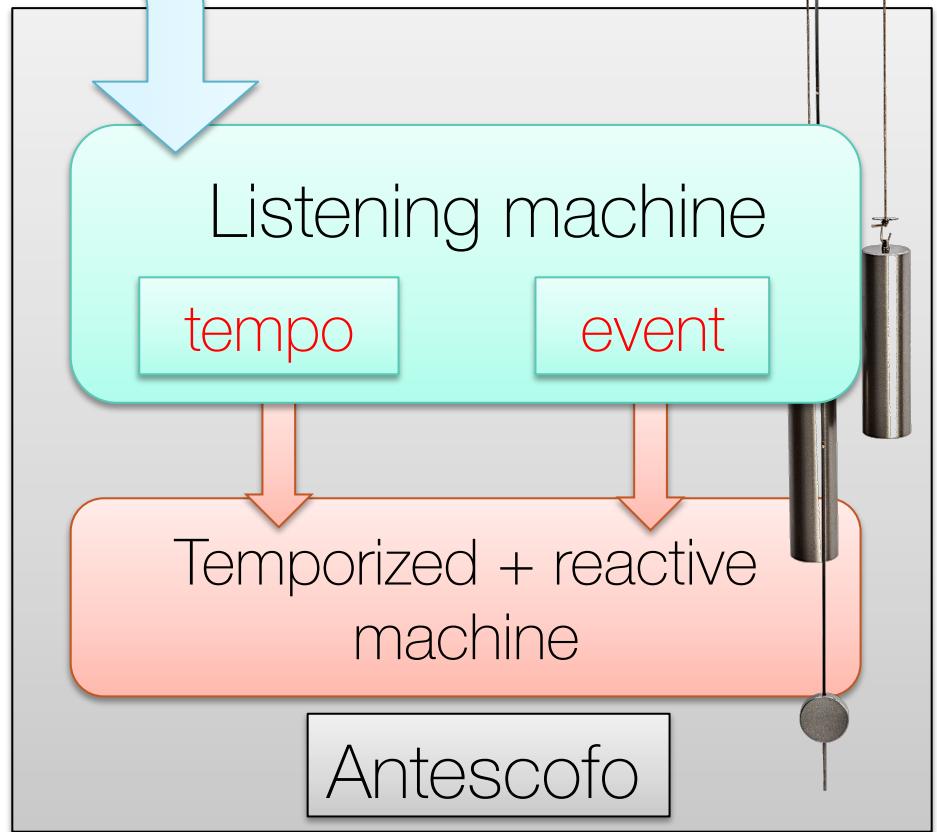
Extraction du tempo et sympathie des horloges







Event



le langage

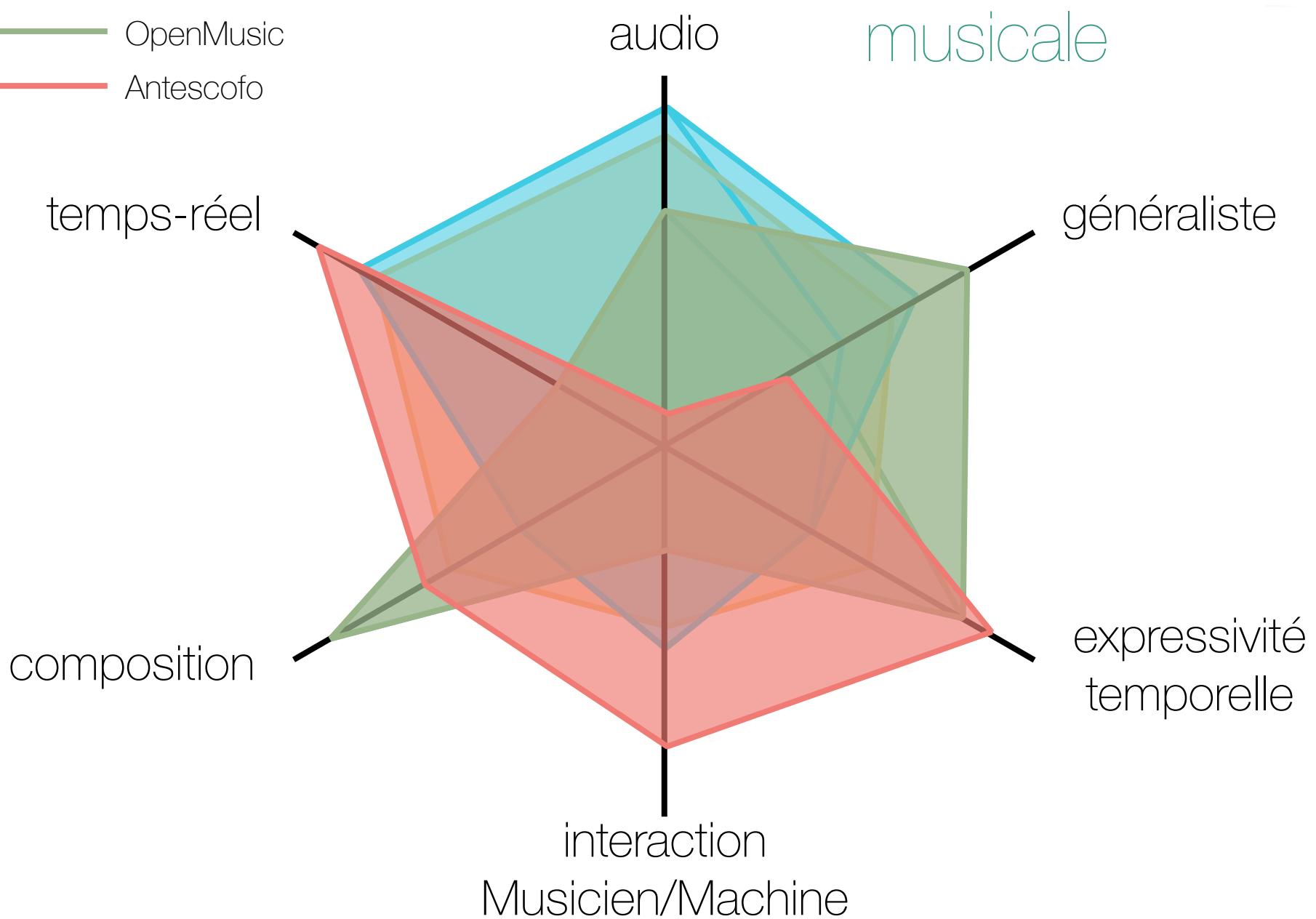
ANTESCOFO

le langage d'Antescofo

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

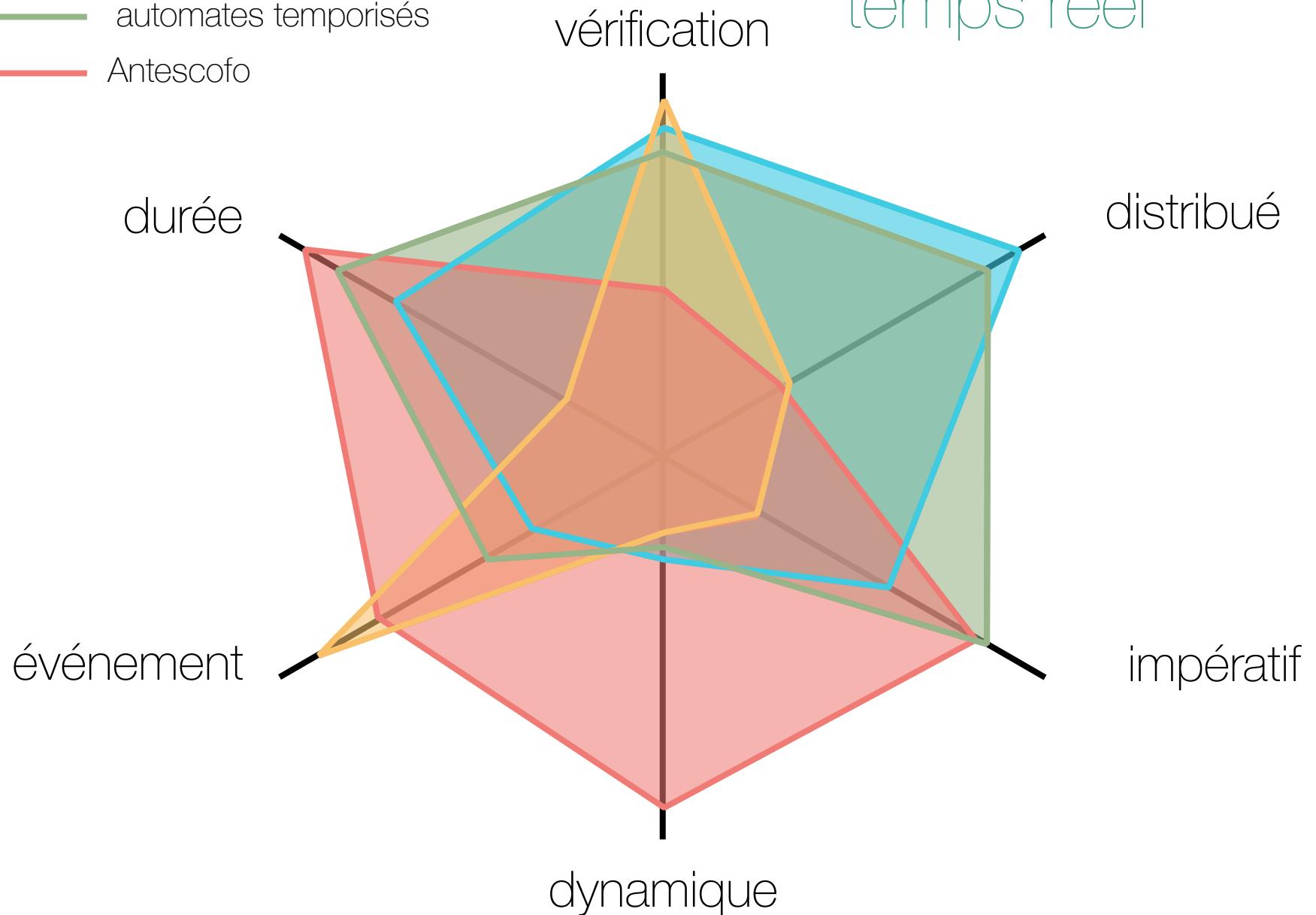
Informatique musicale

- SuperCollider
- MaxMSP
- OpenMusic
- Antescofo



Informatique temps réel

- Lustre
- giotto
- automates temporisés
- Antescofo



L'écriture du temps

- simultanéité
 - succession
 - répétition
- }
- temps strié
-
- durée (e.g., finir en même temps)
 - mouvements (processus continu)
- }
- temps lisse
-
- la hiérarchie
 - l'expression
 - impérative, équationnelle, par contrainte...
- }
- structure

Syntaxe

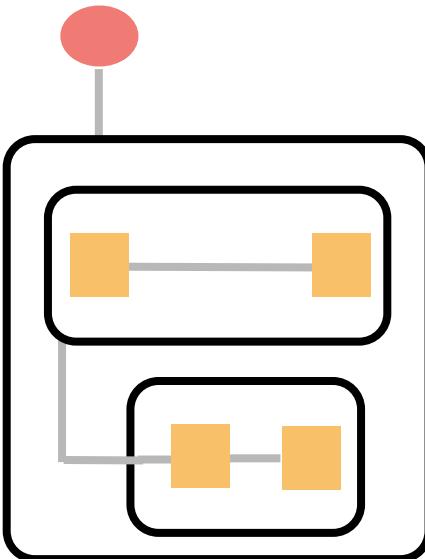
● événements:

NOTE 60 2.0

■ actions atomiques:

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

□ actions structurées:



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

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

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

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

Duration & delay

- in seconds or
- in beats
 - global or
 - local tempo

NOTE C4 2.0

```
(1/4 + 1/8) action1  
1/4 s           action2
```

NOTE D4 1.0

```
group @tempo = ( 2*$RT_TEMPO + sint($t) )  
{  
    action_group1  
    1/2 action_group2  
    $v  action_group3  
}
```

Expressions

■ Valeurs

`int, float, bool, string, symbol, tab, map,
interpolated map, fonctions, processus...`

■ Opérateurs et fonctions prédéfinies

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

■ Variables

□ variables systèmes

`$RT_TEMPO, $NOW, $RNOW, $TEMPO, $PITCH, etc.`

□ historique

`[3#] :$x`

`[3] :$x`

`[3s] :$x`

<code>\$v</code>	.	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)`

Group

Note C3 1.0

Group G1

{



}

Group G2

{



}

0.5 Group G3

{



}

0.5 Group G4

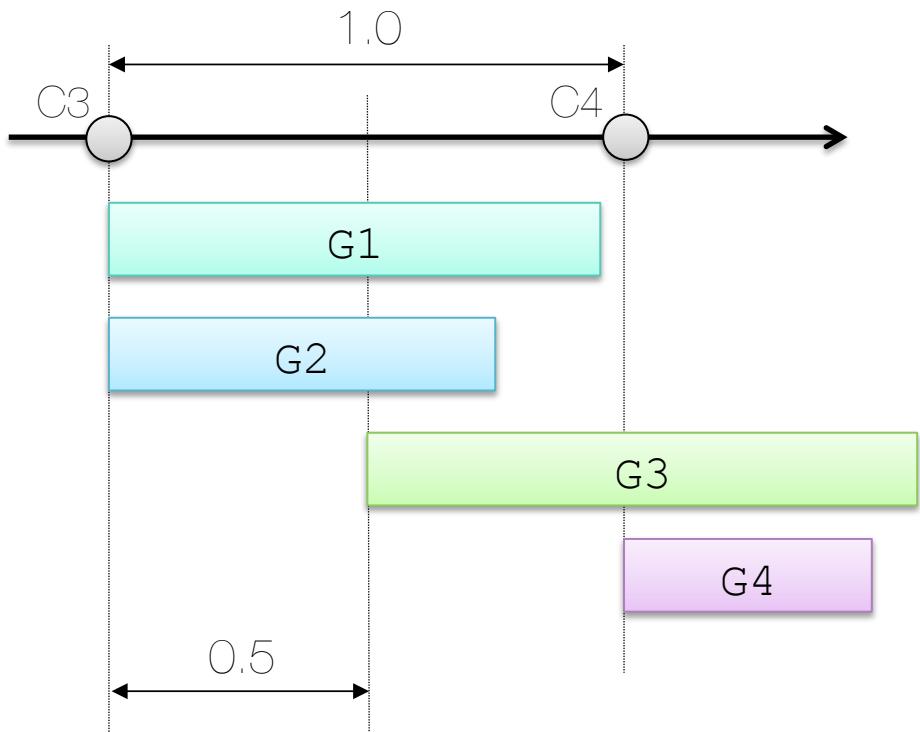
{



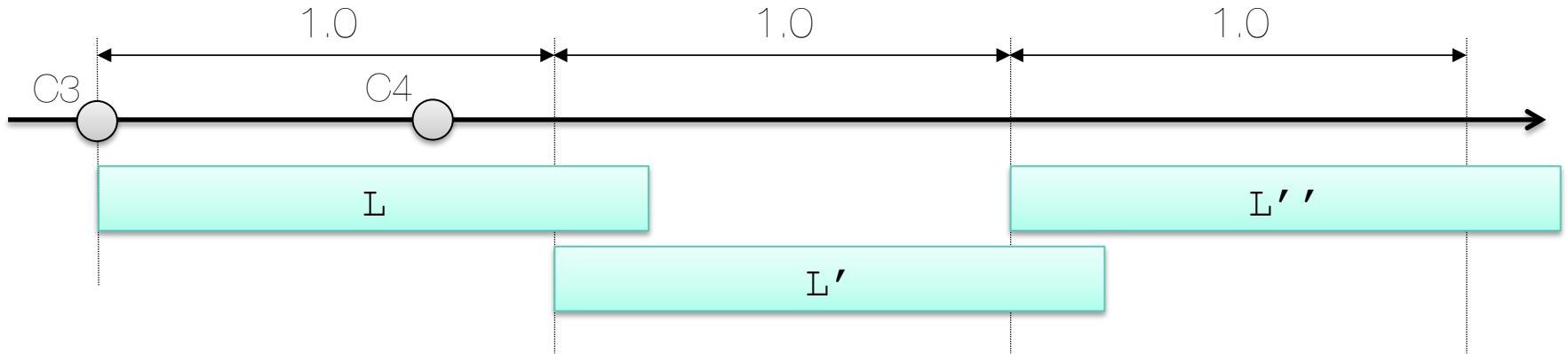
}

Note C4 1.5

...



Loop



Note C3 2/3

Loop L 1.0

{

L

}

Note C4 1.5

...

