

Programming Mixed Music in ReactiveML

Guillaume Baudart, ENS

Louis Mandel, ENS

Marc Pouzet, ENS

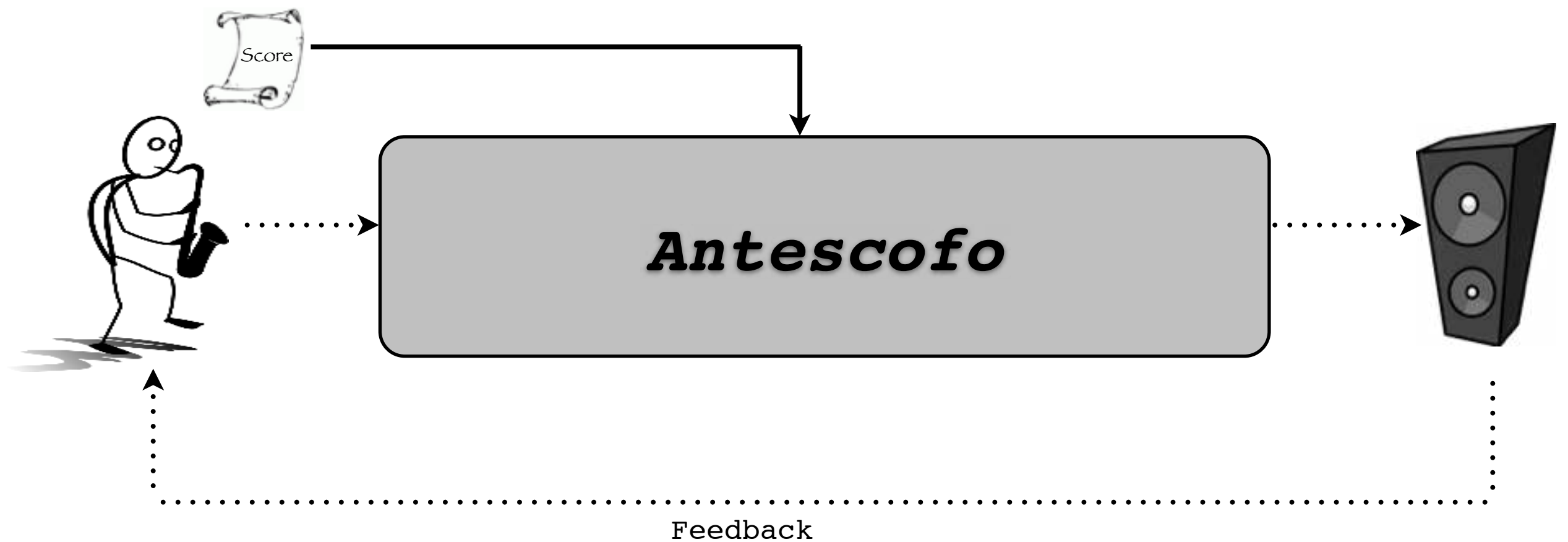
Mixed Music and Antescofo

[Cont 2008]



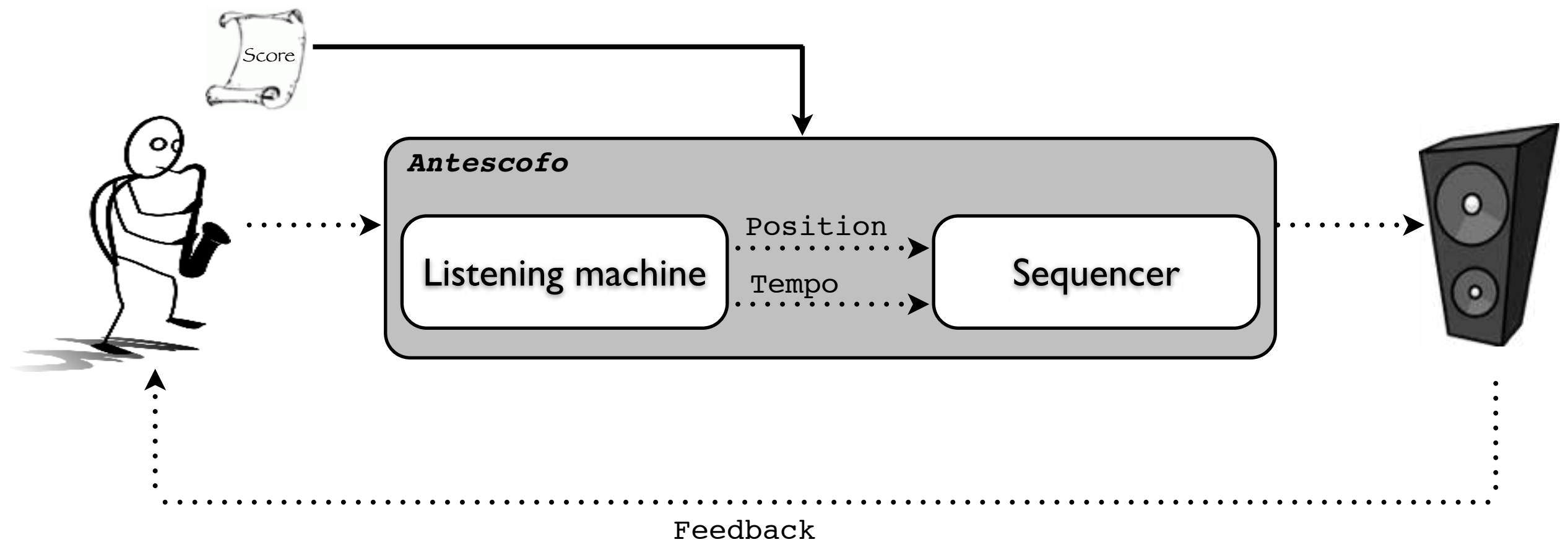
Mixed Music and Antescofo

[Cont 2008]



