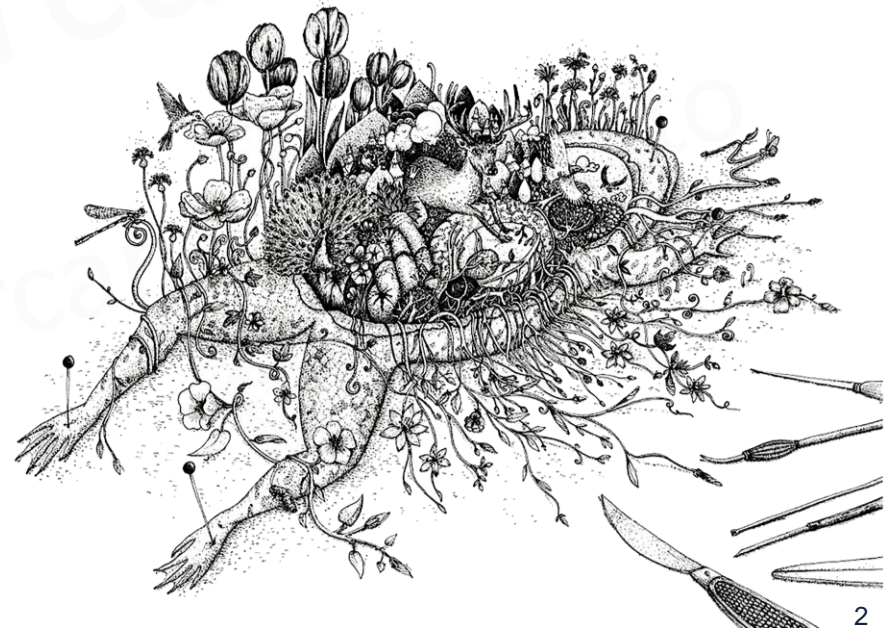




AntesCof

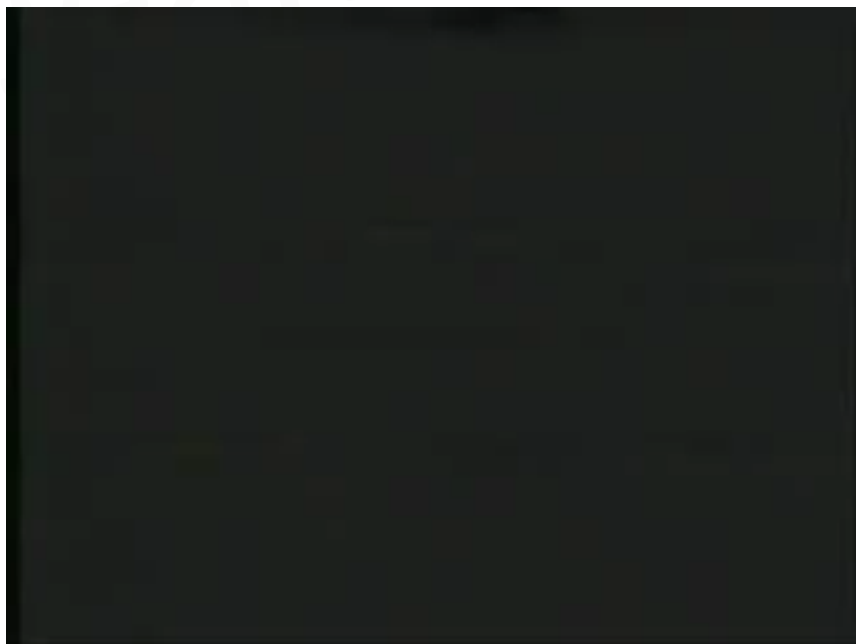
entre composition et performance

- chronologiquement
- quelques exemples
- quelques dichotomies à (dé)tricoter
- écritures de l'électronique ?