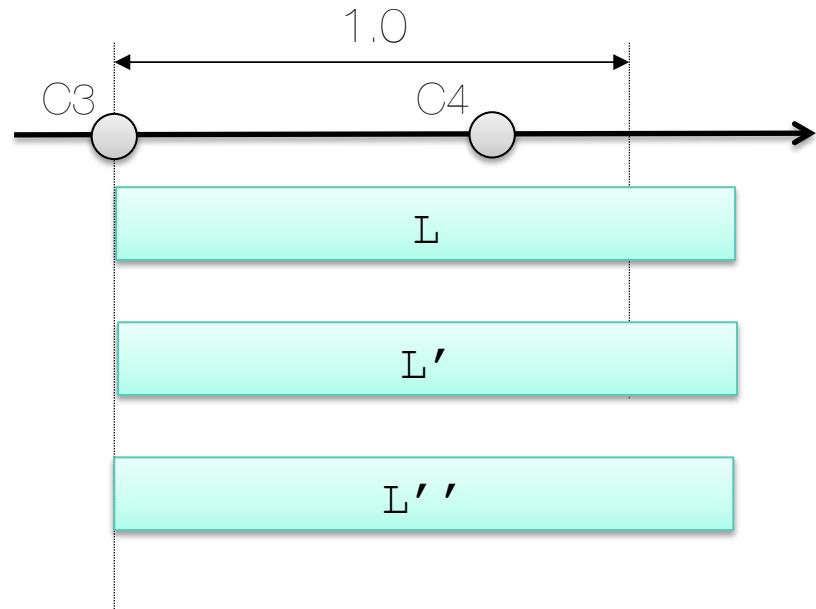
ForAll

Note C3 2/3

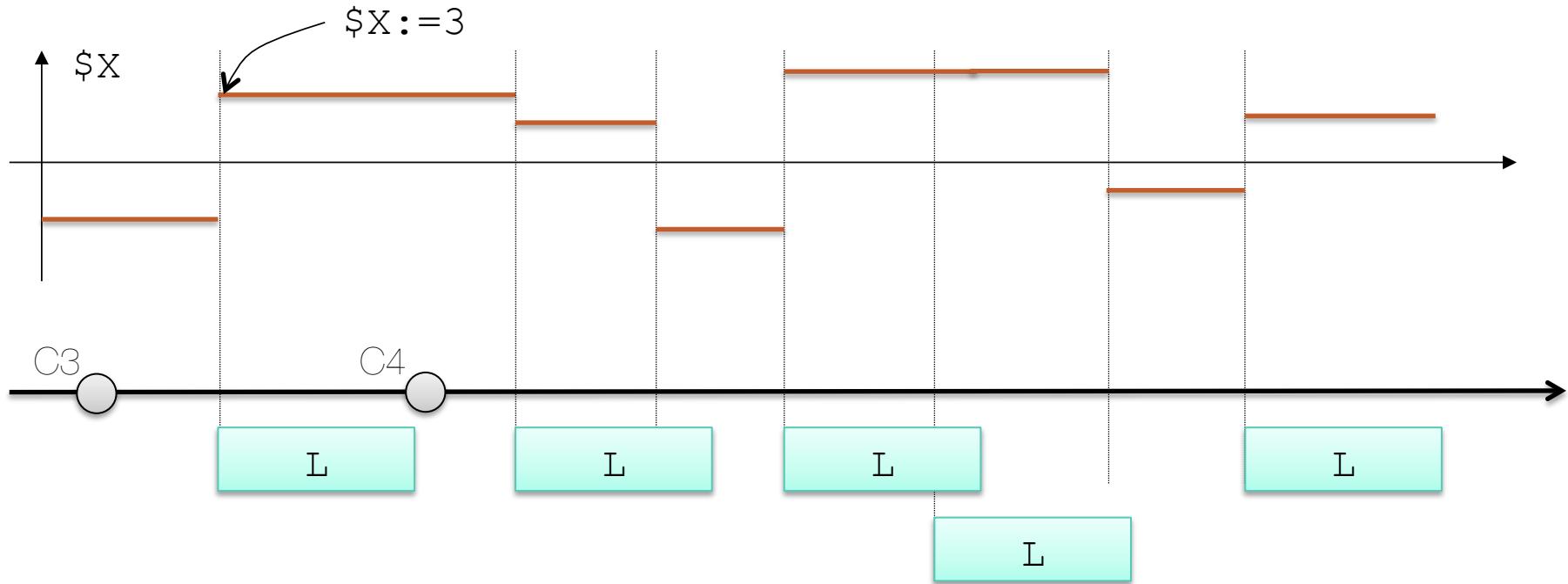
```
ForAll $x in TAB[1, 2, 3] do
{
    L
}
```

Note C4 1.5

...



Whenever



Note C3 2/3

Whenever (\$X > 0)

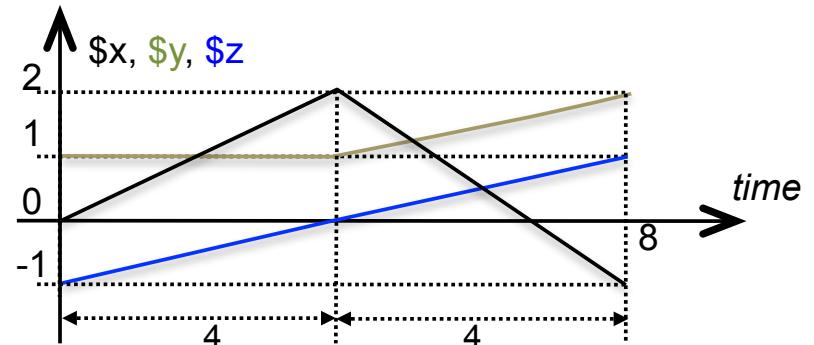
```
{  
    L  
}
```

Note C4 1.5

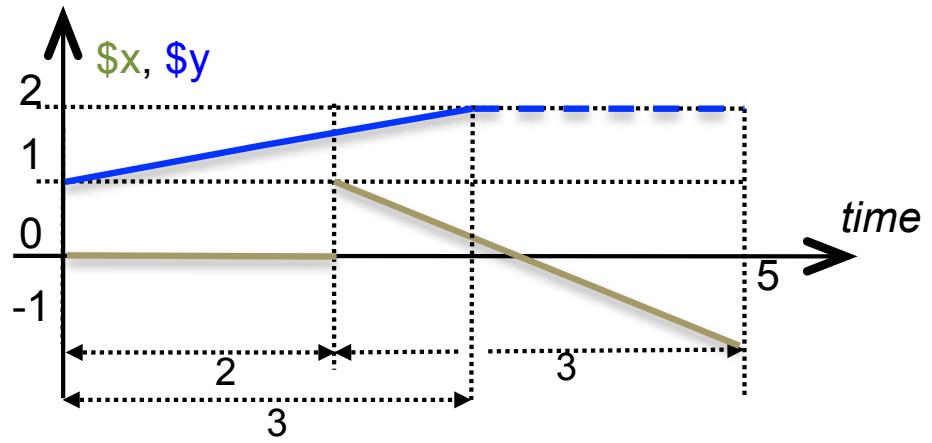
...

Curve

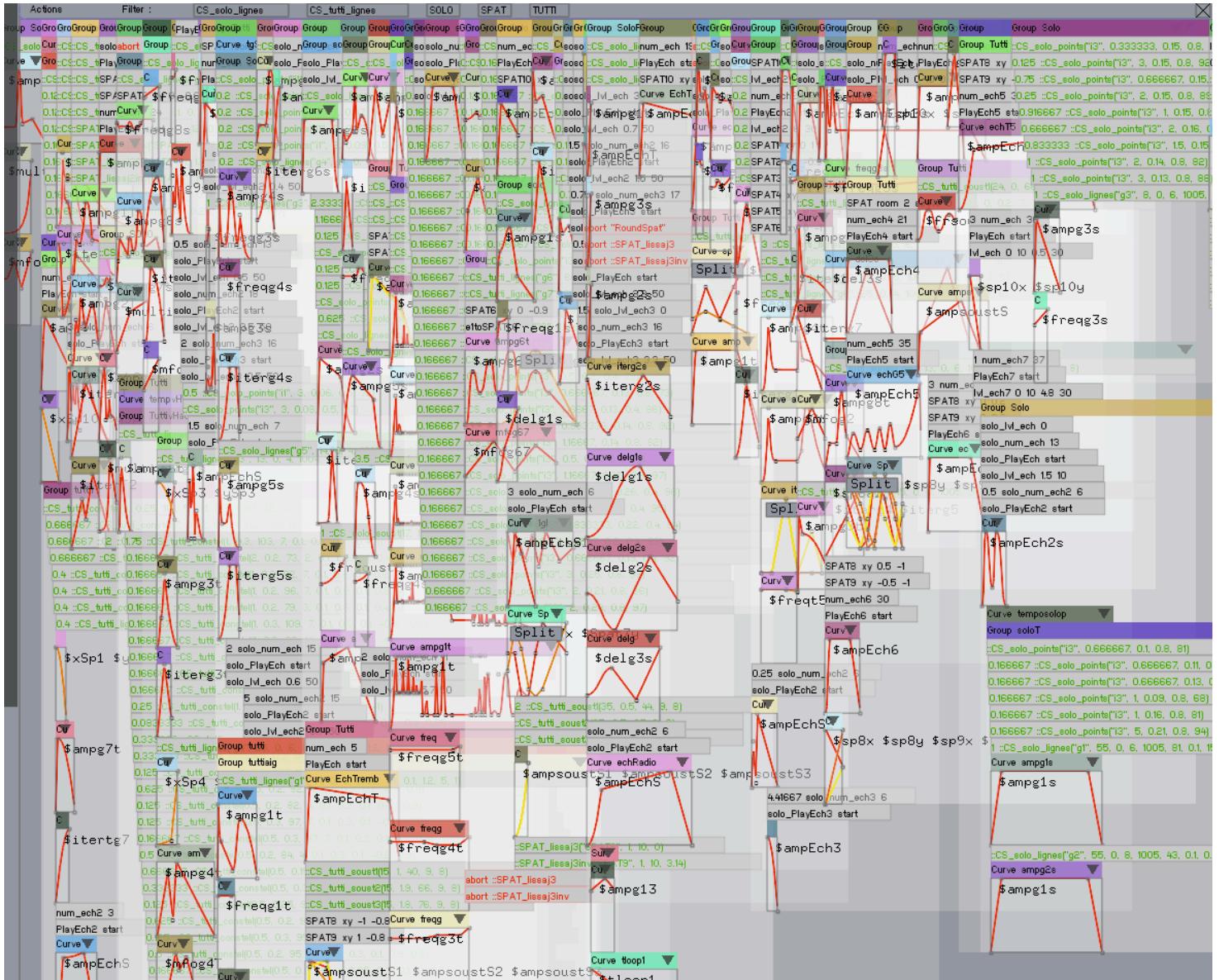
```
curve C
{
    même durée, mêmes breakpoints
    $x, $y, $z
    {
        { 0, 1, -1 } @linear
        4 { 2, 1, 0 } @linear
        4 { -1, 2, 1 }
    }
}
```



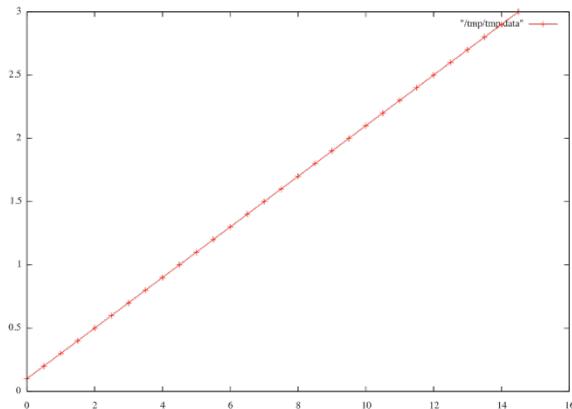
```
curve C
{
    même durée, différents breakpoints
    $x
    {
        { 0 } @constant
        2 { 1 } @linear
        3 { -1 }
    }
    $y
    {
        { 1 } @linear
        3 { 2 }
    }
}
```



Exemple: contrôle de synthèse (Nachtleben, 5mn, Julia Blondeau)

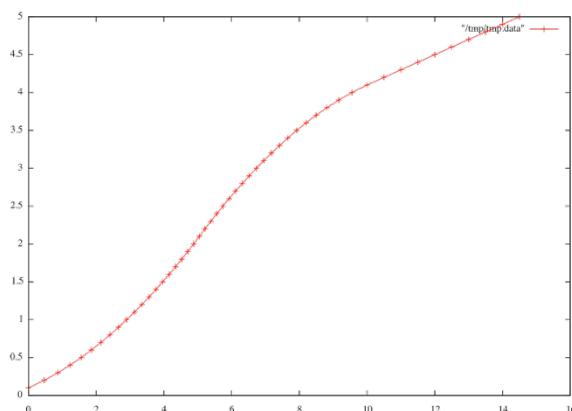


Exemple d'empilement de tempi



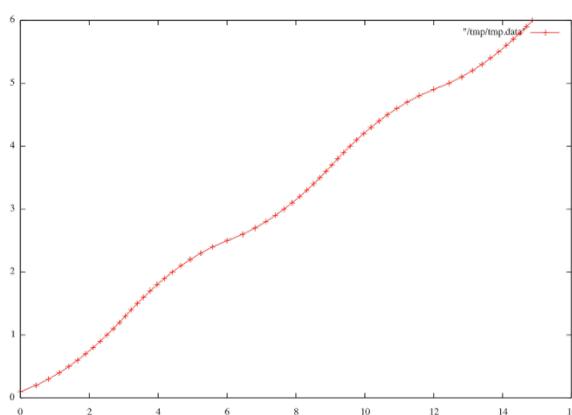
Curve C1 @grain 0.05s

```
{ $t1 { {60} 5 {180} 5 {60} } }
```



Group G1 @tempo := \$t1

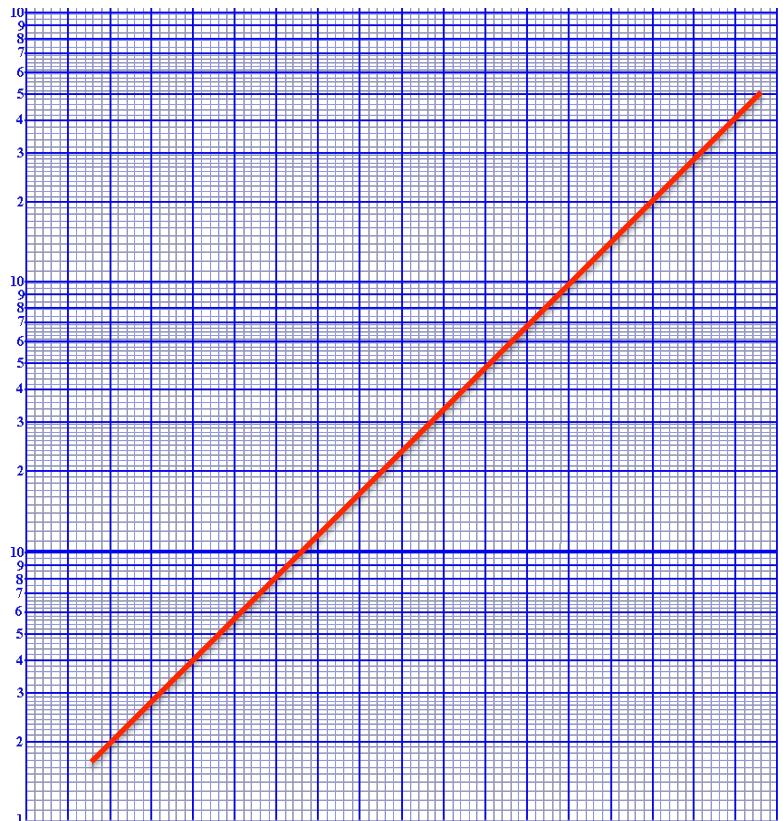
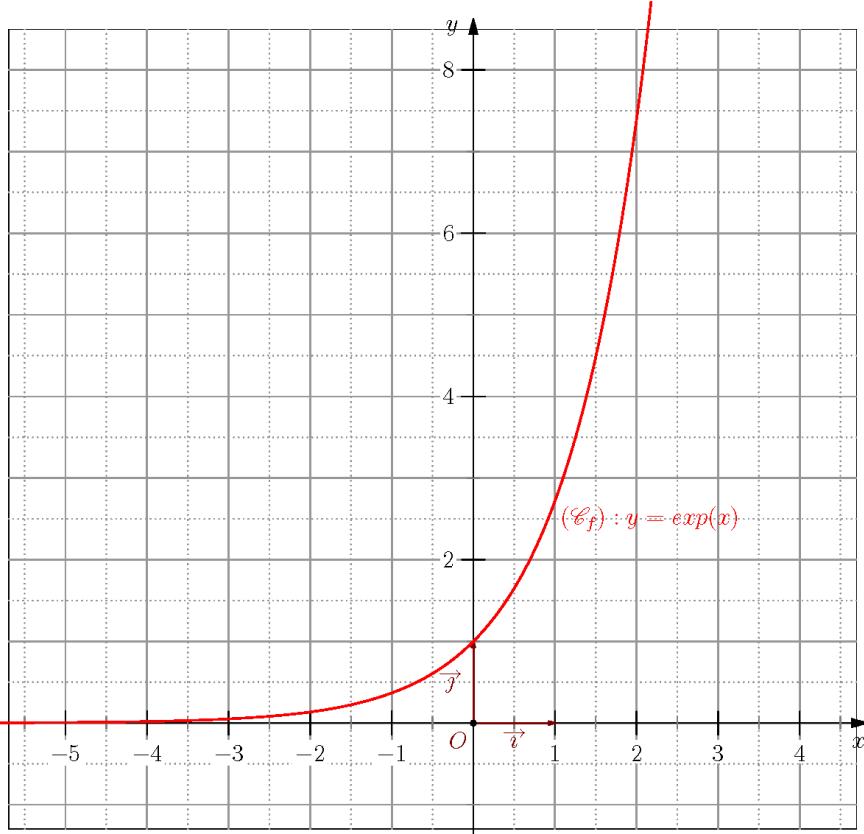
```
{  
    Curve C2 @grain 0.05s  
    { $t2 { {60} 3 {180} 3 {60} 3 {180}  
          3 {60} 3 {180} 3 {60} }  
}
```



Group G3 @tempo := \$t2

```
{  
    @local $x  
    $x := 0  
    Loop L 0.5  
    {  
        $x := $x + 0.1  
        plot $NOW " " $x "\n"  
    } during [15]  
}
```

Pourquoi créer ses propres repères temporels ?