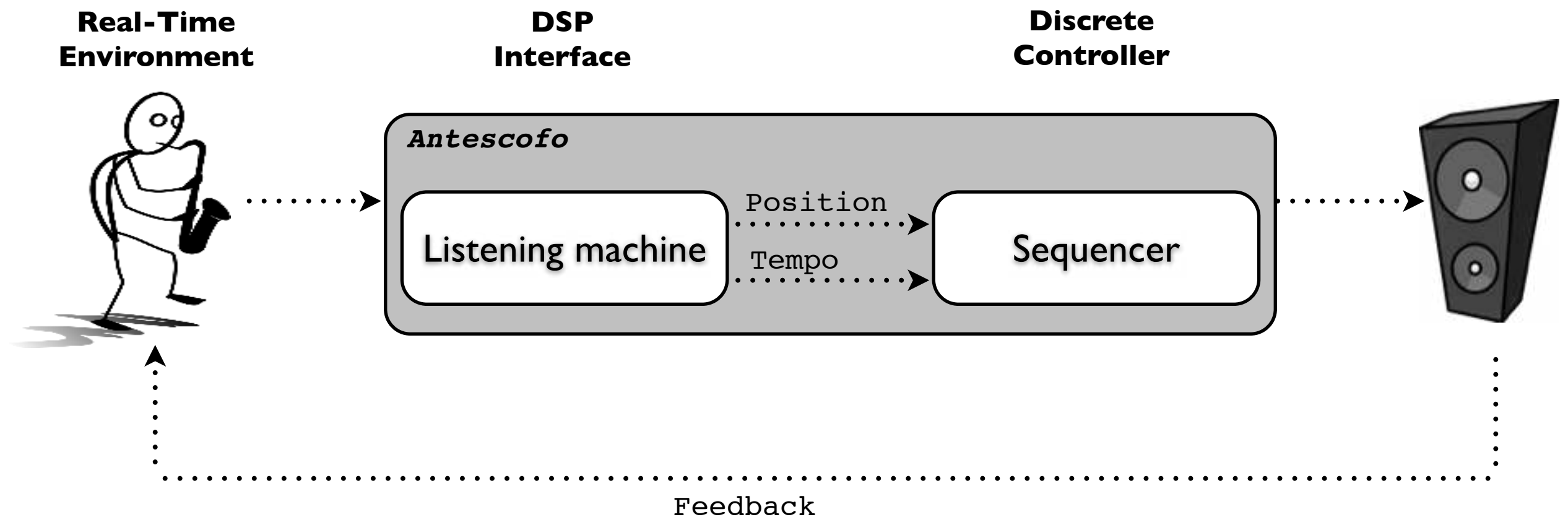
Antescofo Architecture

[Cont 2008]



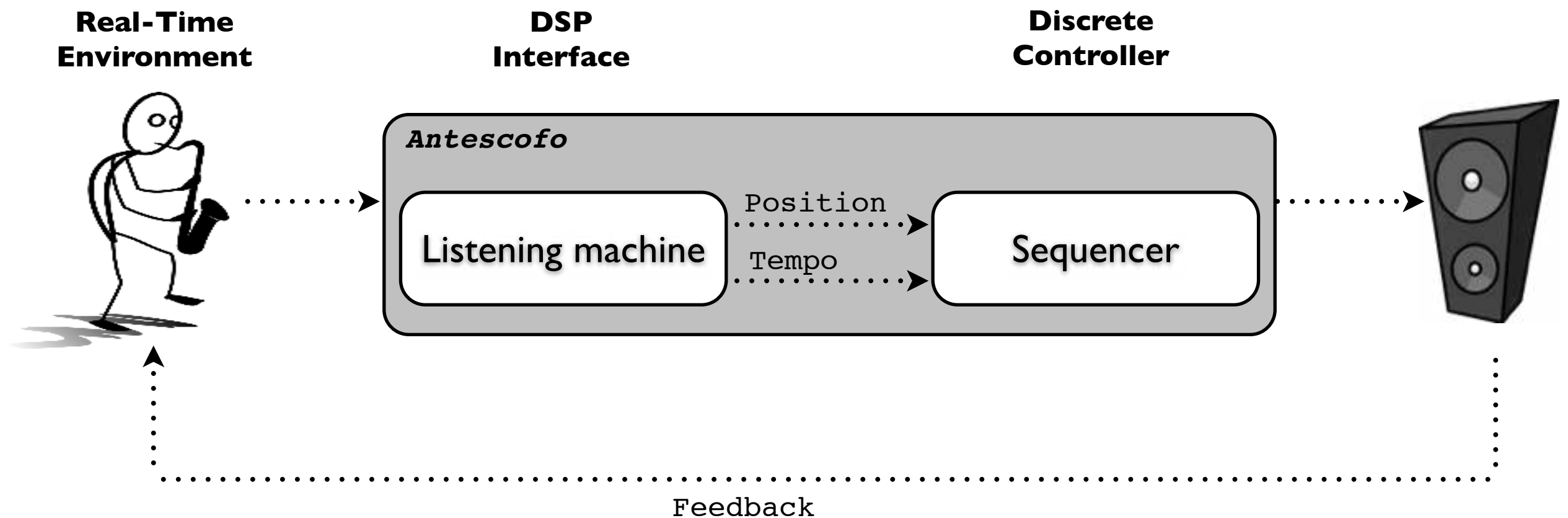
Antescofo Architecture

[Cont 2008]



Antescofo Architecture

[Cont 2008]



The score is a specification of a musical reactive system

The Antescofo Language

Goal: Jointly specify electronic and instrumental parts

Anthèmes II (1994)

Pierre Boulez
(*1925)

Libre brusque
(♩ = 92)

Violon
f *fff* *mf* *ff*

Spatialization
F -11/-18/-18/2.0

Inf. Rev.
reverb. time: 60"

Spatialization
F -11/-18/-18/2.0

Sampl. IR
MIDI: 93 90 85 84 82 80 75 77 76 75 74
reverb. time: 60"

Spatialization
F -11/-18/-18/2.0

Sampler
pizz. *f* = 93 msec.
MIDI: [74 73 70 69 68 67 66 65]

Spatialization
MR -4/-12/-24/2.0

Freq. Shift.

Spatialization

1 **2** **3** **4** **5** **6** **7** **8** **9**

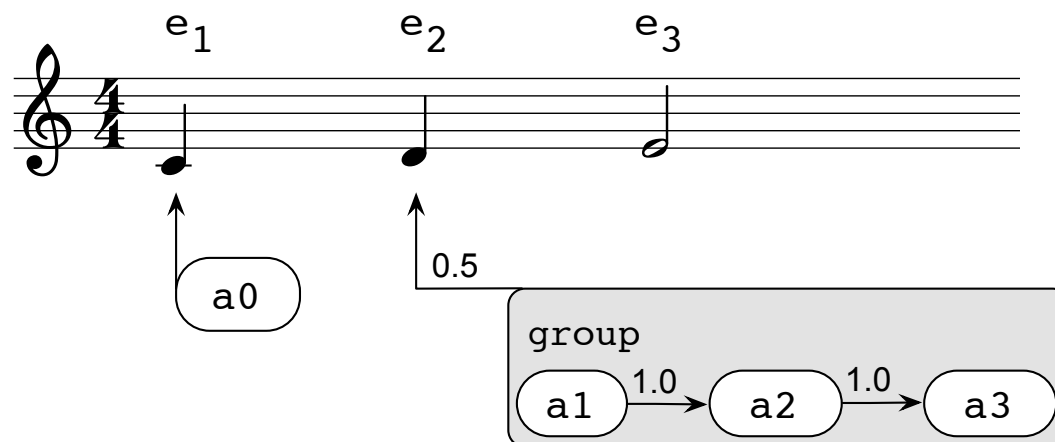
batt. (archet normal)
(♩ = 92) *rall.* (♩ = 66)