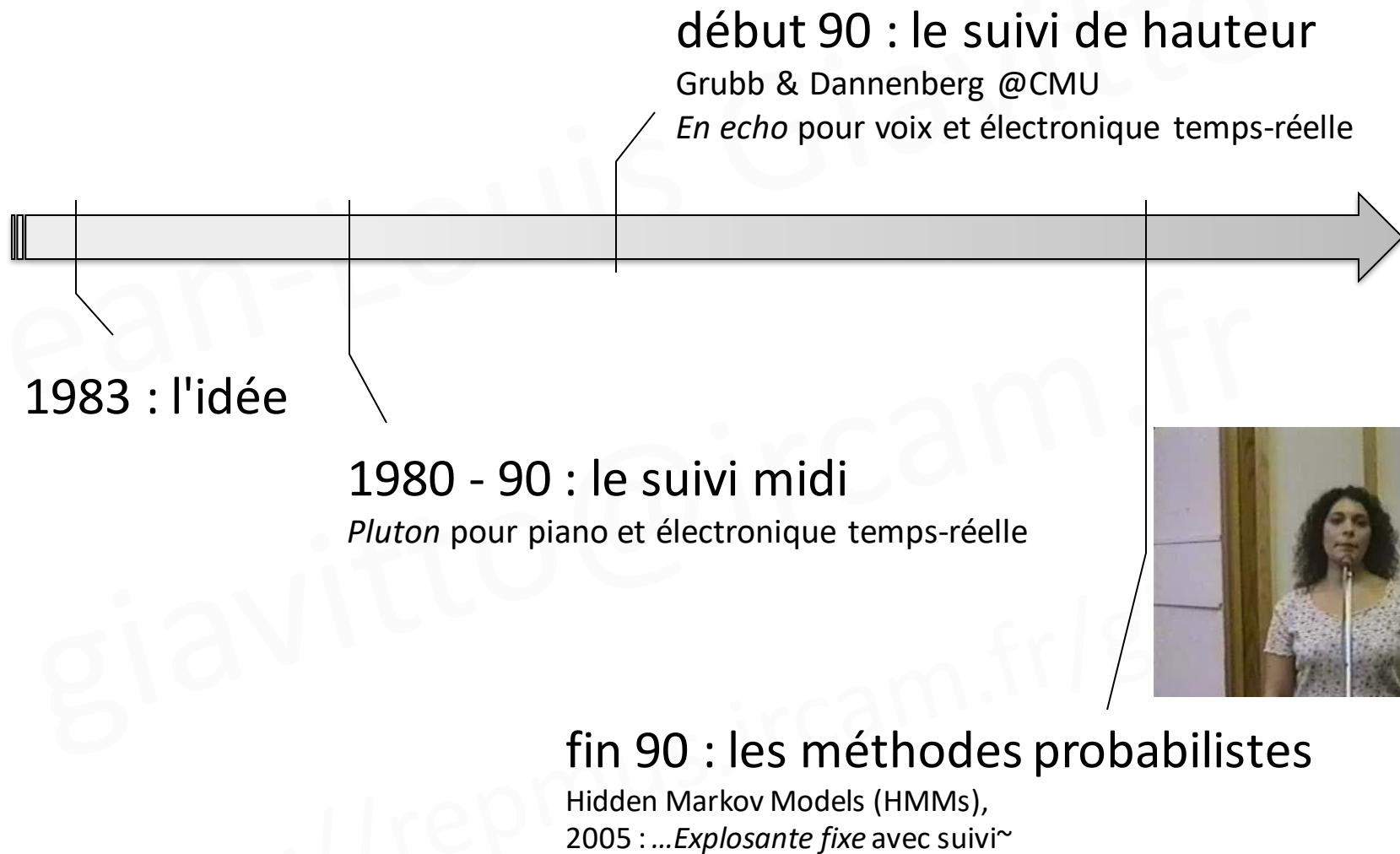


Suivi de partition / Accompagnement automatique

- En 1983, la même année
 - Barry Vercoe à l'IRCAM
 - Roger Dannenberg à CMU

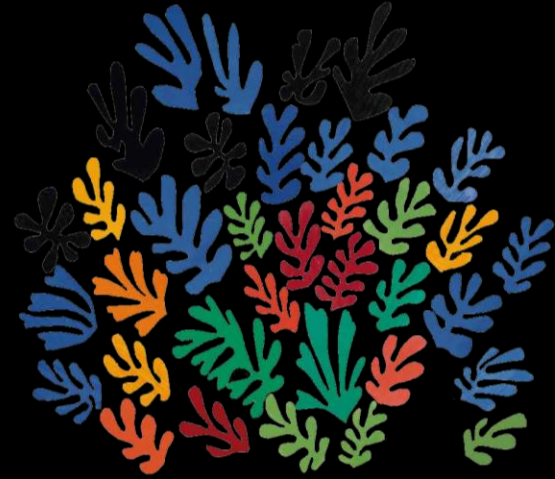
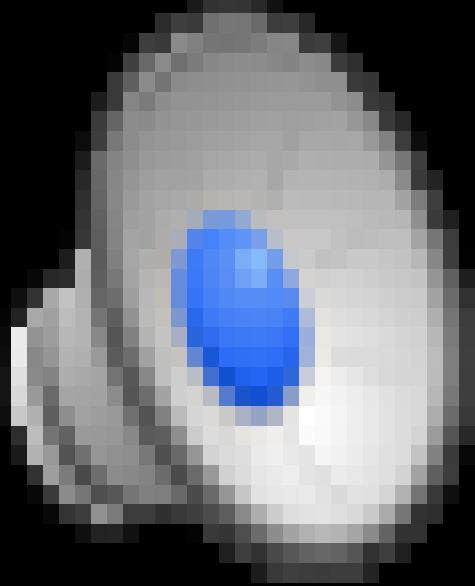