Processus

1. ce sont des **valeurs** (comme les fonctions)
2. qu'on peut appeler
le résultat est la création d'une **instance** de groupe
3. **l'appel** est soit une expression, soit une action
4. les processus peuvent être **récursifs**
5. et d'ordre sup

Processus

```
@Proc_def :: Tic($x) {  
    $x print TIC  
}  
  
@proc_def :: Toc($x) {  
    $x print TOC  
}  
  
@proc_def :: Clock($p, $q) {  
    :: $p(1)  
    :: $q(2)  
    3 :: Clock($q, $p)  
}
```

généricité: on peut abstraire sur toute expression

appel calculé (décision, contrôle dynamique)

appel récursif (avec une fonction ça bouclerait)

passage en paramètre (un proc est une valeur comme les autres)

Clock(::Tic, ::Toc)

Processus comme agent

```
@global $incremente, $decremente ; canaux de communication

@Proc_def ::P($id)
{
    @local $state

    whenever ($incremente = $id)
    {
        $state := $state + 1
    }

    whenever ($decremente = $id)
    {
        $state ;:: $state - 1
    }
}

...
::P("José") ; instantiation de l'agent « José »

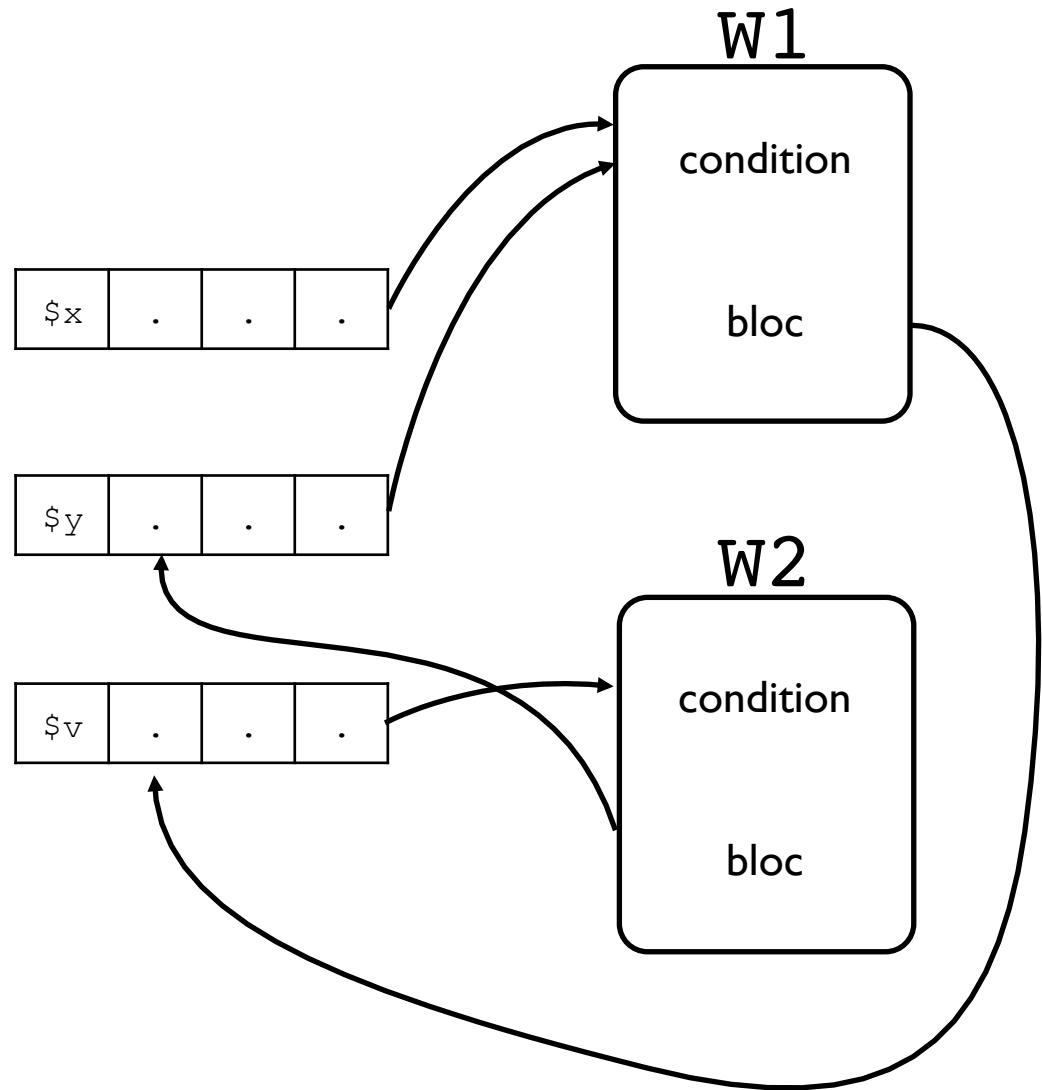
...
$incremente := "José" ; envoi du message « incrémenté » à José
$decremente := "José" ; envoi du message « décrémenté » à José
```

CAUSALITÉ

Causalité circulaire (court-circuit temporel)

```
whenever W1 ($x>$y)
{
    $v := $v+1
}
```

```
whenever W2 ($v>10)
{
    $y := 34
}
```

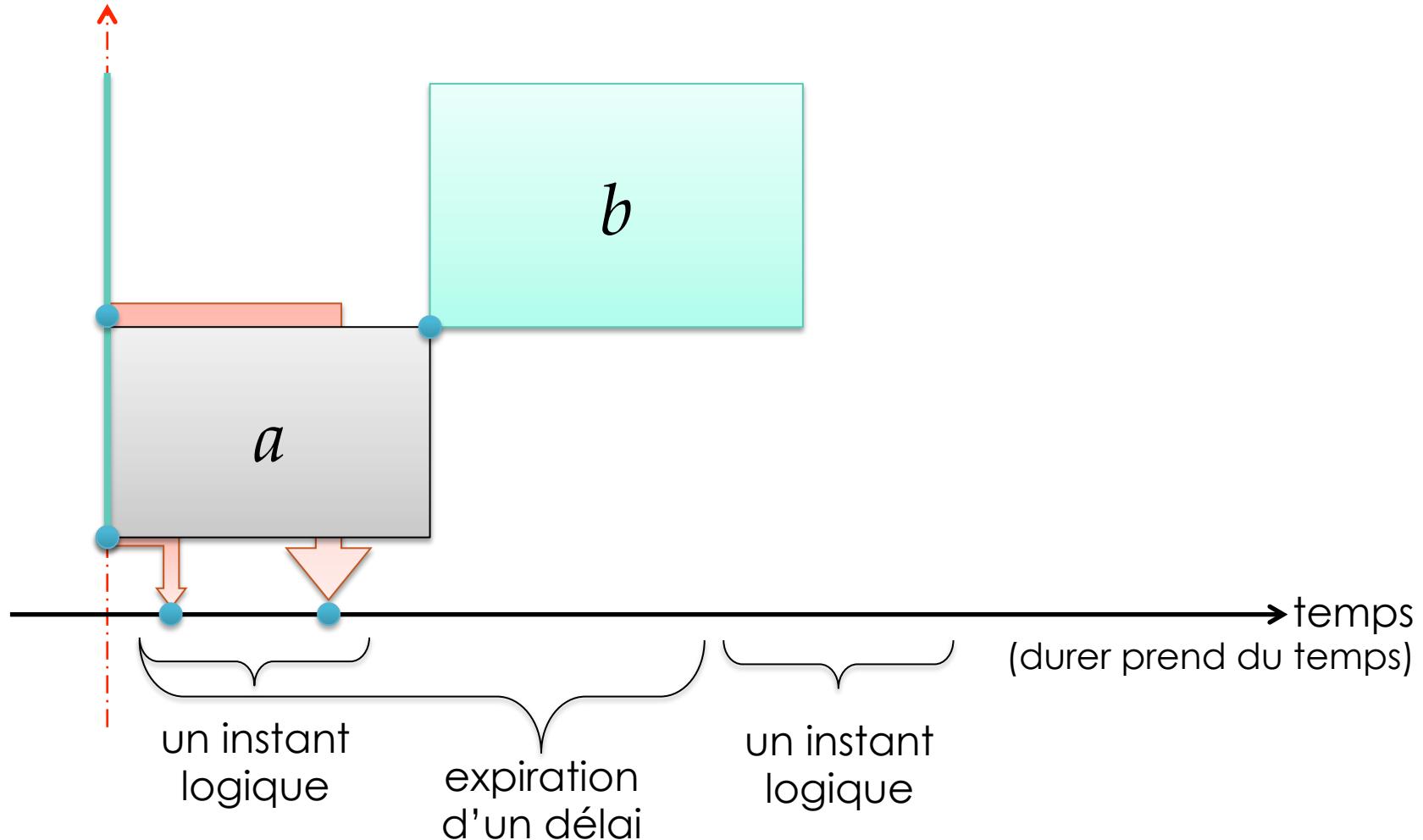


Causalité « normale »

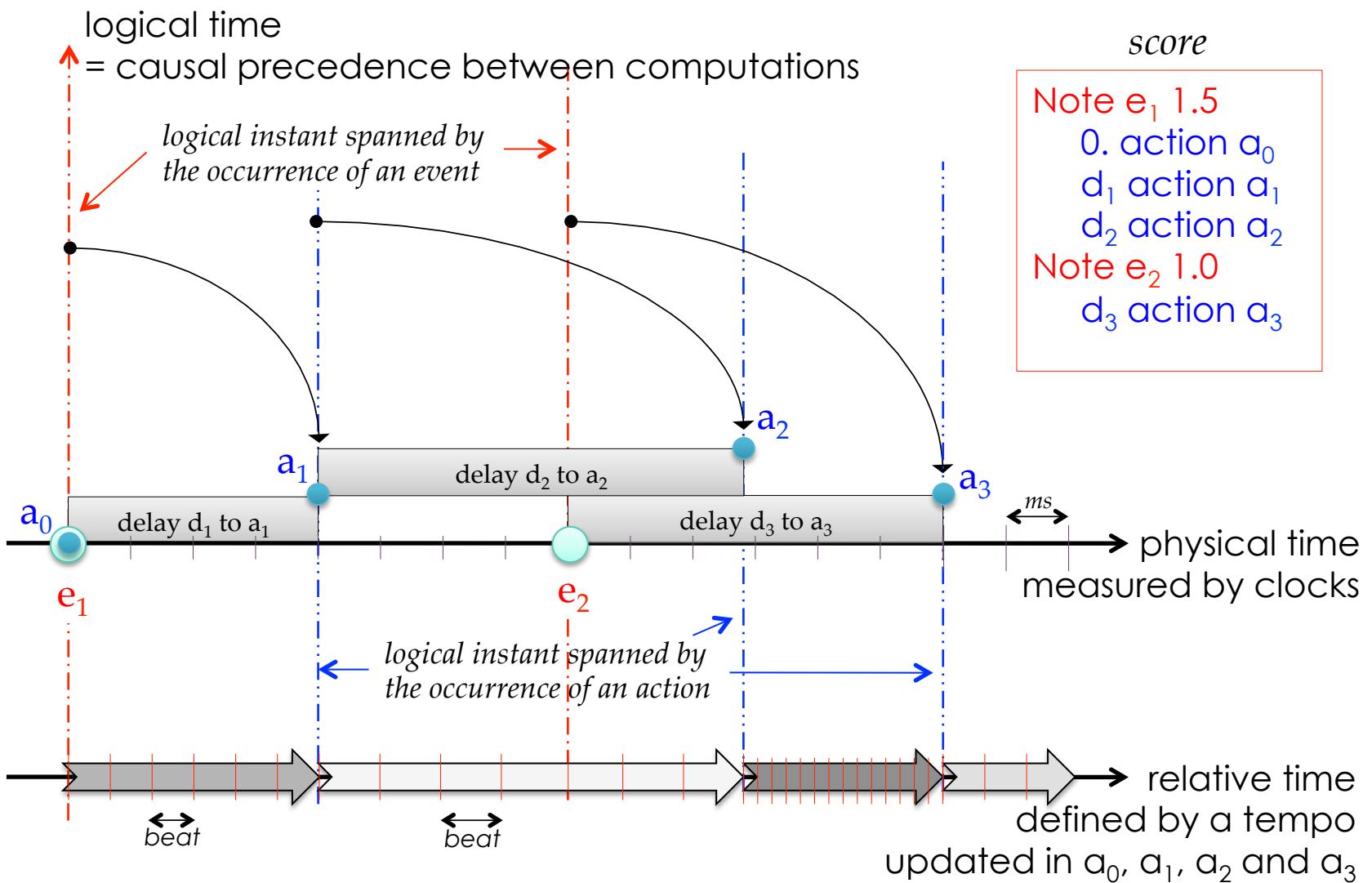
```
let $x := 1                                0: Start →
let $y := 1                                1: W1 →
whenever W1 ($x > 0)                      2: W2 →
{
    1 let $y := $y + 1                     3: W1 →
}
whenever W2 ($y > 0)                      4: W2 →
{
    1 let $x := $x + 1                     5: W1 →
}
let $x := 10 @name Start                   6: W2 →
                                            7: W1 →
                                            8: W2 →
                                            9: W1 →
                                            ...
...
```

Causalité & Durée

dépendances
(causation → succession)

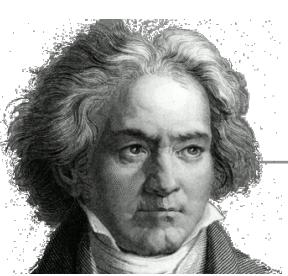


Synchronous (e.g. zero duration) & timed action

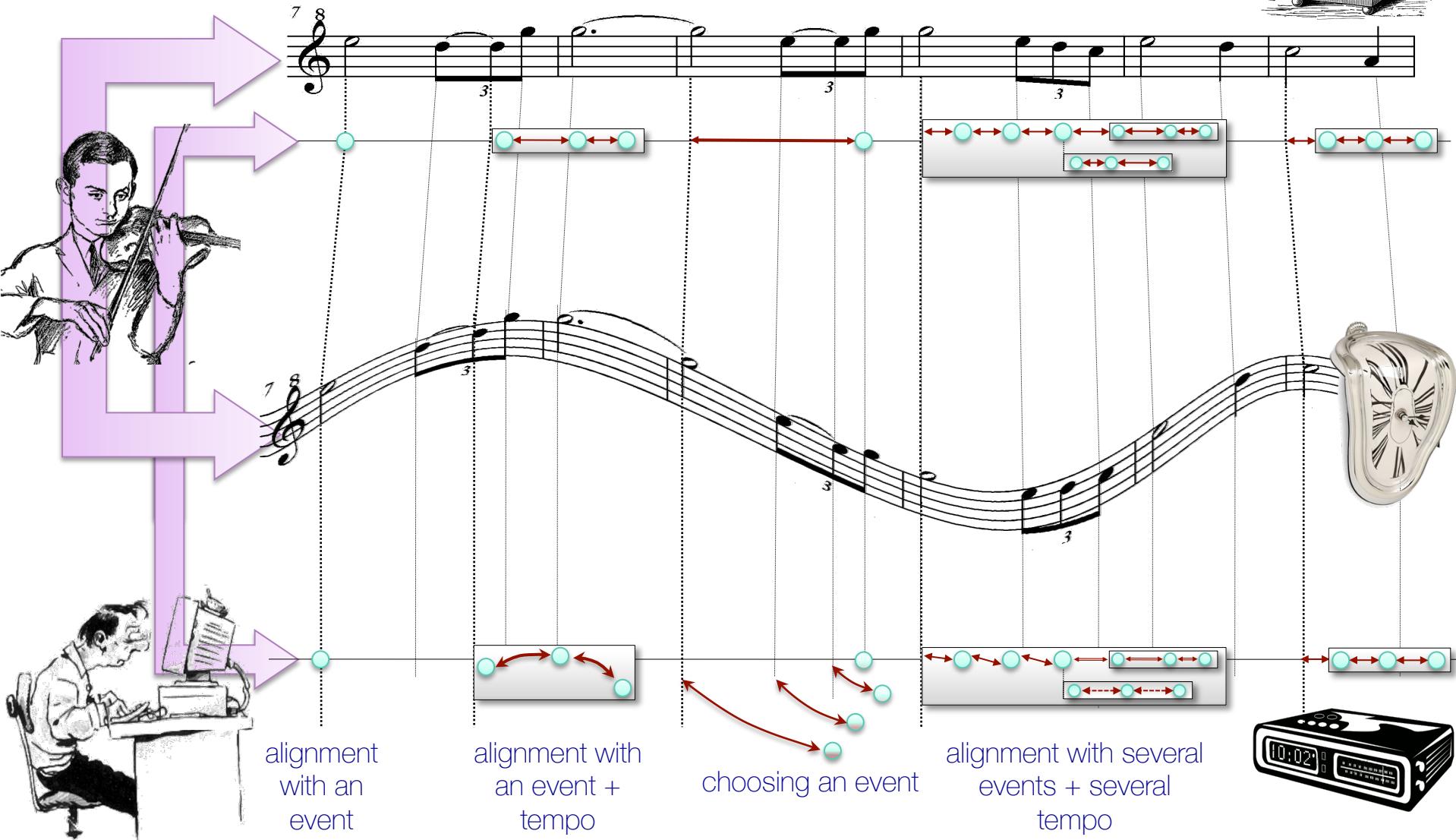
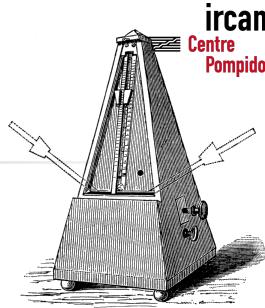