New version using antescofo (2008)

The Antescofo Language

Goal: Jointly specify electronic and instrumental parts

[Echeveste et al. 2012]



```
NOTE 60 1.0
0.0 'a_0'

NOTE 62 1.0
0.5 GROUP loose causal
    { 0.0 'a_1'
      1.0 'a_2'
      1.0 'a_3' }

NOTE 64 2.0
```

- Time is relative to the tempo
- Electronic actions are characterized by a delay
- Hierarchical structure: *groups and nested groups*
- Synchronization with the musician : *tight, loose*
- Error handling strategies : *partial, causal*

The Antescofo Language

Goal: Jointly specify electronic and instrumental parts

[Echeveste et al. 2012]

Link with synchronous languages?

NOTE 60 1.0
0.0 'a_0'

NOTE 64 2.0

loose causal
1' 2' 3' }

- Time is relative to the tempo
- Electronic actions are characterized by a delay
- Hierarchical structure: *groups and nested groups*
- Synchronization with the musician : *tight, loose*
- Error handling strategies : *partial, causal*

ReactiveML

The temporal expressiveness of synchronous languages with the power of functional programming

ReactiveML

[Mandel-Pouzet 2005]

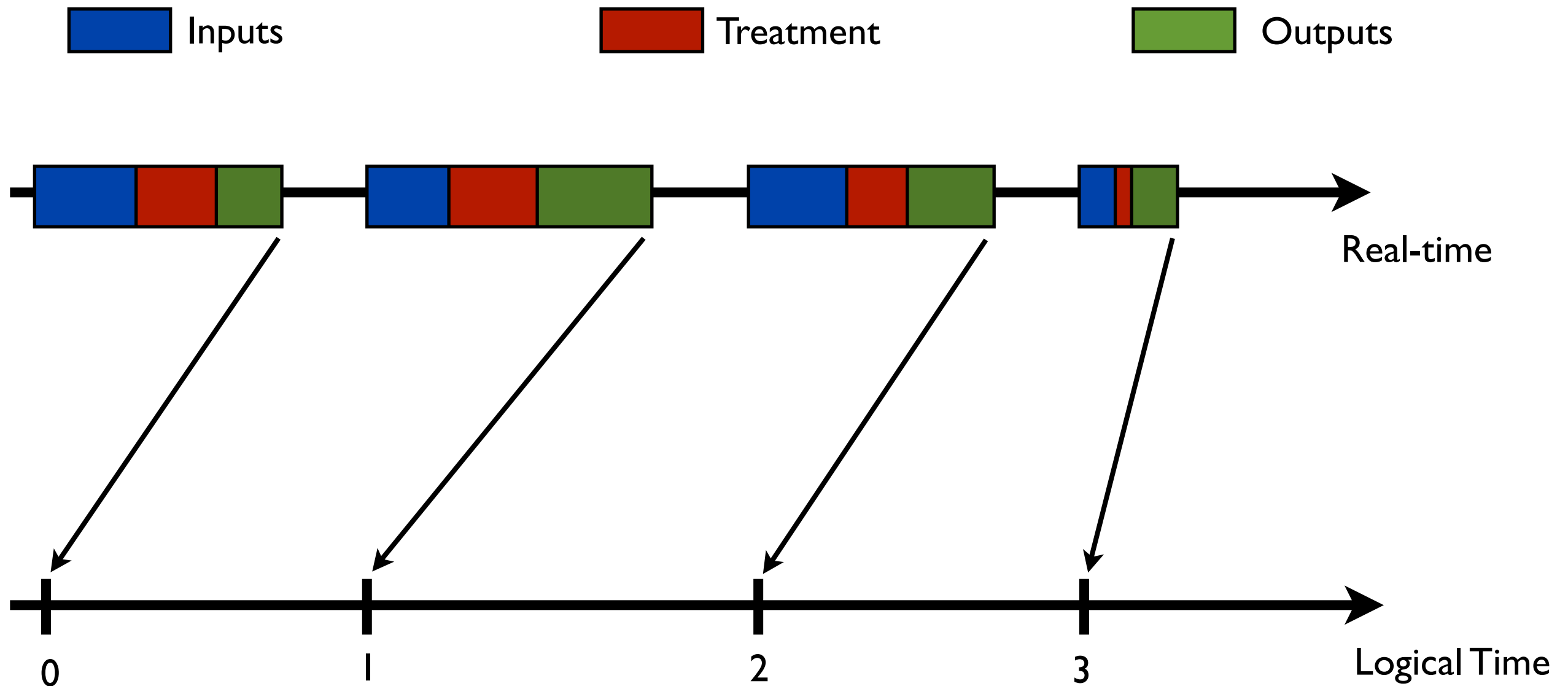
OCaml

- Data structures
- Control structures

Synchronous model of concurrency

- A global logical time
- Parallel composition
- Communication between processes

The Synchronous Hypothesis



The Language

Process

`let process <id> {<pattern>} = <expr>`

*State machines, executed through several instants.
Simple OCaml functions are considered to be instantaneous.*

Basics

Synchronization: `pause`

Execution: `run <expr>`

Composition

Sequence: `<expr> ; <expr>`

Parallelism: `<expr> || <expr>`

Signals

Definition: `signal <id>`

Emission: `emit <id>`

Waiting: `await <id>`

*Broadcast communication
between processes*

First Example

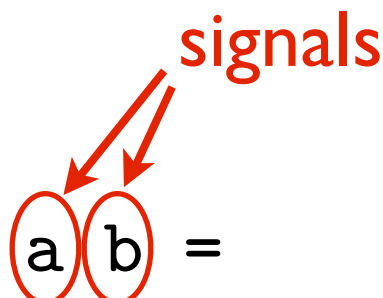
Wait in parallel for the emission of two signals

```
let process simple a b =  
  (await a; print "a")  
  ||  
  (await b; print "b")  
val simple:  
  (unit, unit) event -> (unit, unit) event ->  
  unit process
```

First Example

Wait in parallel for the emission of two signals

```
let process simple (a) (b) =  
  (await a; print "a")  
  ||  
  (await b; print "b")  
val simple:  
  (unit, unit) event -> (unit, unit) event ->  
  unit process
```



First Example

Wait in parallel for the emission of two signals

```
let process simple (a) (b) =  
  (await a; print "a")  
  ||  
  (await b; print "b")  
val simple:  
  (unit, unit) event -> (unit, unit) event ->  
  unit process
```

signals

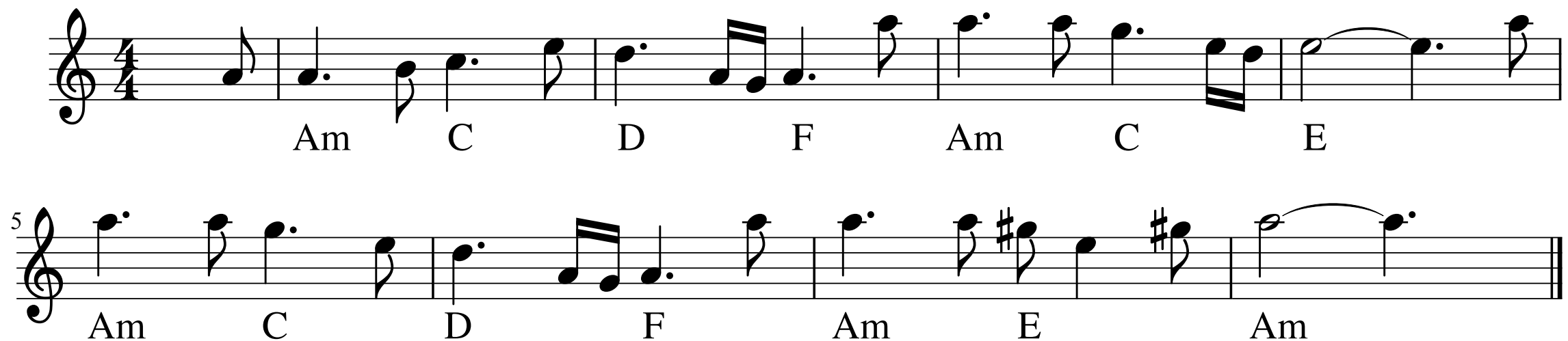
Parallel composition

Live Coding

Modify, correct and interact with the score
during the performance

Automatic Accompaniment

The house of the rising sun



- **Functional programming**
modular definition of the accompaniment
- **Reactive programming**
interaction with the score during the performance

Definitions

1. Define the bass line

```
let bass = [0.0, (A, Min); 2.0, (C, Maj); ...]  
val bass: (delay * chord) list
```

2. Define the accompaniment style

```
let arpeggio chord =  
  ...  
  group Loose Local  
    [0.0, action_note (fond);  
     1.0, action_note (third);  
     2.0, action_note (fifth);}]  
val arpeggio: chord -> asco_event
```

3. Link with the performance

```
let process basic_accomp =  
  run (link asco 2 roots)  
val basic_accomp: unit process
```

Interactions

- **Kill a process when a signal is emitted**
allow to modify the accompaniment
- **Suspend a the execution of a process**
pause and resume a process with a signal
- **Dynamically change the behavior of a process**
switch between different kinds of accompaniment

Kill a Process

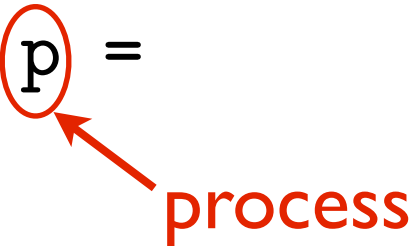
Example of a higher-order process

```
let process killable k p =  
  do  
    run p  
  until k done  
val killable:  
  (unit, unit) event -> unit process ->  
  unit process
```

Kill a Process

Example of a higher-order process

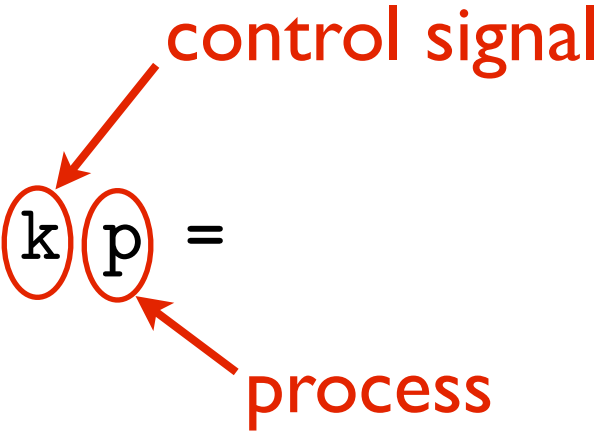
```
let process killable k (p) =  
  do  
    run p  
  until k done  
val killable:  
  (unit, unit) event -> unit process ->  
  unit process
```



Kill a Process

Example of a higher-order process

```
let process killable (k) (p) =  
  do  
    run p  
  until k done  
val killable:  
  (unit, unit) event -> unit process ->  
  unit process
```



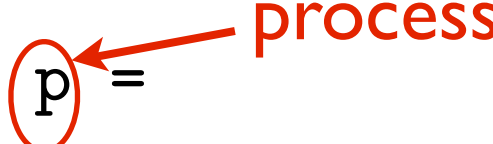
Dynamic Changes

Example of a recursive higher-order process

```
let process rec replaceable replace p =  
  do  
    run p  
  until replace (q) ->  
    run (replaceable replace q)  
done  
val replaceable:  
  (unit process, unit process) event ->  
  unit process -> unit process
```

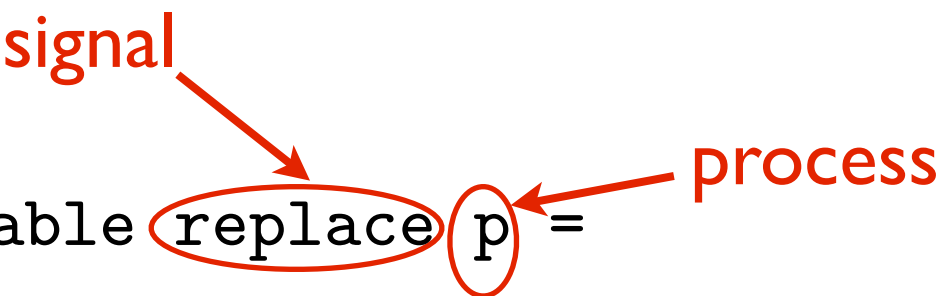

Dynamic Changes

Example of a recursive higher-order process

```
let process rec replaceable replace (p) = 
do
  run p
until replace (q) ->
  run (replaceable replace q)
done
val replaceable:
  (unit process, unit process) event ->
    unit process -> unit process
```

Dynamic Changes

Example of a recursive higher-order process



```
let process rec replaceable replace p =  
  do  
    run p  
  until replace (q) ->  
    run (replaceable replace q)  
done  
val replaceable:  
  (unit process, unit process) event ->  
  unit process -> unit process
```

Dynamic Changes

Example of a recursive higher-order process

```
let process rec replaceable replace p =  
  do  
    run p  
  until replace (q) ->  
    run (replaceable replace q)  
done  
val replaceable:  
  (unit process, unit process) event ->  
    unit process -> unit process
```

signal

process

new behavior
signal can carry processes

New Reactive Behaviors

Example: Steve Reich's Piano Phase

Piano Phase ...