Antescofo

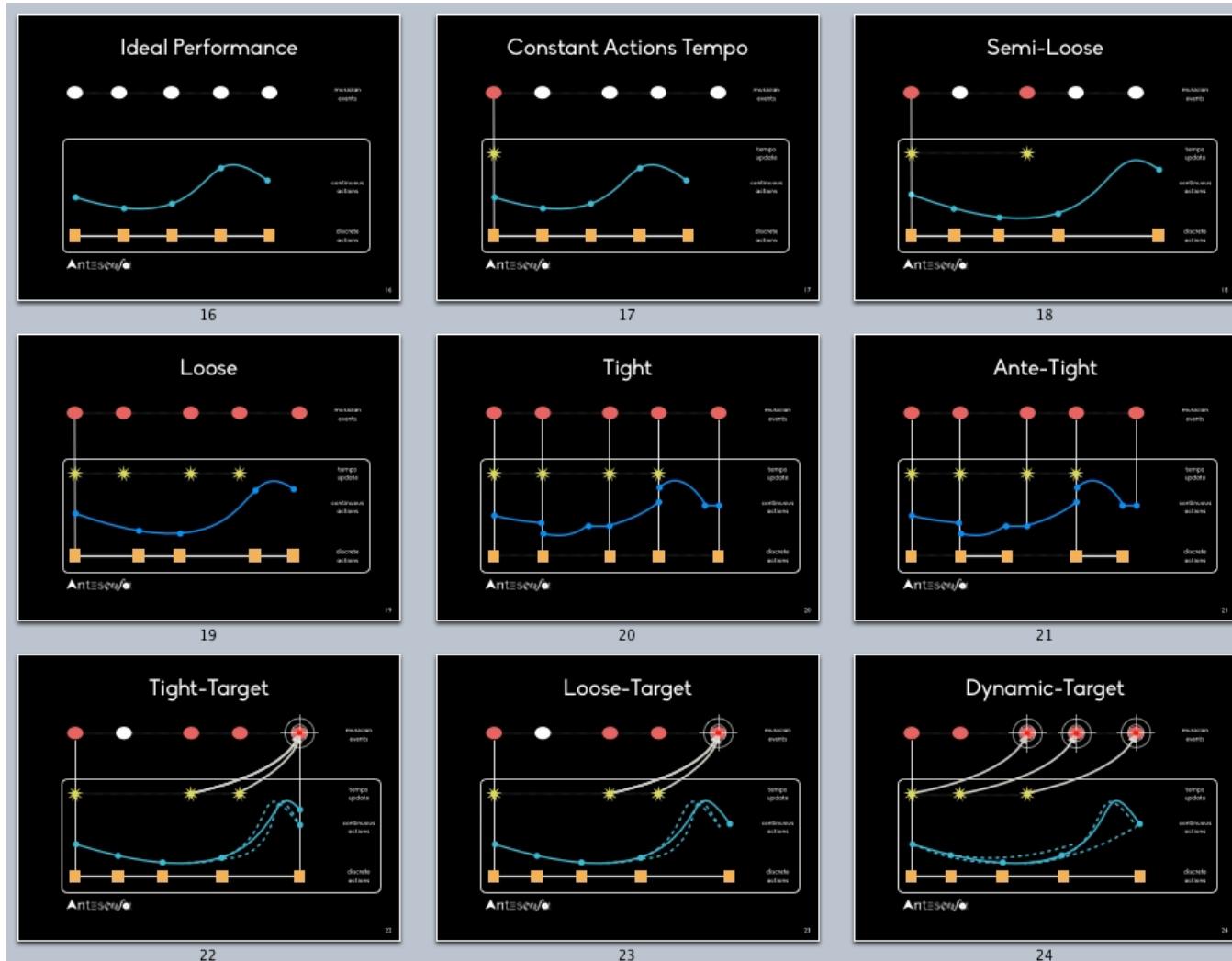
SYNCHRONISATION



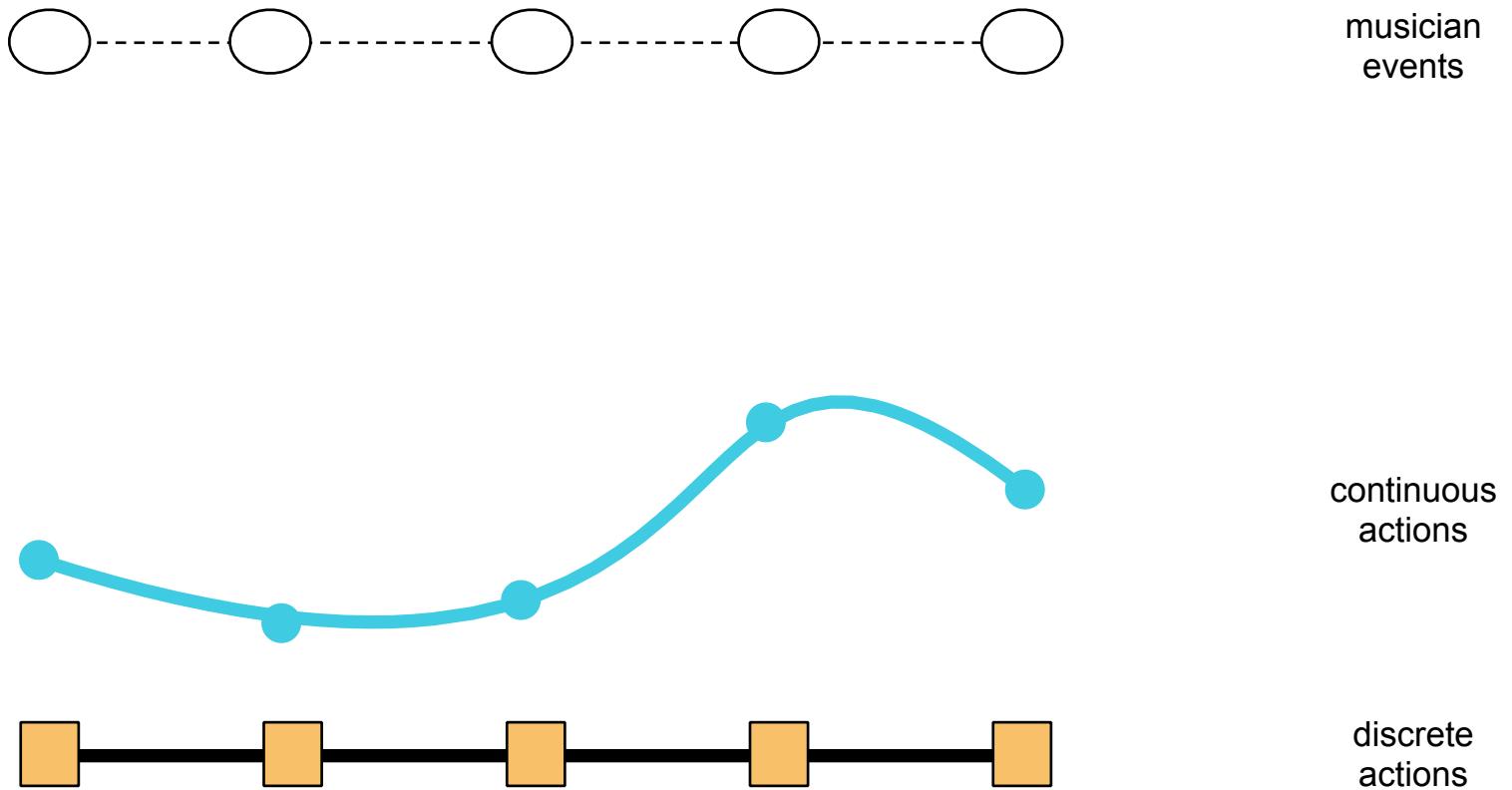
Time alignment of Score, Human-Performance and Electronic Actions