Bob

Alice

Piano Phase,

pour 2 pianos ou 2 marimbas

Steve Reich

$\text{♩} = \text{ca. } 72$

The musical score for 'Piano Phase' by Steve Reich is presented for two staves, Bob and Alice. The tempo is marked as $\text{♩} = \text{ca. } 72$. The score consists of six measures, each with a specific range of measures in parentheses above the staff.

- Measure 1:** Bob (r.h.) and Alice (l.h.) play a sequence of notes. Bob's part is marked *mf* non legato. The range is (x4-8).
- Measure 2:** Bob (r.h.) and Alice (l.h.) play a sequence of notes. Bob's part is marked *mf* non legato. The range is (x12-18).
- Measure 3:** Bob (r.h.) and Alice (l.h.) play a sequence of notes. Bob's part is marked *mf* non legato. The range is (x4-16). Alice's part is marked *mf* non legato. The range is (x16-24).
- Measure 4:** Bob (r.h.) and Alice (l.h.) play a sequence of notes. Bob's part is marked *mf* non legato. The range is (x4-16). Alice's part is marked *mf* non legato. The range is (x16-24).
- Measure 5:** Bob (r.h.) and Alice (l.h.) play a sequence of notes. Bob's part is marked *mf* non legato. The range is (x4-16). Alice's part is marked *mf* non legato. The range is (x16-24).
- Measure 6:** Bob (r.h.) and Alice (l.h.) play a sequence of notes. Bob's part is marked *mf* non legato. The range is (x4-16). Alice's part is marked *mf* non legato. The range is (x16-24).

Performance instructions include 'hold tempo 1', 'accel very slightly', 'fade in', and 'a.v.s.' (a tempo).

Piano Phase ...

Bob

Alice

Piano Phase,
pour 2 pianos ou 2 marimbas

$\text{♩} = \text{ca. } 72$

1 (x4-8) 2 (x12-18) 3 (x16-24) 4 (x16-24) 5 (x16-24) 6 (x16-24)

r.h. l.h. r.h. l.h. r.h. l.h. r.h. l.h. r.h. l.h. r.h. l.h.

mf non legato *mf* non legato *mf* non legato *mf* non legato *mf* non legato *mf* non legato

fade in *mf* *mf* *mf* *mf* *mf* *mf*

hold tempo 1 hold tempo 1 hold tempo 1 hold tempo 1 hold tempo 1 hold tempo 1

accel very slightly a.v.s. a.v.s. a.v.s. a.v.s. a.v.s. a.v.s.

(tempo 1) (tempo 1) (tempo 1) (tempo 1) (tempo 1) (tempo 1)

Synchronization

Piano Phase ...

Bob

Alice

Piano Phase,
pour 2 pianos ou 2 marimbas

$\text{♩} = \text{ca. } 72$

1 (x4-8) 2 (x12-18) 3 (x16-24) 4 (x16-24) 5 (x16-24) 6 (x16-24)

r.h. l.h. r.h. l.h. r.h. l.h. r.h. l.h. r.h. l.h. r.h. l.h.

mf non legato *mf* non legato *mf* non legato *mf* non legato *mf* non legato *mf* non legato

fade in non legato *mf* hold tempo 1 accel. very slightly hold tempo 1 a.v.s. hold tempo 1 a.v.s. hold tempo 1 a.v.s. hold tempo 1 a.v.s.

(tempo 1) (tempo 1) (tempo 1) (tempo 1) (tempo 1) (tempo 1)

The image shows a musical score for 'Piano Phase' by Steve Reich. It is written for two pianos or two marimbas. The tempo is marked as approximately 72 beats per minute. The score is divided into six measures, each with a specific range of repetitions (e.g., 1 (x4-8), 2 (x12-18), 3 (x16-24), 4 (x16-24), 5 (x16-24), 6 (x16-24)). The notation includes right-hand (r.h.) and left-hand (l.h.) parts. The first measure is marked 'mf non legato'. The second measure is marked 'fade in non legato'. The third measure is marked 'mf non legato'. The fourth measure is marked 'hold tempo 1'. The fifth measure is marked 'accel. very slightly hold tempo 1'. The sixth measure is marked 'a.v.s. hold tempo 1'. The score is written for two staves, Bob and Alice, with complex rhythmic patterns and tempo markings.

Desynchronization

Piano Phase ...

Bob

Alice

Piano Phase,
pour 2 pianos ou 2 marimbas

$\text{♩} = \text{ca. } 72$

1 (x4 - 8) r.h.
l.h. *mf* non legato

2 (x12 - 18) r.h.
l.h. fade in non legato

4 (x16 - 24) (tempo 1) a.v.s. hold tempo 1

5 (x16 - 24) (tempo 1) a.v.s. hold tempo 1

(x4 - 16)
hold tempo 1

accel
very slightiy

Steve Reich

(x4 - 16) (tempo 1) a.v.s.

24 (tempo 1) a.v.s.

(tempo 1) a.v.s.

Desynchronization

Piano Phase ...

Bob

Alice

Piano Phase,
pour 2 pianos ou 2 marimbas

$\text{♩} = \text{ca. } 72$

Steve Reich

The musical score for 'Piano Phase' by Steve Reich is presented for two staves, labeled 'Bob' (top) and 'Alice' (bottom). The score consists of six measures, each with a measure number and a range in parentheses: 1 (x4-8), 2 (x12-18), 3 (x16-24), 4 (x16-24), 5 (x16-24), and 6 (x16-24). The top staff (Bob) is marked 'r.h.' and 'mf non legato'. The bottom staff (Alice) is marked 'l.h.' and 'mf non legato'. A red arrow points to the start of measure 3 on the top staff. The score includes various performance instructions: 'fade in' under the first measure of the bottom staff, 'hold tempo 1' under the first measure of the top staff, 'accel very slightly' under the first measure of the bottom staff, and 'a.v.s.' (ad libitum) under the first measure of the bottom staff. The tempo is marked '(tempo 1)' at the end of measures 3, 4, 5, and 6.