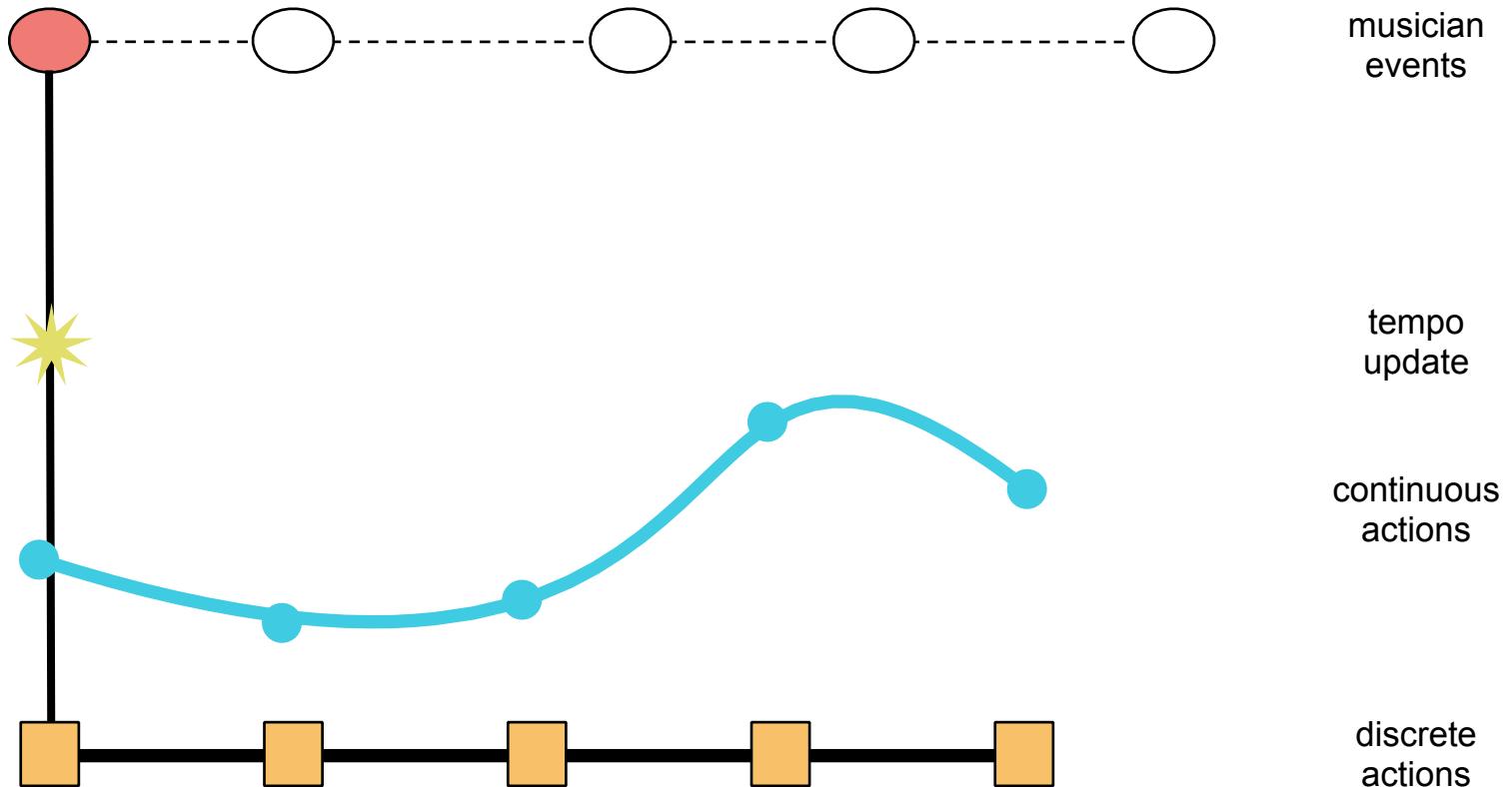
Stratégies de Synchronisation



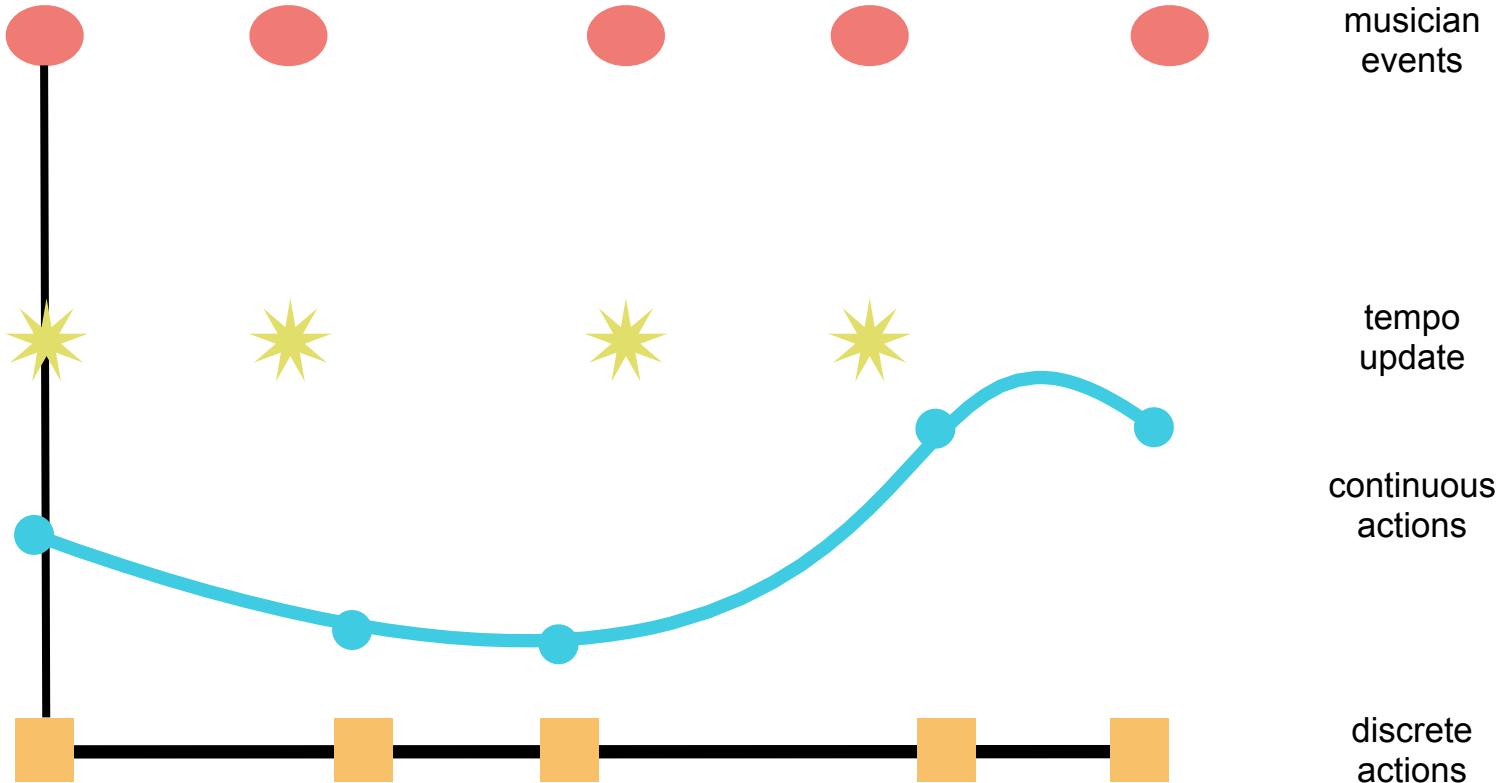
Ideal Performance



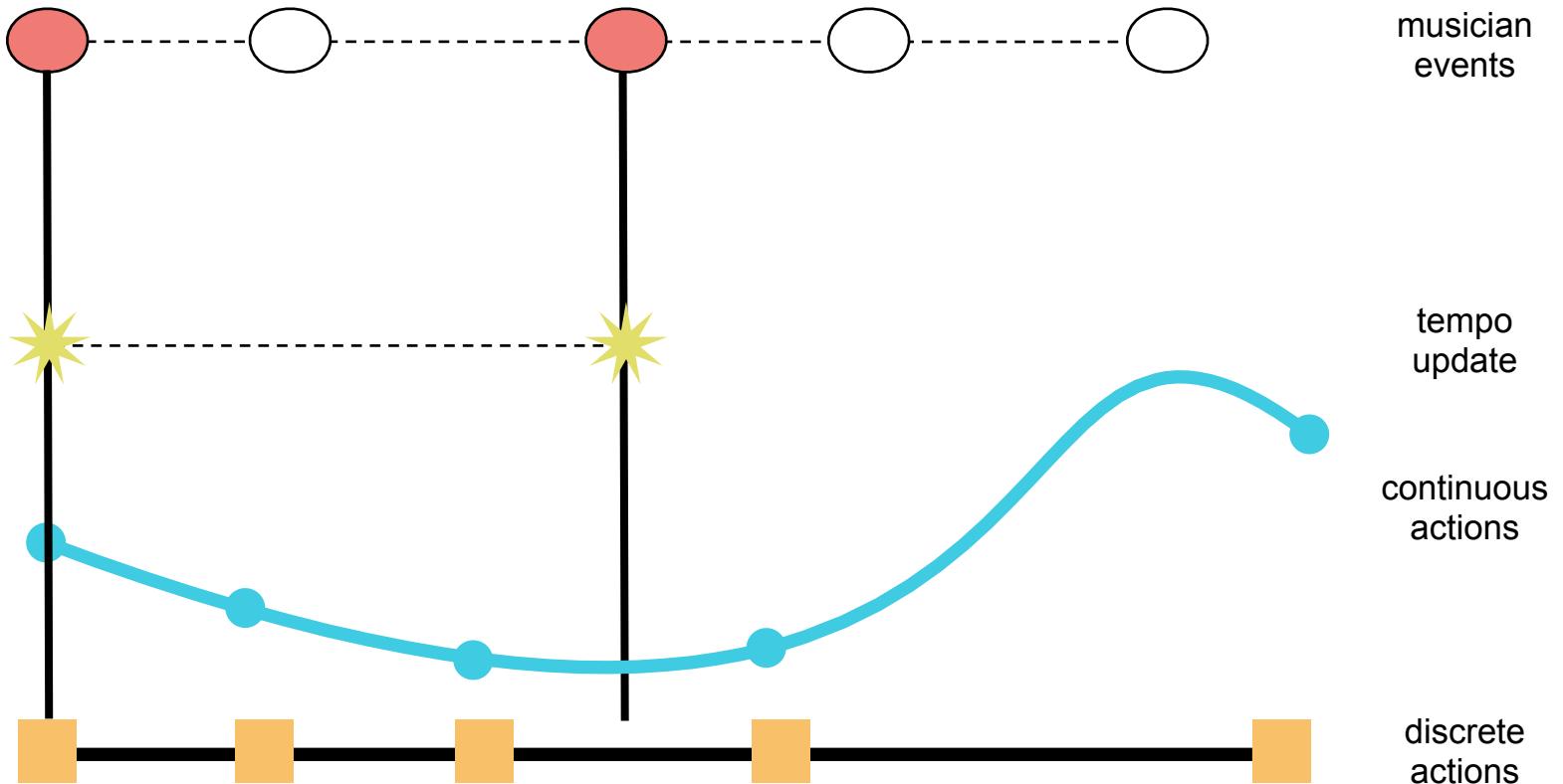
Constant Actions Tempo



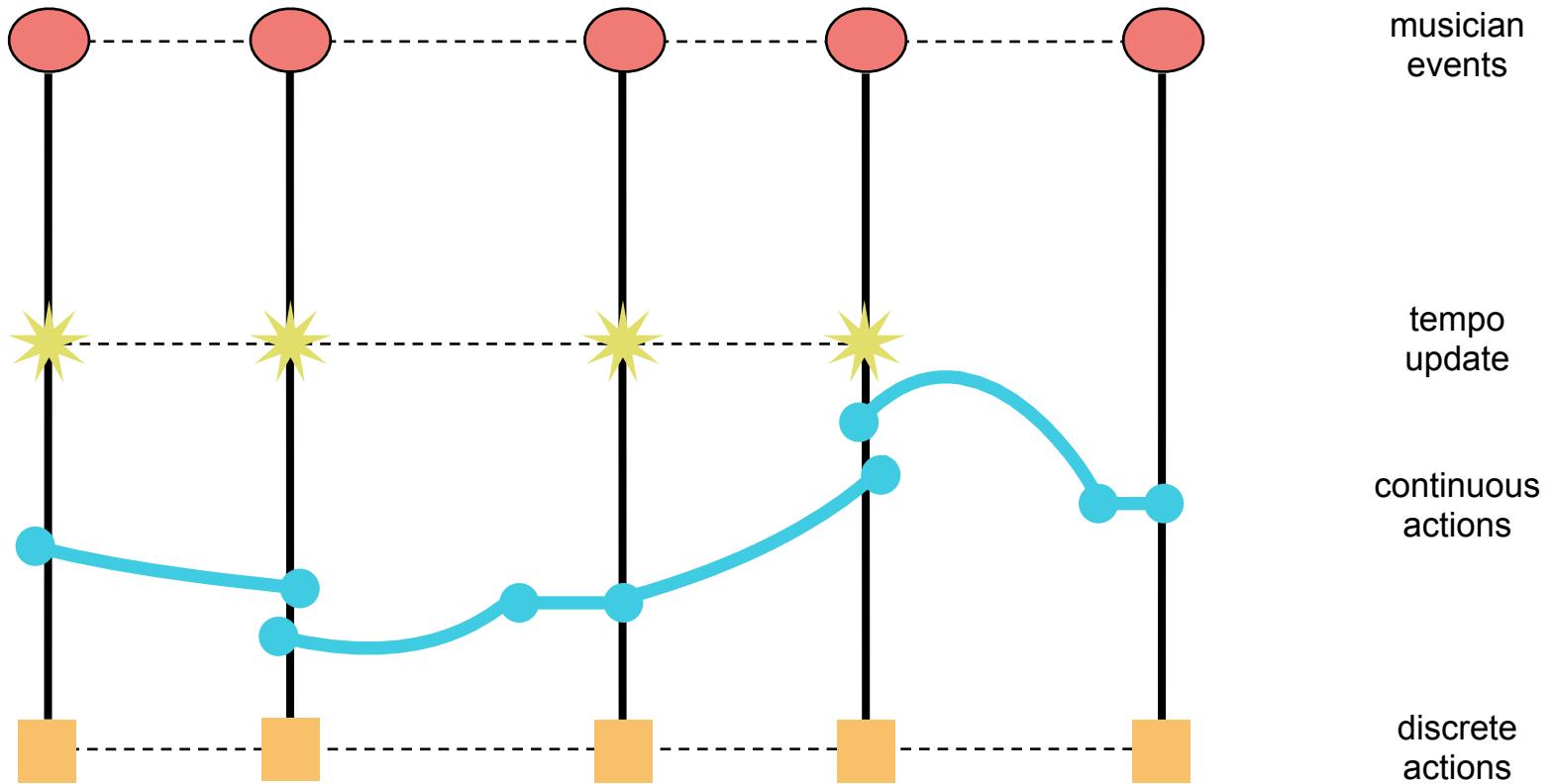
Loose



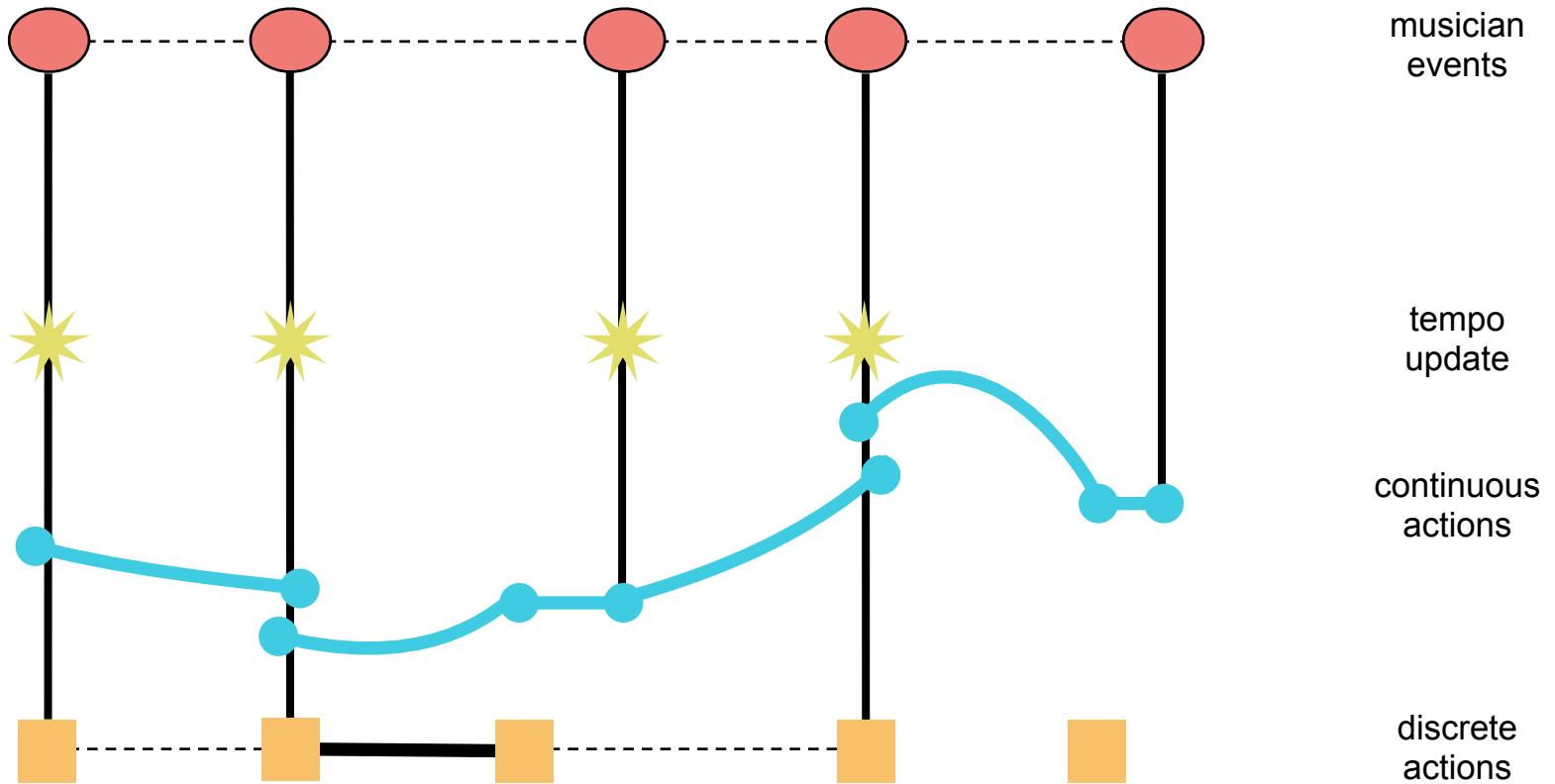
Target



Tight

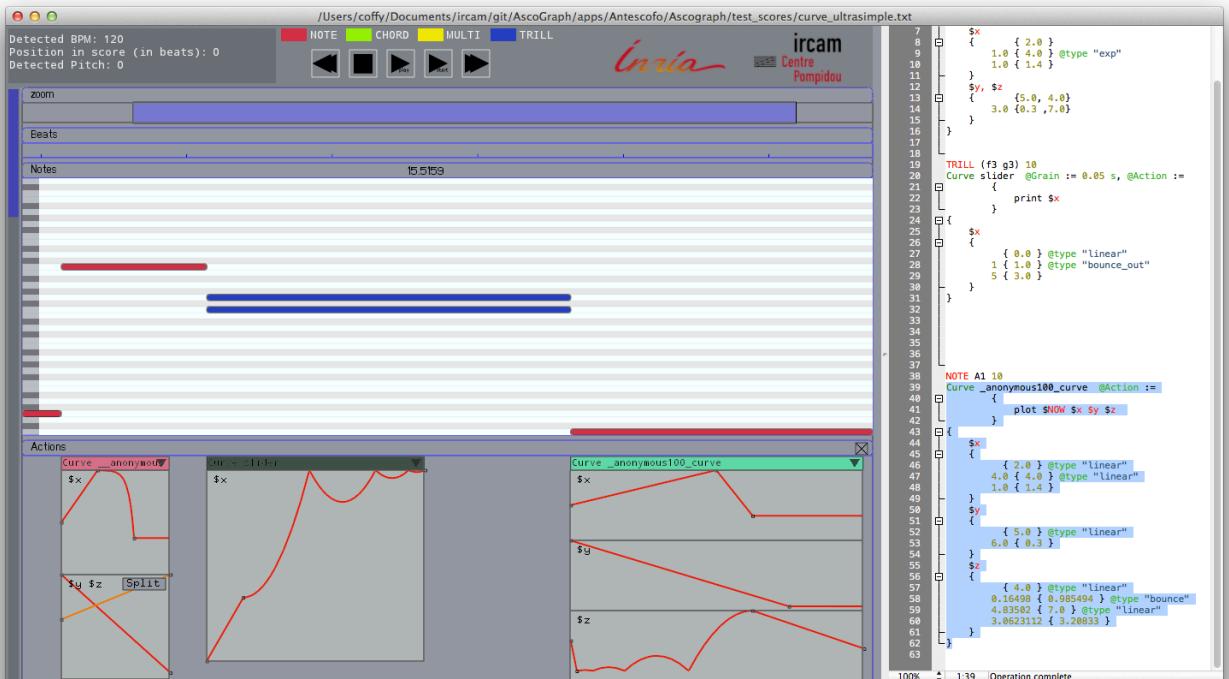


Ante-Tight



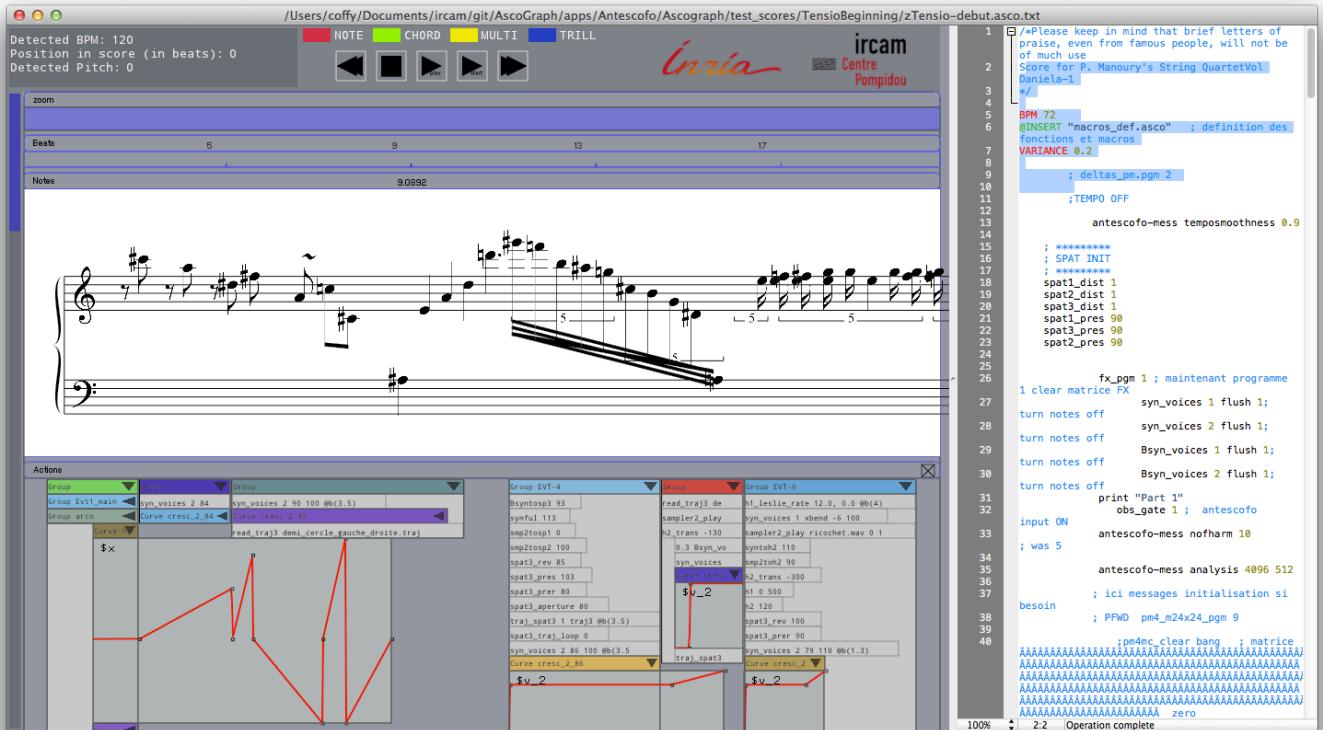
IMPLEMENTATION

Implementation



- best effort
- OS X or Linux or Windows
- embedded in Max/msp, PureData or standalone
- graphical interface for edition and monitoring
- used in several creations and re-creation
- with BBC, Berliner Philharmoniker, Los Angeles Philharmonic, etc.

Implementation



- best effort
- OS X or Linux or Windows
- embedded in Max/msp, PureData or standalone
- graphical interface for edition and monitoring
- used in several creations and re-creation
- with BBC, Berliner Philharmoniker, Los Angeles Philharmonic, etc.

Implementation

The screenshot shows the AscoGraph software interface. At the top, there's a status bar with 'Detected BPM: 120', 'Position in score (in beats): 0', and 'Detected Pitch: 0'. Below it is a legend with four categories: NOTE (red), CHORD (green), MULTI (yellow), and TRILL (blue). To the right is the 'ircam Centre Pompidou' logo.

The main area has several sections:

- zoom:** A horizontal timeline showing beats from 20 to 286.
- Jumps:** A section showing transitions between notes and chords.
- Score View:** A grid-based view of the musical score with various notes and chords represented by colored shapes.
- Actions:** A row of buttons for actions like cue, cu21, etc.
- Score File:** On the right, the score is shown as a text file with musical instructions. Some parts are highlighted in blue.