Piano Phase ...

Bob

Alice

Piano Phase,
pour 2 pianos ou 2 marimbas

$\text{♩} = \text{ca. } 72$

1 (x4-8) 2 (x12-18) (x4-16) 3 (x16-24) (x4-16)

r.h. l.h. r.h. l.h. r.h. l.h. r.h. l.h.

mf non legato *mf* non legato *mf* non legato *mf* non legato

fade in *mf* *mf* *mf* *mf*

hold tempo 1 hold tempo 1 hold tempo 1 hold tempo 1

accel very slightly

(tempo 1) (tempo 1) (tempo 1) (tempo 1)

a.v.s. a.v.s. a.v.s. a.v.s.

4 (x16-24) 5 (x16-24) 6 (x16-24) (x4-16) (x4-16)

(tempo 1) (tempo 1) (tempo 1) (tempo 1) (tempo 1)

hold tempo 1 hold tempo 1 hold tempo 1 hold tempo 1 hold tempo 1

a.v.s. a.v.s. a.v.s. a.v.s. a.v.s.

The image shows a musical score for 'Piano Phase' by Steve Reich. It consists of two staves, one for 'Bob' (top) and one for 'Alice' (bottom). The score is written for two pianos or two marimbas. The tempo is marked as 'ca. 72' (approximately 72 beats per minute). The score is divided into six measures, each with a number and a range of beats in parentheses: 1 (x4-8), 2 (x12-18), 3 (x16-24), 4 (x16-24), 5 (x16-24), and 6 (x16-24). The notation includes right-hand (r.h.) and left-hand (l.h.) parts, dynamics (mf), and tempo markings (hold tempo 1, accel very slightly, tempo 1). A red box highlights the end of the first measure, showing a repeat sign and the tempo marking '(tempo 1)'. The score also includes 'a.v.s.' (ad libitum) markings and 'fade in' instructions.

Piano Phase ...

Piano Phase,

pour 2 pianos ou 2 marimbas

$\text{♩} = \text{ca. } 72$

Steve Reich

Bob

Alice

4 (x 16 - 24) 5 (x 16 - 24) 6 (x 16 - 24)

(tempo 1) (tempo 1) (tempo 1)

hold tempo 1 a.v.s. hold tempo 1 a.v.s. hold tempo 1 a.v.s.

Piano Phase ...

Bob

Alice

Piano Phase,
pour 2 pianos ou 2 marimbas

$\text{♩} = \text{ca. } 72$

1 (x4-8) 2 (x12-18) (x4-16) 3 (x16-24) (x4-16)

r.h. l.h. mf non legato r.h. l.h. mf non legato

fade in non legato

hold tempo 1 accel very slightly hold tempo 1 a.v.s.

4 (x16-24) 5 (x4-16) 6 (x16-24) (x4-16) (x4-16)

(tempo 1) (tempo 1) (tempo 1)

hold tempo 1 a.v.s. hold tempo 1 a.v.s. hold tempo 1 a.v.s.

The image shows a musical score for 'Piano Phase' by Steve Reich. It is written for two pianos or two marimbas. The tempo is marked as approximately 72 beats per minute. The score is divided into two systems, each with two staves. The first staff is labeled 'Bob' and the second 'Alice'. The music consists of a single melodic line that is played by both instruments, with the phase of the second instrument gradually shifting relative to the first. The score includes various musical notations such as dynamics (mf), articulation (non legato), and tempo markings (hold tempo 1, accel very slightly). There are also repeat signs and measures marked with 'a.v.s.' (a volte). The score is numbered 1 through 6, with some measures marked with 'x' and a range of measures (e.g., x4-8, x12-18, x16-24, x4-16).

Piano Phase ...

Piano Phase,

pour 2 pianos ou 2 marimbas

$\text{♩} = \text{ca. } 72$

Steve Reich

Bob

Alice

1 (x4-8) 2 (x12-18) (x4-16) 3 (x16-24) (x4-16)

r.h. l.h. mf non legato

hold tempo 1 (tempo 1)

mf non legato

fade in

acc. very slightly hold tempo 1 a.v.s.

The image shows a musical score for the song "The Rose Tree". It consists of two systems of music, each with a treble and bass staff. The first system is labeled "4 (x 16 - 24)" and the second system is labeled "5 (x 16 - 24)". Both systems include a tempo change to "tempo 1" and a "hold tempo 1" instruction. A red oval highlights the first measure of the second system, which contains a treble clef, a key signature of one sharp (F#), and a 4/4 time signature. The melody in this measure is: G4 (quarter), A4 (quarter), B4 (quarter), C5 (quarter), B4-A4 (beamed eighth notes), G4 (quarter), F#4 (quarter), E4 (quarter), D4 (half).

Piano Phase ...

Bob

Alice

Piano Phase,

pour 2 pianos ou 2 marimbas

Steve Reich

$\text{♩} = \text{ca. } 72$

The musical score for 'Piano Phase' by Steve Reich is displayed. It consists of two staves, labeled 'Bob' and 'Alice'. The score is written for two pianos or two marimbas. The tempo is marked as 'ca. 72' (approximately 72 beats per minute). The score is divided into four measures, each with a specific duration: 1 (x4-8), 2 (x12-18), 3 (x16-24), and 4 (x16-24). The notation includes various musical symbols such as treble clefs, notes, rests, and dynamic markings like 'mf' (mezzo-forte) and 'non legato'. The score also includes instructions like 'hold tempo 1' and 'a.v.s.' (ad libitum). The score is presented in a way that highlights the phase relationships between the two staves.