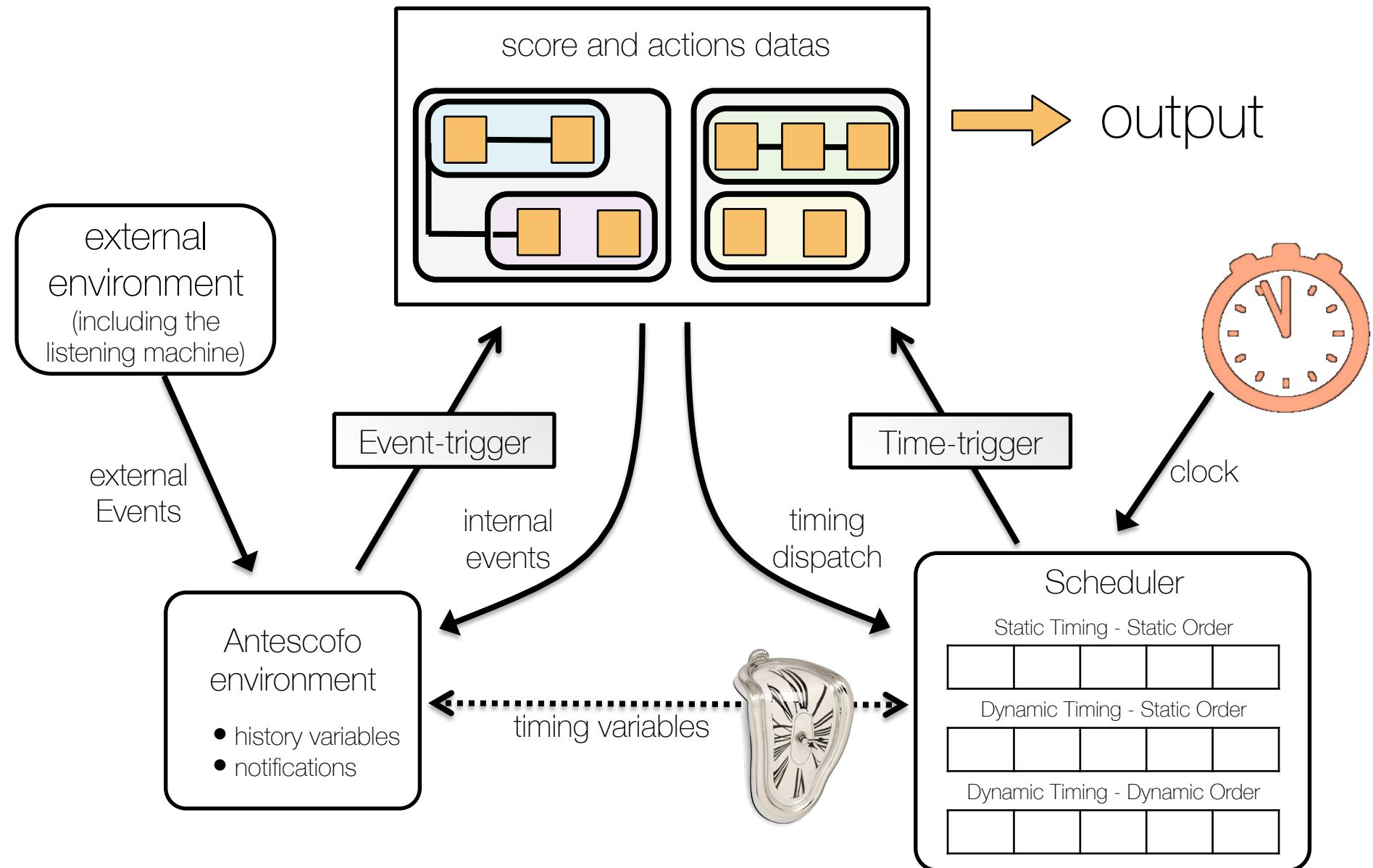
```

1 ; Pluto part III - Ouverte
2 BPM 60
3 VARIANCE 0.2
4 : ATTRACTOR
5 NOTE 0.0 8.0 @jump R1, R2, R3, R4, R5, R6,
6 ; Part R1
7 CHORD (F#3 C4 Eb4 B4 D5 Bb5) 4.0 R1
8 cue21 1
9 CHORD (C#3 F#3 C4 Eb4) 0.5
10 CHORD (D2 G2 C#3 F#3 C4 Eb4) 4.0 R1_fin @ju
11 cue21 3
12 NOTE 0 4.0 @ju
13 ; Part R2
14 BPM 80
15 CHORD (C4 G#4 B4) 0.125 R2
16 cue21 7
17 CHORD (C4 G#4 B4 D5 Bb5 Db6) 0.875
18 CHORD (F#5 A5) 0.125
19 CHORD (G6 C7) 0.875 ; add previous chord
20 CHORD (C4 G#4 B4) 1/3
21 CHORD (G6 C7) 0.75
22 CHORD (D6 G#6 A6 C#7) 0.125
23 CHORD (Bb7 E7) 2.0 R2_fin
24 cue21 8
25 NOTE 0 4.0 @ju
26 ; Part R3
27 CHORD (C4 D4 Gb4 Bb4 D#5 B5) 4.0 R3
28 cue21 11
29 NOTE 0 2.0
30 CHORD (C4 D4 G#4 B4 E5 Bb5 C#6 D#6 F#6) 4.0
31 NOTE 0 2.0
32 CHORD (C4 F#4 G#4 D5 E5 A5 B5 Eb6 Ab6 Cb7) 4.0
33 NOTE 0 2.0
34 CHORD (Bb2 F#3 C4 F5 C#6 D#6 G6) 4.0 R3,
35 cue21 12
36 NOTE 0 2.0 @ju
37 ; Part R4
38 TRILL ( (C4 F#6 B4 Ab5) (C4 F#6 Bb4 G5) ) 4.0
39 cue21 13
40 NOTE 0 4.0 R4_fin @ju
41 cue21 14
42 ; Part R5
43 CHORD (C4 A#4 D5 A6 Cb7 G7) 8.0 R5
44 cue21 15
45 NOTE 0 4.0 R5_fin
46 cue21 16
47 1:1 Operation complete
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

- best effort
- OS X or Linux or Windows
- embedded in Max/msp, PureData or standalone
- graphical interface for edition and monitoring
- used in several creations and re-creation
- with BBC, Berliner Philharmoniker, Los Angeles Philharmonic, etc.

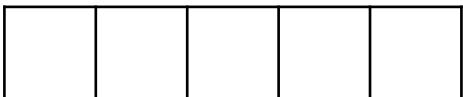
Runtime



Scheduler

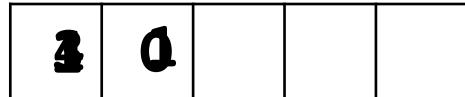
```
group
{
  2s a31
  1s a32
}
group
@tempo=65{
  4 a41
  2 a42
}
```

Static Timing - Static Order



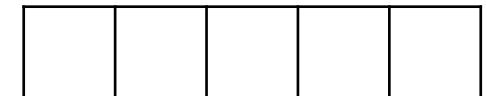
```
group{
  1 a00
  4 a11
  2 a12
}
group{
  1 a10
  3 a21
  2 a22
}
```

Dynamic Timing - Static Order



```
group
@tempo=$RT_TEMPO*2 {
  1/2 a51
  1 a52
}
group
@tempo=$v {
  2 a61
  3 a62
}
```

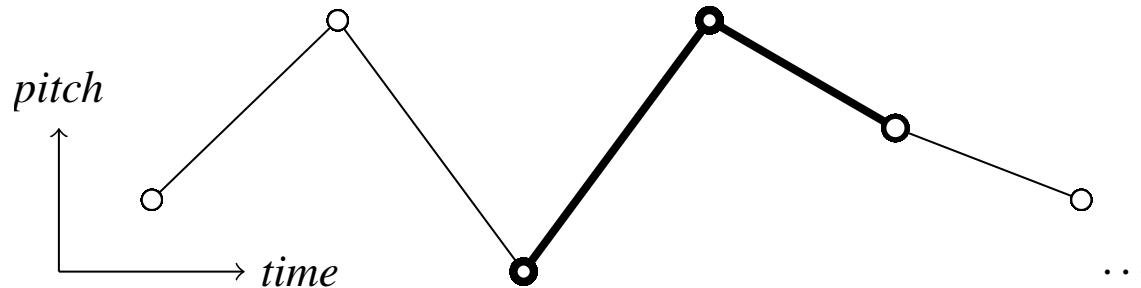
Dynamic Timing - Dynamic Order



Plus d'expressivité

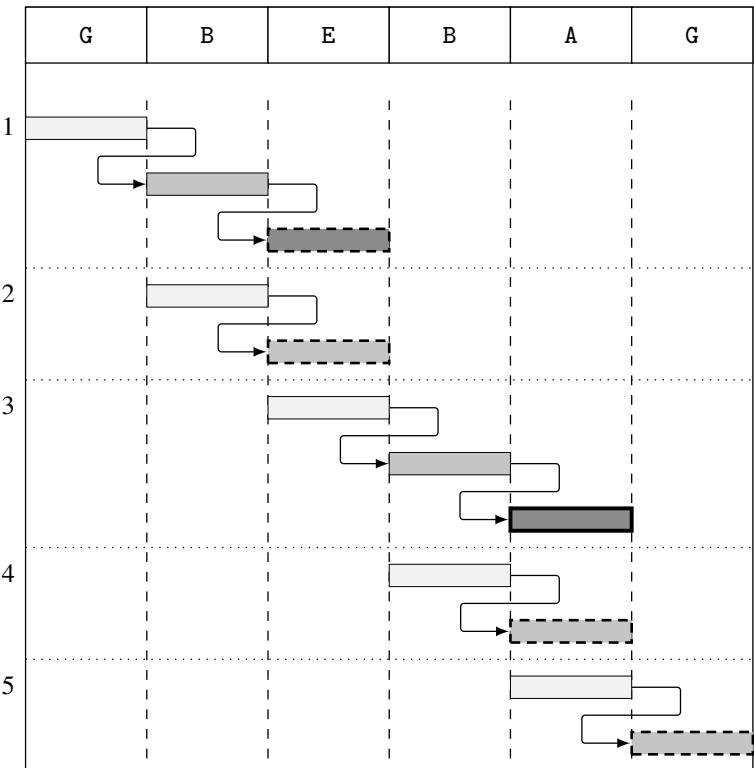
PATTERN & NEUME

Détecer un motifs temporel



```

1 whenever ($PITCH) {
2   @local $x
3   $x := $PITCH
4   whenever ($PITCH > $x) {
5     @local $y
6     $y := $PITCH
7     whenever ($PITCH<$y & $PITCH>$x) {
8       @local $z
9       $z := $PITCH
10      a
11    } during [1#]
12  } during [1#]
13 }
```



Event: une propriété ponctuelle

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 "j'ai vu un P" }
```

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

State: une propriété qui dure

une variable \$X prend une valeur v pendant au moins 2 beats

```
@Local $start, $stop
Event $X value v at $start
Event $X value v at $stop
  where ($stop - $start) >= 2
```

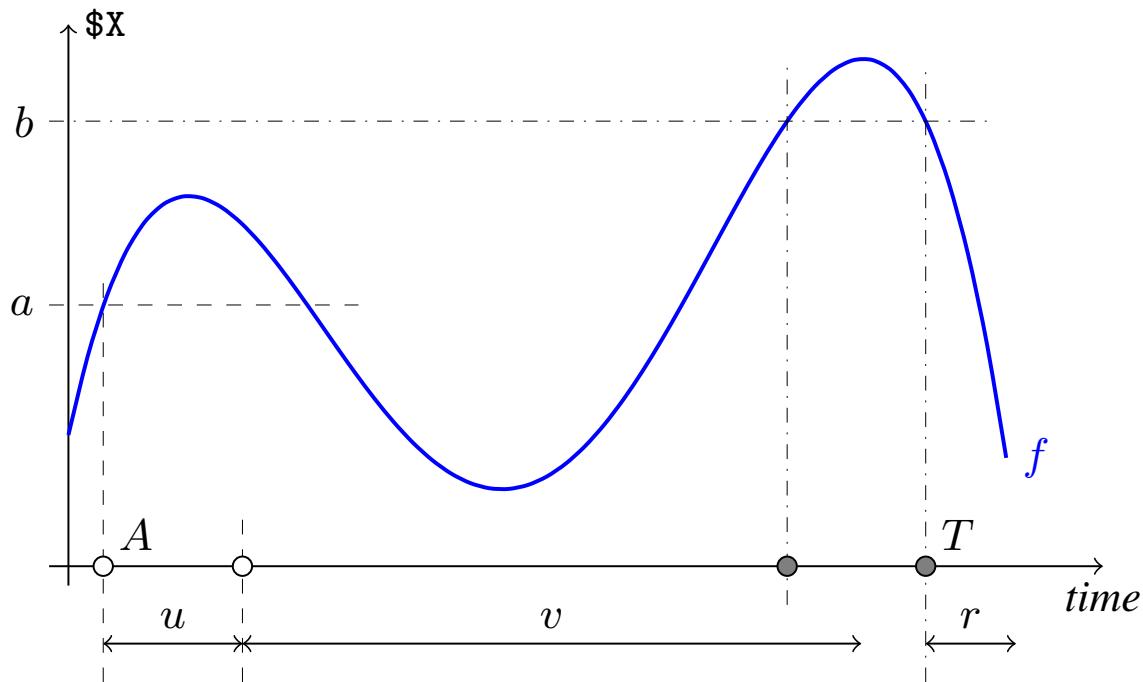
State \$X where (\$X == v) during 2

State: une propriété qui dure

State \$X during u where $$X > a$

Before $[v]$

State $$X$ where $$X > b$



Semantique de trace

- (1) $\mathbf{M}[\varepsilon] \rho S = \rho(\$NOW)$
- (2) $\mathbf{M}[P] \rho \epsilon = fail, \quad P \neq \varepsilon$
- (3) $\mathbf{M}[P_x \cdot Q] \rho (x' := v \cdot S) = \mathbf{M}[P_x \cdot Q] \rho[x' := v] S \quad \text{where } x \neq x'$

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

- (4) $\mathbf{M}[\text{before}[d] P_x \cdot Q] \rho (d' \cdot S) = \begin{cases} fail, & \text{if } d \leq d' \\ \mathbf{M}[\text{before}[d - d'] P_x \cdot Q] \rho[\$NOW += d'] S, & \text{if } d > d' \end{cases}$
- (5) $\mathbf{M}[\text{before}[0\#] P_x \cdot Q] \rho S = fail$
- (6) $\mathbf{M}[\text{before}[n\#] P_x \cdot Q] \rho (d' \cdot S) = \mathbf{M}[\text{before}[n\#] P_x \cdot Q] \rho S$
- (7) $\mathbf{M}[\text{before}[D] P_x \cdot Q] \rho (x := v \cdot S) = \begin{cases} \mathbf{M}[P'_x \cdot Q] \rho' S, & \text{if } \mathbf{E}[e]\rho'' = false \\ \min(\mathbf{M}[P'_x \cdot Q] \rho' S, \mathbf{M}[Q] \rho'' S) & \text{if } \mathbf{E}[e]\rho'' = true \end{cases}$
where $\rho' = \rho[x := v]$ and $\rho'' = \rho'[y := \rho(\$NOW), z := v]$ and $P'_x = \begin{cases} \text{before}[d] P_x, & \text{if } D = d \\ \text{before}[(n-1)\#] P_x, & \text{if } D = n\# \end{cases}$

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

- (8) $\mathbf{M}[\text{before}[d] \bar{P}_x \cdot Q] \rho (d' \cdot S) = \begin{cases} fail & \text{if } d \leq d' \wedge \mathbf{E}[e]\rho = false \\ \mathbf{M}[\text{before}[d - d'] \bar{P}_x \cdot Q] \rho' \cdot S & \text{if } d > d' \wedge \mathbf{E}[e]\rho = false \\ \mathbf{M}_{\mathbf{S}}[\bar{P}_x \cdot Q] \rho (d' \cdot S) & \text{if } d \leq d' \wedge \mathbf{E}[e]\rho = true \\ \min \left(\mathbf{M}_{\mathbf{S}}[\bar{P}_x \cdot Q] \rho (d' \cdot S), \mathbf{M}[\text{before}[d - d'] \bar{P}_x \cdot Q] \rho' \cdot S \right) & \text{if } d > d' \wedge \mathbf{E}[e]\rho = true \end{cases}$
where $\rho' = \rho[\$NOW += d']$

- (9) $\mathbf{M}[\text{before}[d] \bar{P}_x \cdot Q] \rho (x := v \cdot S) = \begin{cases} \mathbf{M}[\text{before}[d] \bar{P}_x \cdot Q] \rho' S & \text{if } \mathbf{E}[e]\rho' = false \\ \min \left(\mathbf{M}_{\mathbf{S}}[\bar{P}_x \cdot Q] \rho' \cdot S, \mathbf{M}[\text{before}[d] \bar{P}_x \cdot Q] \rho' \cdot S \right) & \text{if } \mathbf{E}[e]\rho' = true \end{cases}$
where $\rho' = \rho[x := v]$

- (10) $\mathbf{M}_{\mathbf{S}}[\bar{P}_x \cdot Q] \rho \epsilon = fail$

$$\mathbf{M}_{\mathbf{S}}[\bar{P}_x \cdot Q] \rho (x' := v \cdot S) = \mathbf{M}_{\mathbf{S}}[\bar{P}_x \cdot Q] \rho[x' := v] S \quad \text{where } x \neq x'$$

- (11) $\mathbf{M}_{\mathbf{S}}[P_x \cdot Q] \rho (d' \cdot S) = \mathbf{M}_{\mathbf{S}}[P_x \cdot Q] \rho[\$NOW := d'] S$

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

- (12) $\mathbf{M}_{\mathbf{S}}[P_x \text{ during}[d] \cdot Q] \rho (d' \cdot S) = \begin{cases} \mathbf{M}[Q] \rho[\$NOW += d] (d' - d \cdot S) & \text{if } d \leq d' \\ \mathbf{M}_{\mathbf{S}}[P_x \text{ during}[d - d'] \cdot Q] \rho[\$NOW += d'] S & \text{if } d > d' \end{cases}$

$$\mathbf{M}_{\mathbf{S}}[P_x \text{ during}[d] \cdot Q] \rho (x := v \cdot S) = \begin{cases} fail & \text{if } \mathbf{E}[e]\rho[x := v] = false \\ \mathbf{M}_{\mathbf{S}}[P_x \text{ during}[d] \cdot Q] \rho[x := v] S & \text{if } \mathbf{E}[e]\rho[x := v] = true \end{cases}$$