Problem:

We do not want to compute a priori
when resynchronizations will occur

... in Mixed Music

Live musician

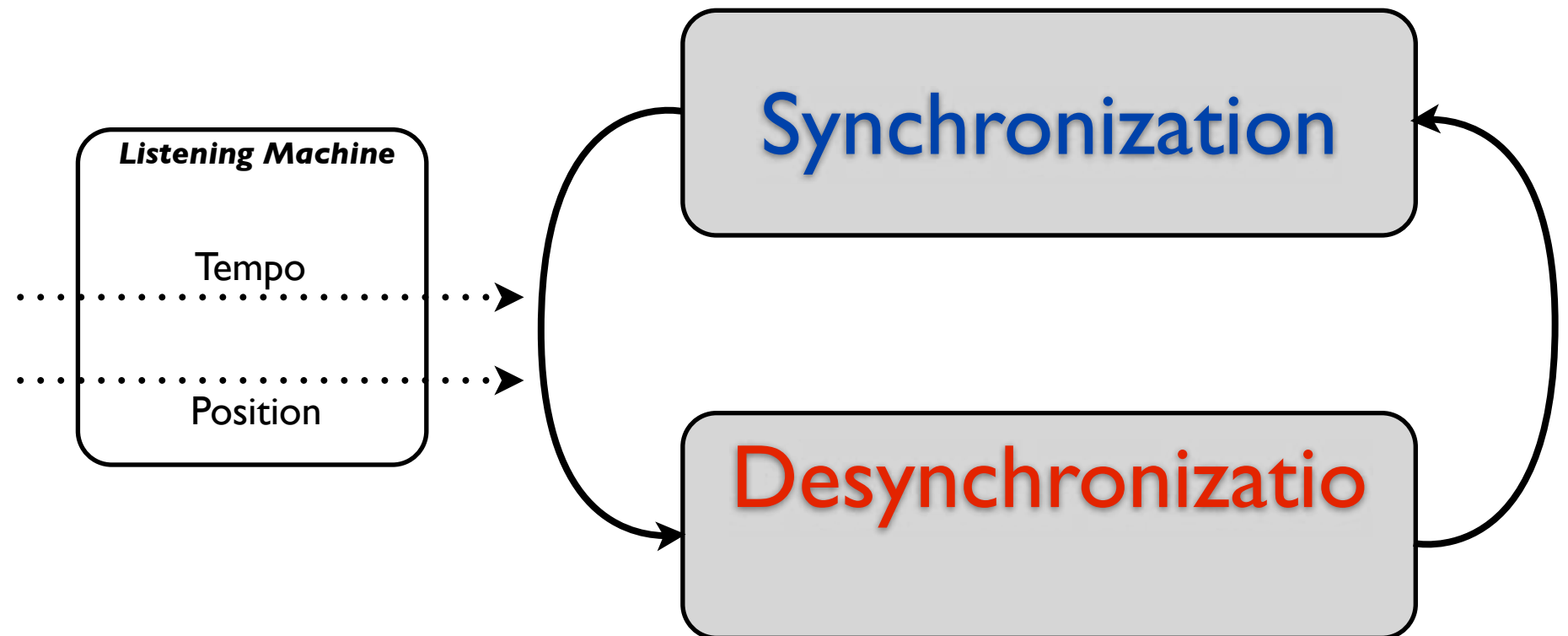
Plays the constant speed part



Bob

Electronic

Handles the desynchronization



Alice

... in Mixed Music

Live musician

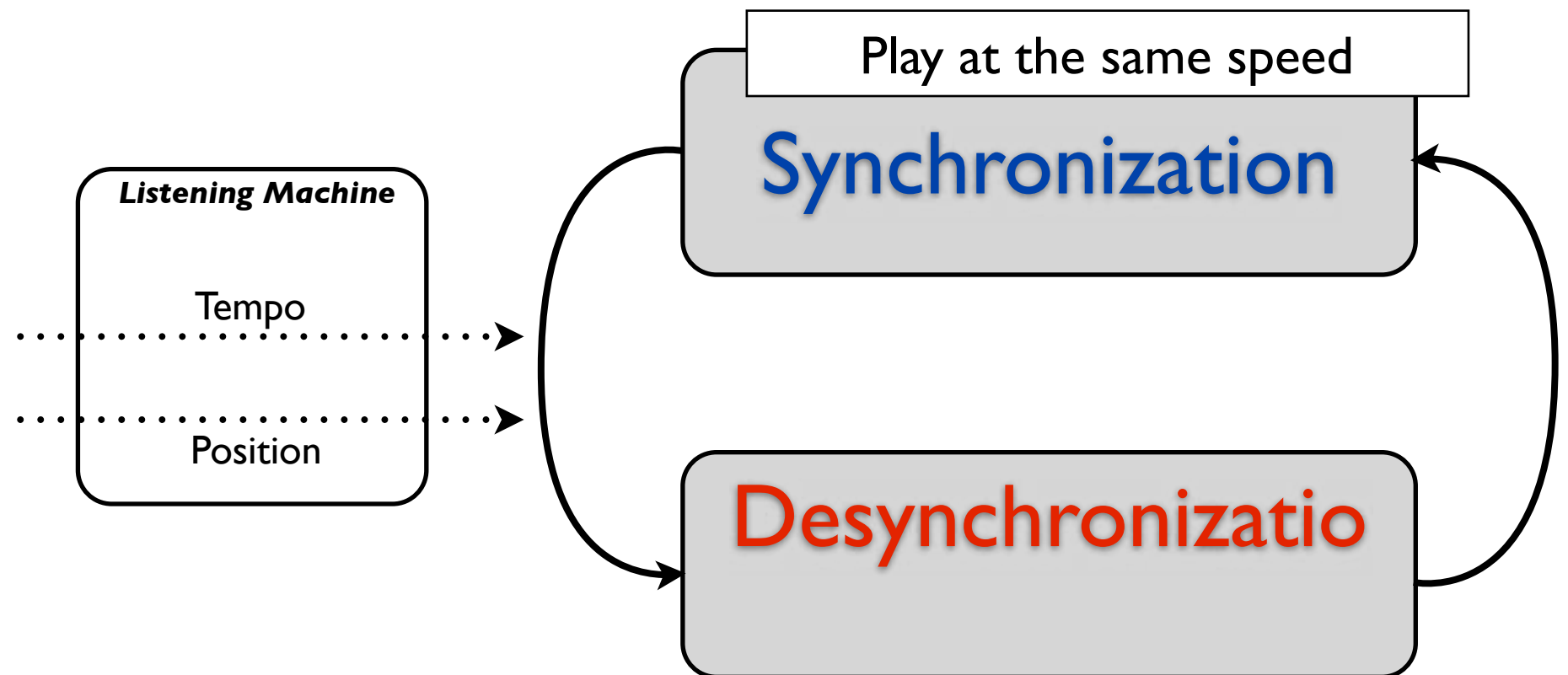
Plays the constant speed part



Bob

Electronic

Handles the desynchronization



Alice

... in Mixed Music

Live musician

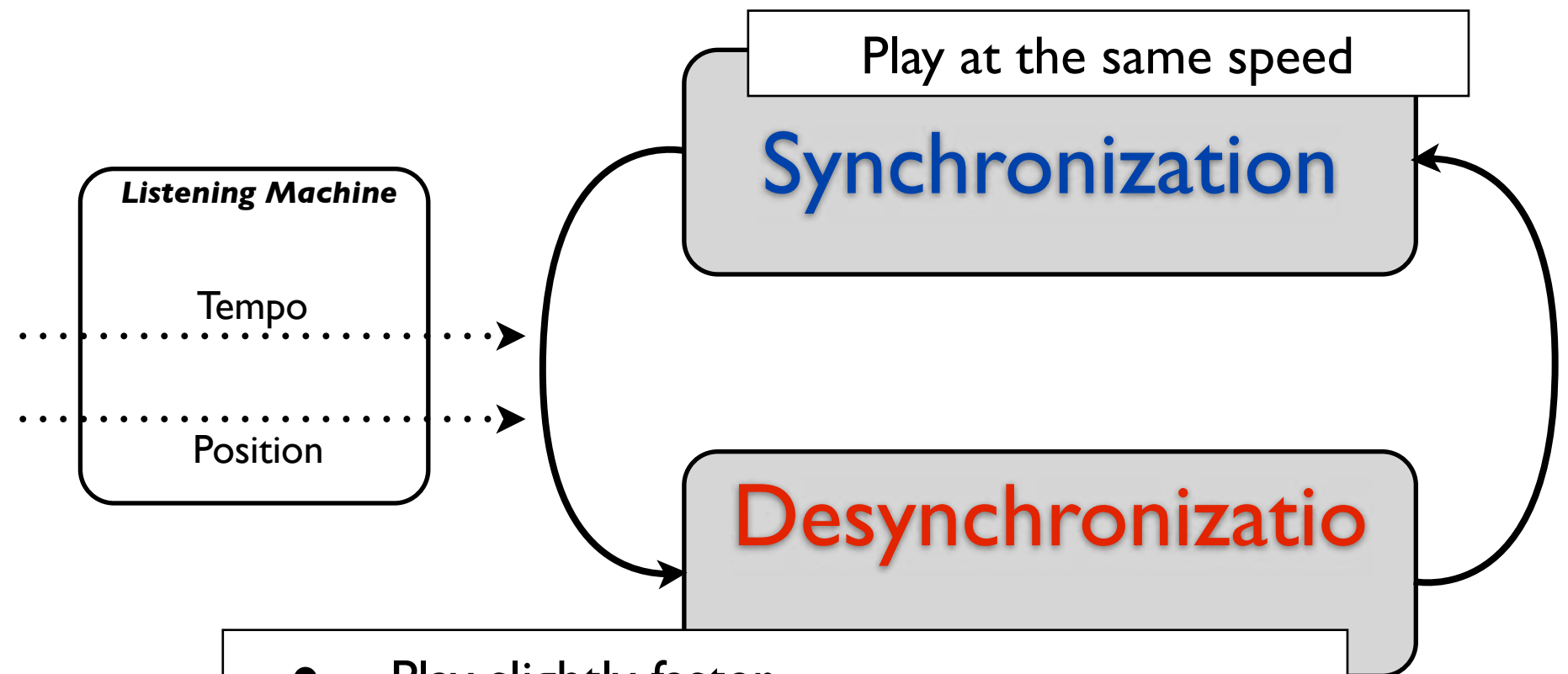
Plays the constant speed part



Bob

Electronic

Handles the desynchronization



- Play slightly faster
- Track the first note of Bob
- Resynchronize when the k-th note of Alice is close enough of the first note of Bob

Implementation

Two phases:
Synchronization
Desynchronization

```
let piano_phase sync desync first_note kth_note =  
  let rec process piano_phase k =  
    let ev = last_event asco in  
    run (melody ev 4 0.25 first_note);  
    emit desync;  
    do  
      let ev = last_event asco in  
      run (melody (ev+1) 16 0.2458 first_note) ||  
      run (track asco k kth_note) ||  
      run (compare asco first_note kth_note sync 0.05)  
    until sync done;  
    run (piano_phase ((k + 1) mod 12))  
  in  
  piano_phase 1  
in
```

Implementation

Synchronization

*Play the melody four times
and follow the tempo*

*Emit the signal `desync` after
four iterations of the melody*

```
let piano_phase sync desync first_note kth_note =  
  let rec process piano_phase k =  
    let ev = last_event asco in  
    run (melody ev 4 0.25 first_note);  
    emit desync;  
  do  
    let ev = last_event asco in  
    run (melody (ev+1) 16 0.2458 first_note) ||  
    run (track asco k kth_note) ||  
    run (compare asco first_note kth_note sync 0.05)  
  until sync done;  
  run (piano_phase ((k + 1) mod 12))  
in  
  piano_phase 1  
in
```

Implementation

Desynchronization

*Play slightly faster
and emit the signal `first_note`
whenever the first note is played*

Track the k -th note of the musician