Compilation

par réécriture dans un sous-langage

```
@pattern P2
{
  @local $s1, $t1, $s2, $t2
  @refractory 2

  State $X start $s1 stop $t1 during 0.5
    where $X > [1#]:$X
  before [1.3]
  State $X start $s2 stop $t2 during 0.5
    where $X > [1#]:$X
}
```

```
@local $04_last_matching_time,
      $08_started

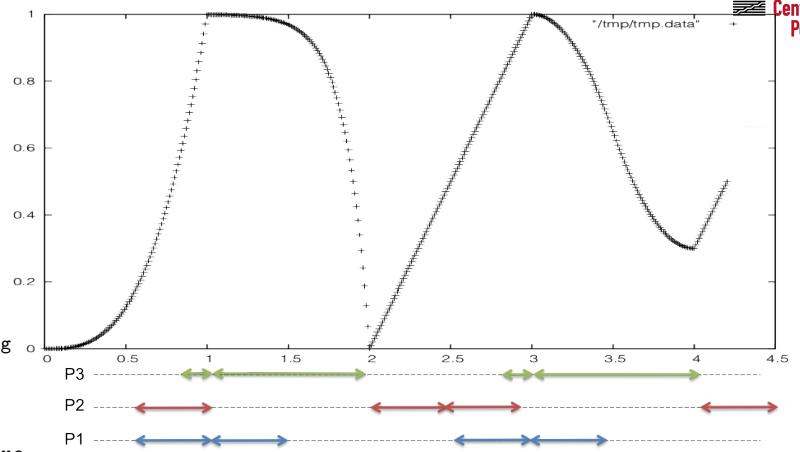
$04_last_matching_time := -1.0
$08_started := -1.0
WHENEVER ($X == $X) @Immediate
{
  @local $21_duration, $s, $t

  $21_duration := 0.2
  $s := $NOW
  $t := $NOW + $21_duration
  if ($X > 0.5 && $X > [1#]:$X)
  {
    @local $03_continue_matching

    if ($08_started < 0.0) {
      $08_started := $NOW
    }
    $03_continue_matching := true
    $21_duration
      if ($08_started >= 0.0 && $s >= $08_started)
    {
      @local $09_started, $17_tscope, $18_cpt, $13_halt

      $09_started := -1.0
      $17_tscope := 2
      $13_halt := false
      $18_cpt := -1
      WHENEVER ($03_continue_matching && $X == $X) @Immediate
      {
        @local $s2, $t2

        $18_cpt := $18_cpt + 1
        $s2 := ($09_started >= 0.0 ? $09_started : $NOW)
        $t2 := $NOW
        if ($X < [1#]:$X)
        {
          if ($09_started < 0.0) {
            $09_started := $NOW
          }
        } else {
          if ($09_started >= 0.0)
          {
            $13_halt := true
            if ($04_last_matching_time < 0.0
                || $NOW - $04_last_matching_time >= 2)
            {
              $04_last_matching_time := $NOW
              $03_continue_matching := false
              print "OK\u201astart1\u201a" $s1 "\u201astop1\u201a" $t1
              print "OK\u201astart2\u201a" $s2 "\u201astop2\u201a" $t2
            }
          }
        }
        $13_halt := $18_cpt >= $17_tscope
        $09_started := -1.0
      }
    } until ($13_halt)
  } else {
    $08_started := -1.0
  }
}
```



Faire cohabiter deux temps (bis)

GALS

Une architecture pour l'œuvre ouverte interactive

L'écriture du temps :

- composition
- calculer du temps

Auteur (compositeur)



Ecriture du temps
dans le temps

lecteur
parsing

« read »



Perception

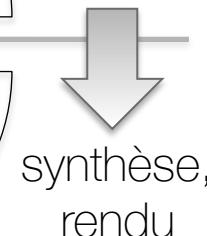
« eval »

Scénario interactif
partition ouverte,
virtuelle, interactive...

« print »

Action

écrivain
pretty-printing



Acquisition

Œuvre
interactive
œuvre musicale mixte,
time-art, ...

Production

traitement du signal

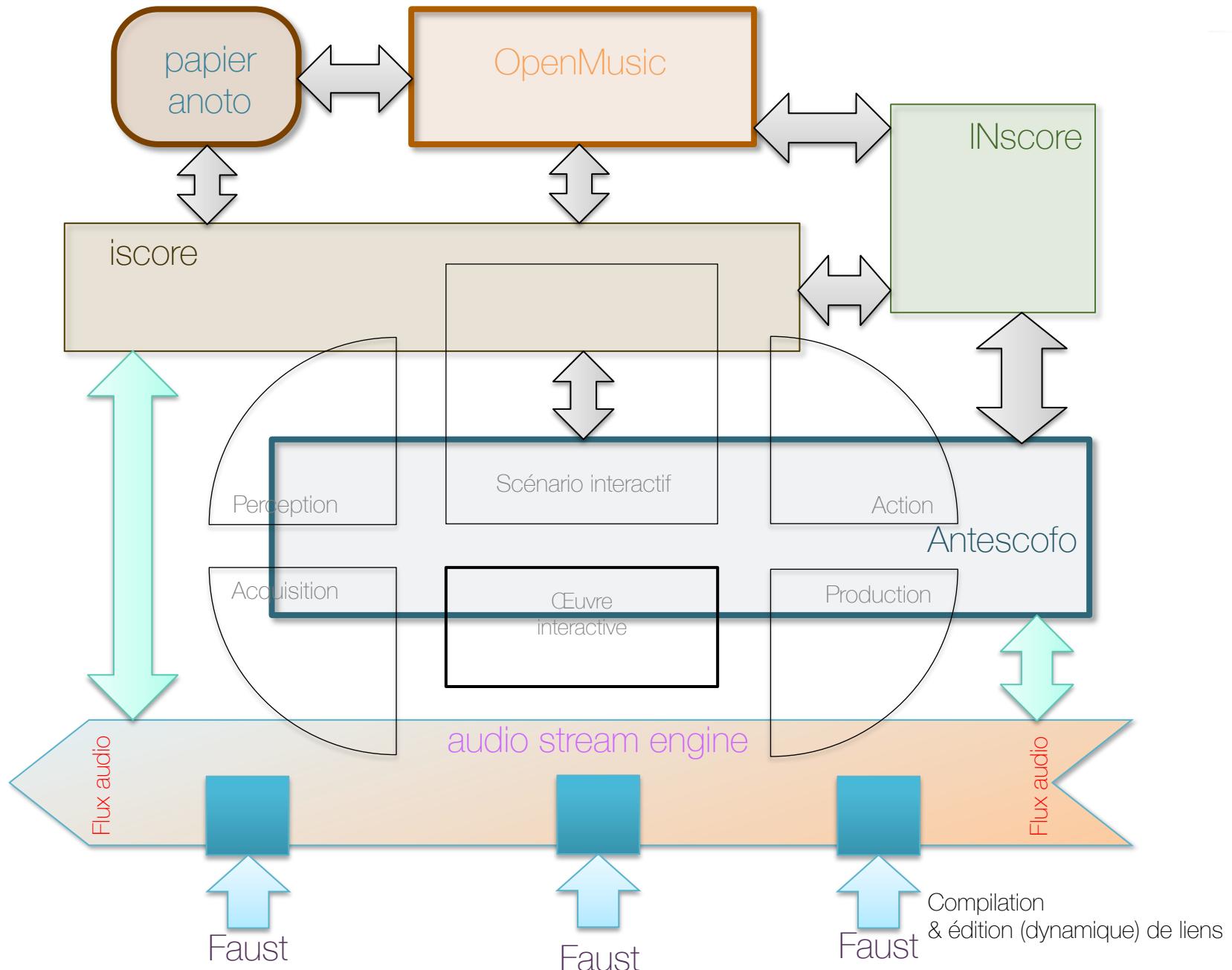
interact

Musiciens & spectateurs

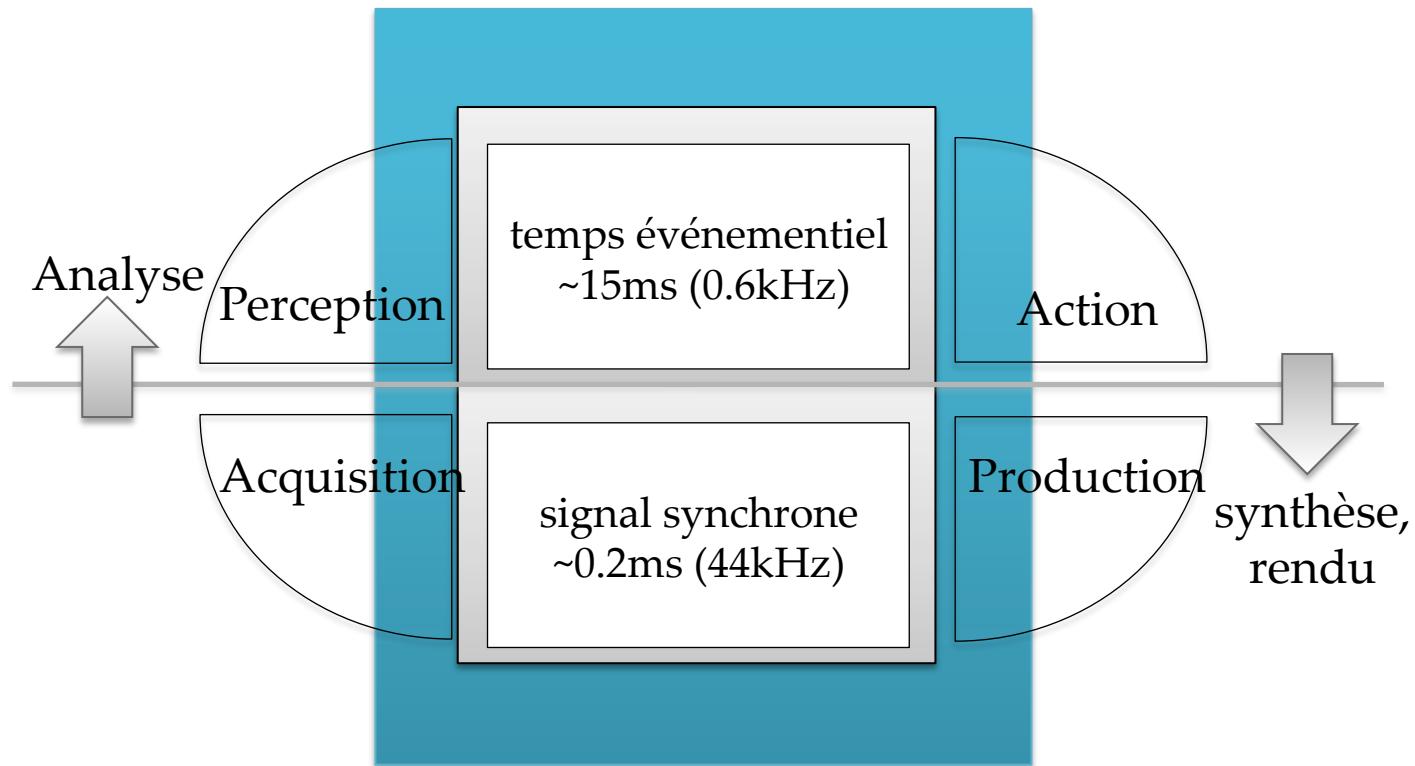
L'écriture de l'interaction :
• traitement du signal
• calculer dans le temps

du signal au symbole
au signal

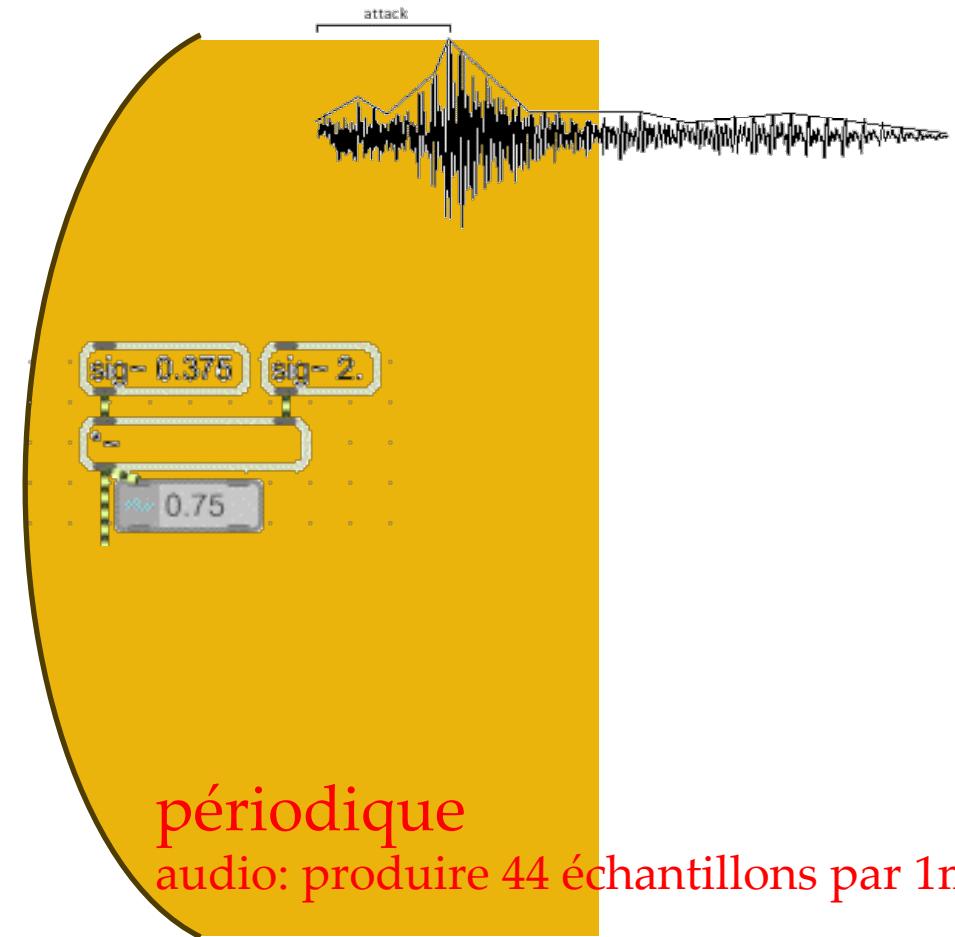
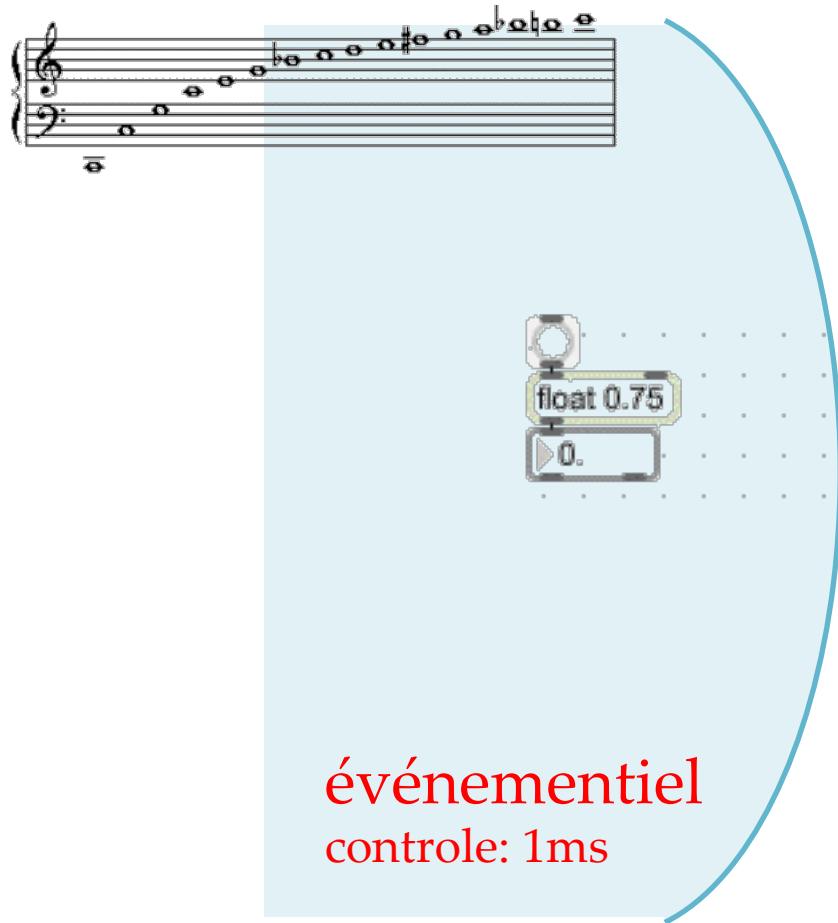
Langages dédiés (en France) pour l'œuvre ouverte interactive



GALS

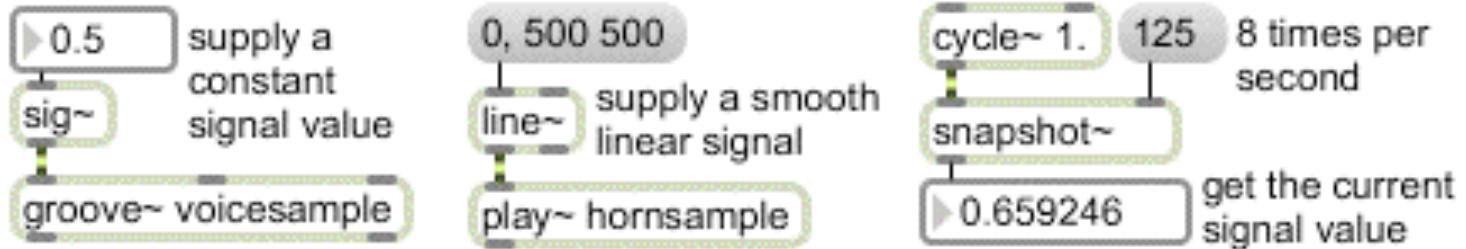


Faire cohabiter les 2 mondes



Faire cohabiter les 2 mondes

sur-echantillonage
tempo variable



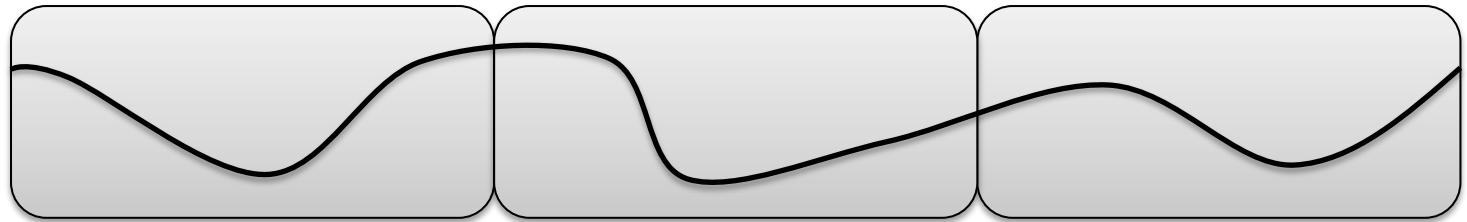
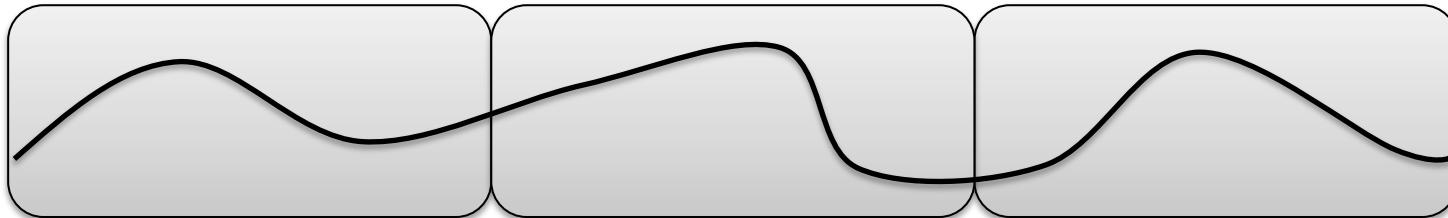
constructeur de signal audio
curve

when
whenever

Faire cohabiter les 2 mondes : traitement précis à l'échantillon des actions

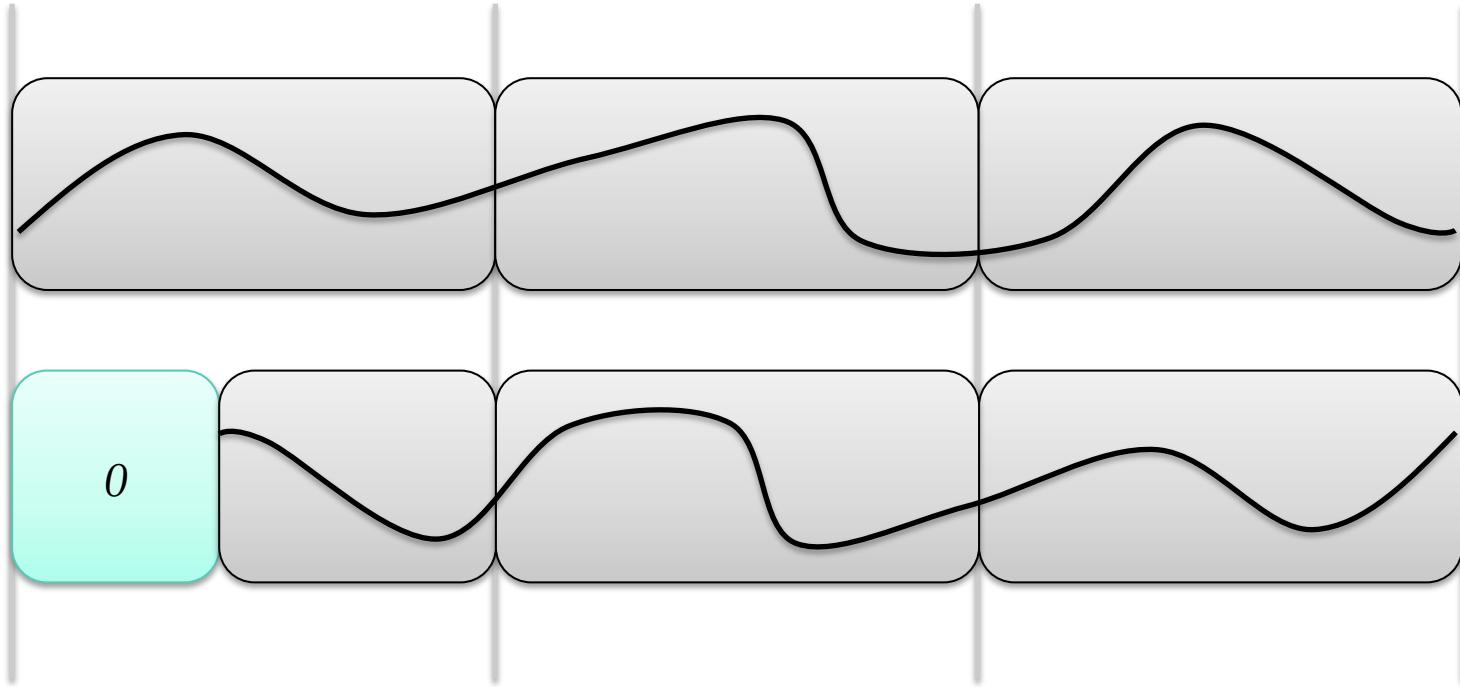
- traiter un échantillon à la fois
- calcul par bloc aligné sur le contrôle
(appel de fonction)
 - le 1er bloc est incomplet
 - pro: les blocs restent alignés
 - cons: la fréquence du contrôle n'est pas préservée
 - démarrer un bloc avec le déclenchement
 - pro: on traite toujours un bloc entier à la fois
 - cons: les blocs ne sont plus alignés

retarder un bloc



- introduit une latence car on doit attendre d'avoir le buffer le plus tardif avant de faire une sortie

Bloc partiel



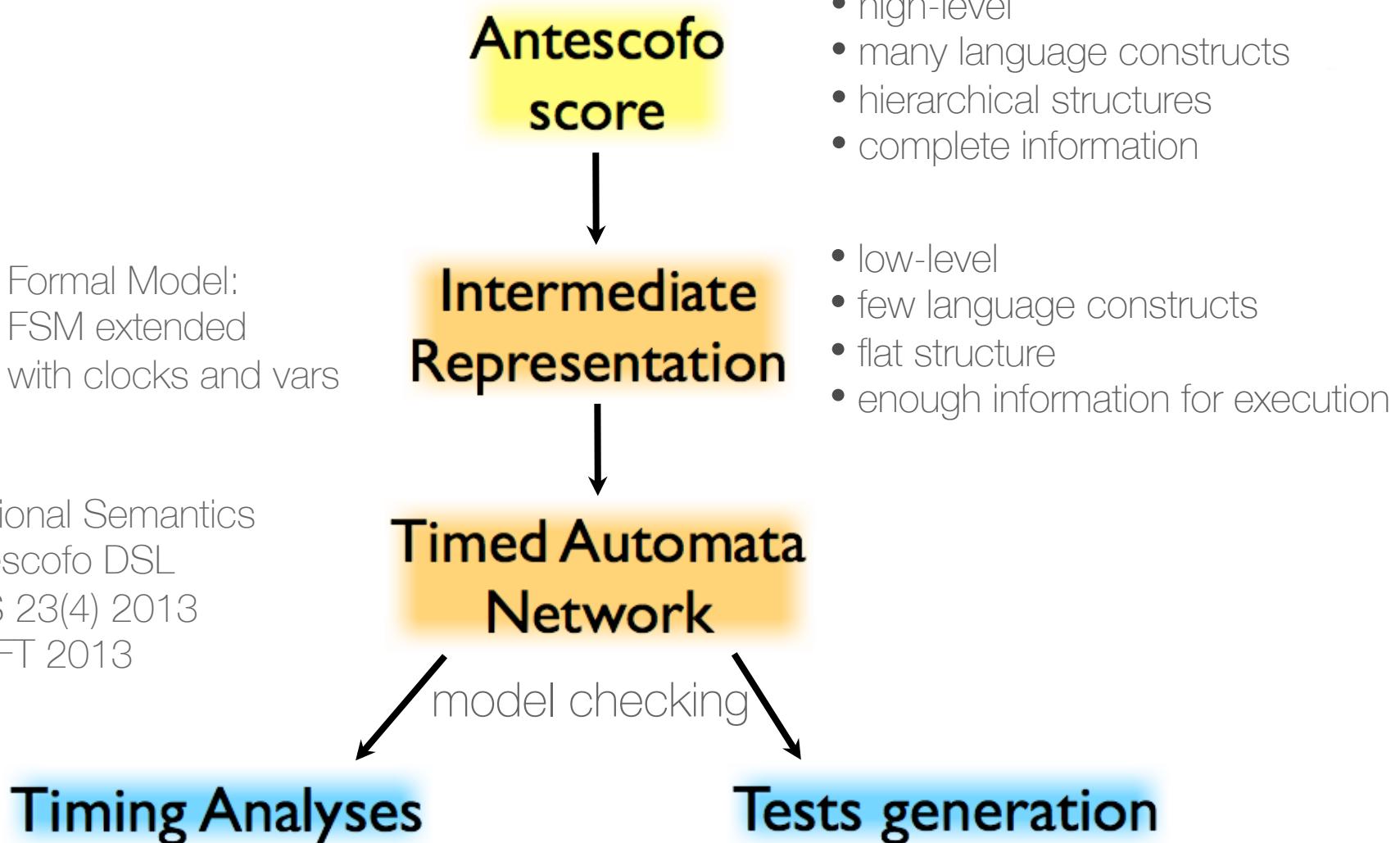
- Les constructeurs de signaux attendent qu'on leur fournissent un contrôle (constant) par bloc
- la même entrée va sortir à des moments différents suivant le retard sur les chemins suivis

Florent Jacquemard, Clément Poncelet and the rest of the Mutant team

VERIFICATION

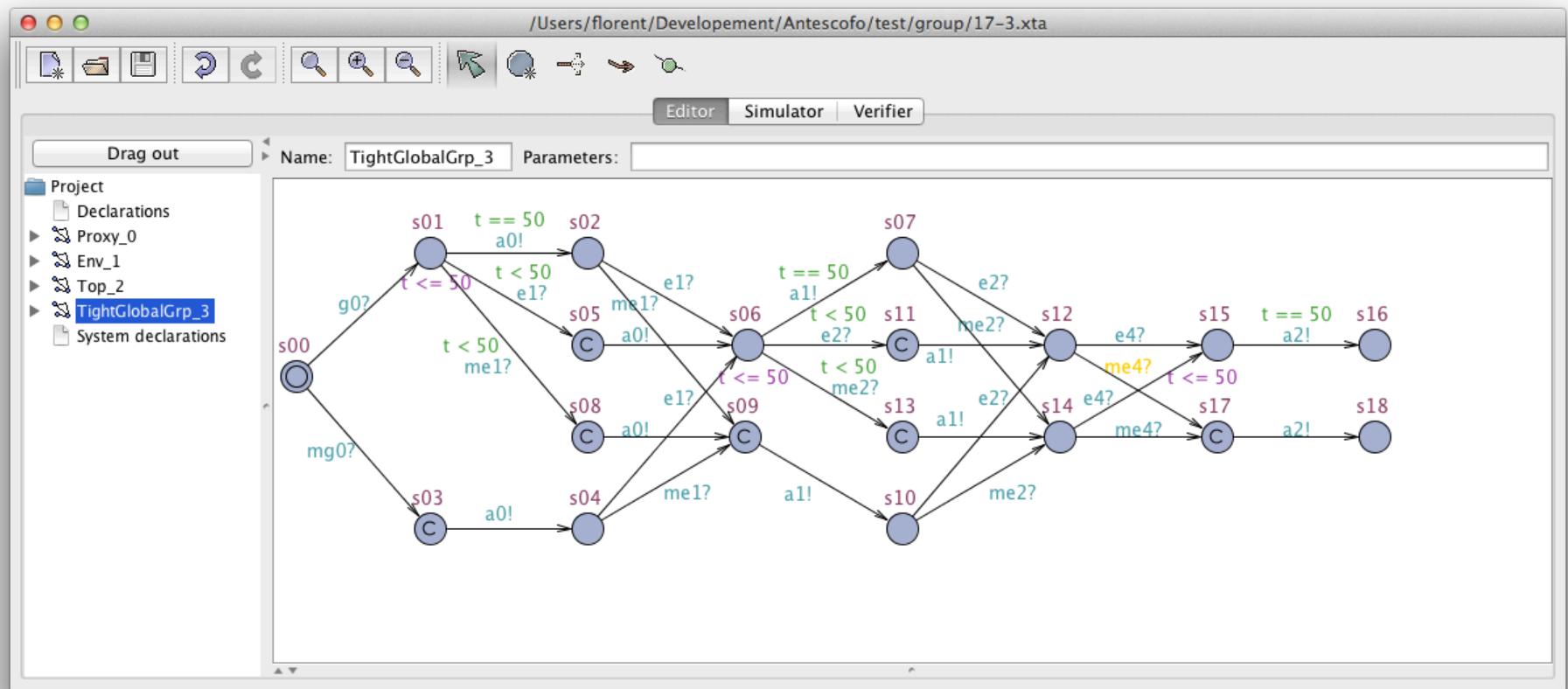
Modeling and analysing Antescofo's time

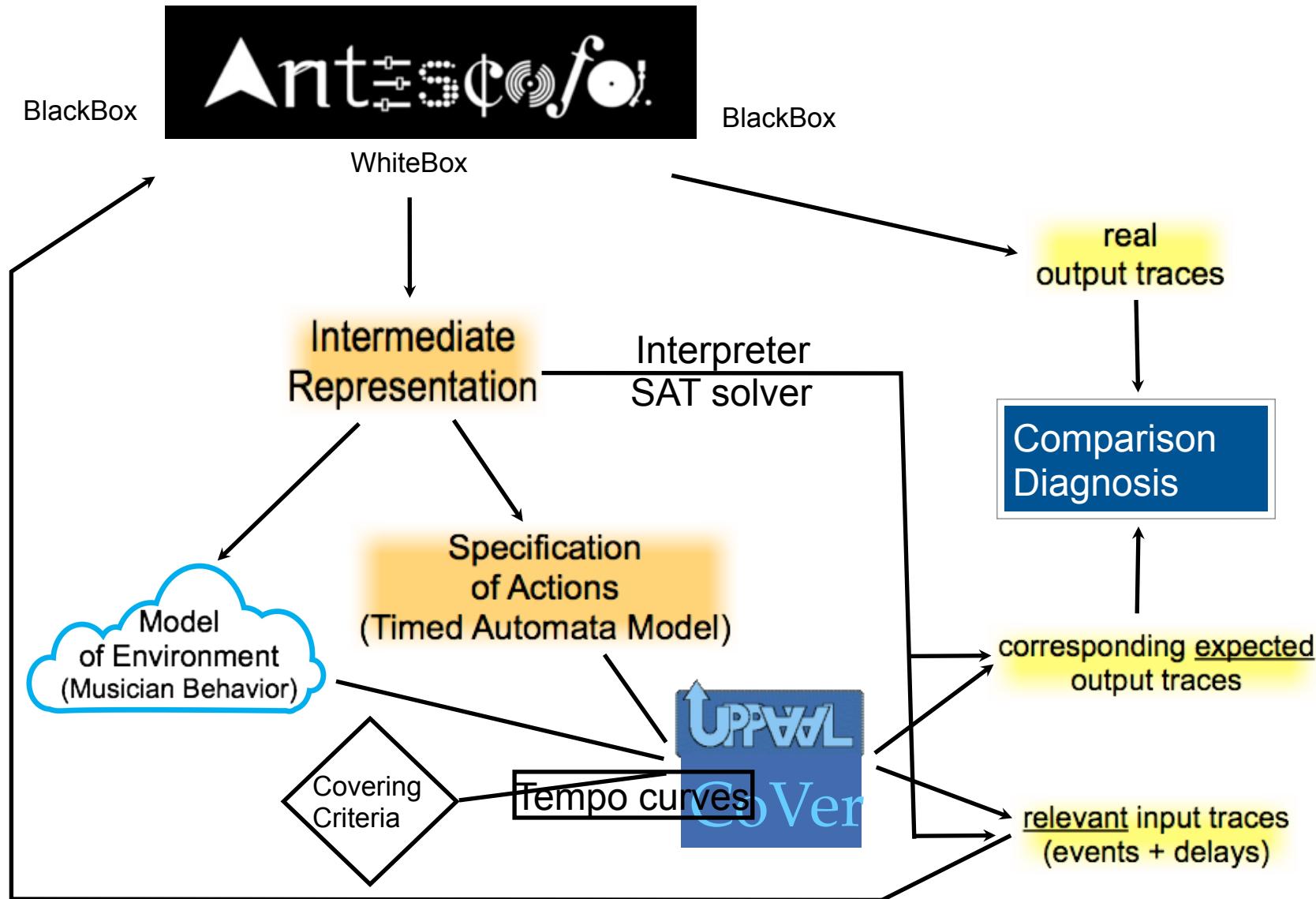
- Model Based Conformance Testing
- quality releases (tools for developpers)
- user assistance (tools for composers / RIMs)
- preservation



Quantitive Evaluation
 of the Robustness of Antescofo Scores
 to Timing Variations at Performance
 ICMC 2013

PhD Clément Poncelet (oct. 2013)





Produire le temps ?

CONCLUSIONS

Produire le temps ?

- une ressource qui se structure formellement
 - instants, simultanéité, succession, durée
 - une forme
- une ressource qu'on
 - coupe en morceau
 - organise, hiérarchise
 - aligne
 - transforme
 - représente spatialement
 - consomme en faisant des calculs
- mais il y a bien d'autres aspects du temps : *vivre le temps*
 - mémoire, apprentissage
 - attente, anticipation
 - conscience
 - ...

Daniel Firmin, 2011