*Compare the emission of signals
`kth_note` and `first_note` and emit
`sync` when they are close enough*

```
let piano_phase sync desync first_note kth_note =  
  let rec process piano_phase k =  
    let ev = last_event asco in  
    run (melody ev 4 0.25 first_note);  
    emit desync;  
    do  
      let ev = last_event asco in  
      run (melody (ev+1) 16 0.2458 first_note) ||  
      run (track asco k kth_note) ||  
      run (compare asco first_note kth_note sync 0.05)  
    until sync done;  
    run (piano_phase ((k + 1) mod 12))  
  in  
  piano_phase 1  
in
```

Why ReactiveML?

- **A synchronous language**
expressiveness for time and events
- **Functional, typed language, on top of OCaml**
recursion and higher order processes
- **Efficient implementation**
no busy waiting
- **Dynamical features**
dynamical creation of processes

In Practice

- **Embedding the Antescofo language**
new implementation of the sequencer
using the actual antescofo listening machine
- **Extend the Antescofo language**
functional and reactive programming
- **A tool for prototyping new features**
reactive behaviors, live coding, new attributes
- **Link with other media**
graphical interface, top-level, ...

To Continue...



www.reactiveml.org/farm13

References

[Mandel-Pouzet 2005] L. Mandel and M. Pouzet. *ReactiveML: a reactive extension to ML*. In Proceedings of the International Conference on Principles and Practice of Declarative Programming, 2005.

[Mandel-Plateau 2008] L. Mandel and F. Plateau. *Interactive programming of reactive systems*. In Proceedings of Model-driven High-level Programming of Embedded Systems, 2008.

[Cont 2008] A. Cont. *Antescofo: Anticipatory synchronization and control of interactive parameters in computer music*. In International Computer Music Conference, 2008.

[Echeveste et al 2012] J. Echeveste, A. Cont, J.-L. Giavitto, and F. Jacquemard. *Operational semantics of a domain specific language for real time musician-computer interaction*. Journal of Discrete Event Dynamic Systems, 2013.