Un chronologie du suivi de partition

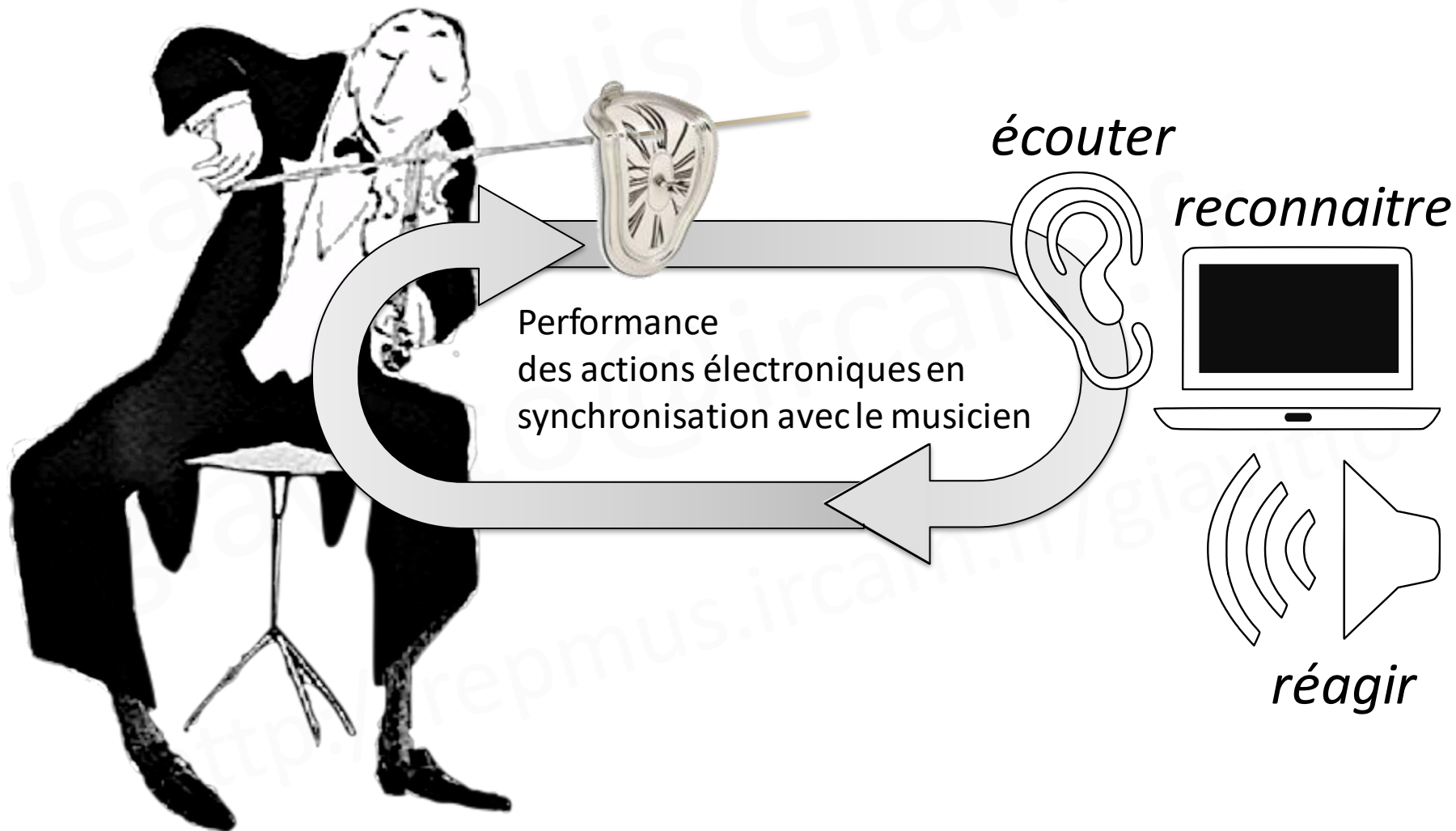


Un chronologie de Antescofo

- 2008 : thèse Arshia Cont (IRCAM + San-Diego)
 - machine d'écoute + envois de message Max
- 2012 : équipe INRIA
 - premier parseur du langage
 - début des thèses de José Echeveste et Philippe Cuvillier, équipe INRIA
- 2013 : action structurée, stratégie loose et tight
- 2014 : équipe-projet INRIA, projet ANR INEDIT
 - 1 IRCAM (50%), 1 CNRS, 1 INRIA
 - NIM, pattern temporel
- 2015 : thèse José Echeveste (Sorbonne Université), financement QWANT
 - les stratégies de synchronisation @tight et @target
- 2016 : thèse de Philippe Cuvillier (Sorbonne Université)
 - 2^{ème} version de la machine d'écoute intégrant la formalisation probabiliste du tempo
- 2017 : création de la start-up
 - application *Metronaut*
 - l'objet antescofo~ reste utilisable librement
 - code partagée entre *antescofocore* et *Metronaut*
- 2018 : deuxième vague du développement du langage
 - solveur EDO + zero-crossing, extensions de la librairie, prototype intégrant Faust, etc.



Musique mixte retrouver la performabilité



Une bonne idée dans le suivi de partition (*Arshia Cont*, 2008)

suivi de partition

machine d'écoute

un langage de
programmation temps-réel

alignement
audio/partition

détection de
hauteur +
estimation tempo

programmation
synchrone
(événement)

système
temporisé
(durée)

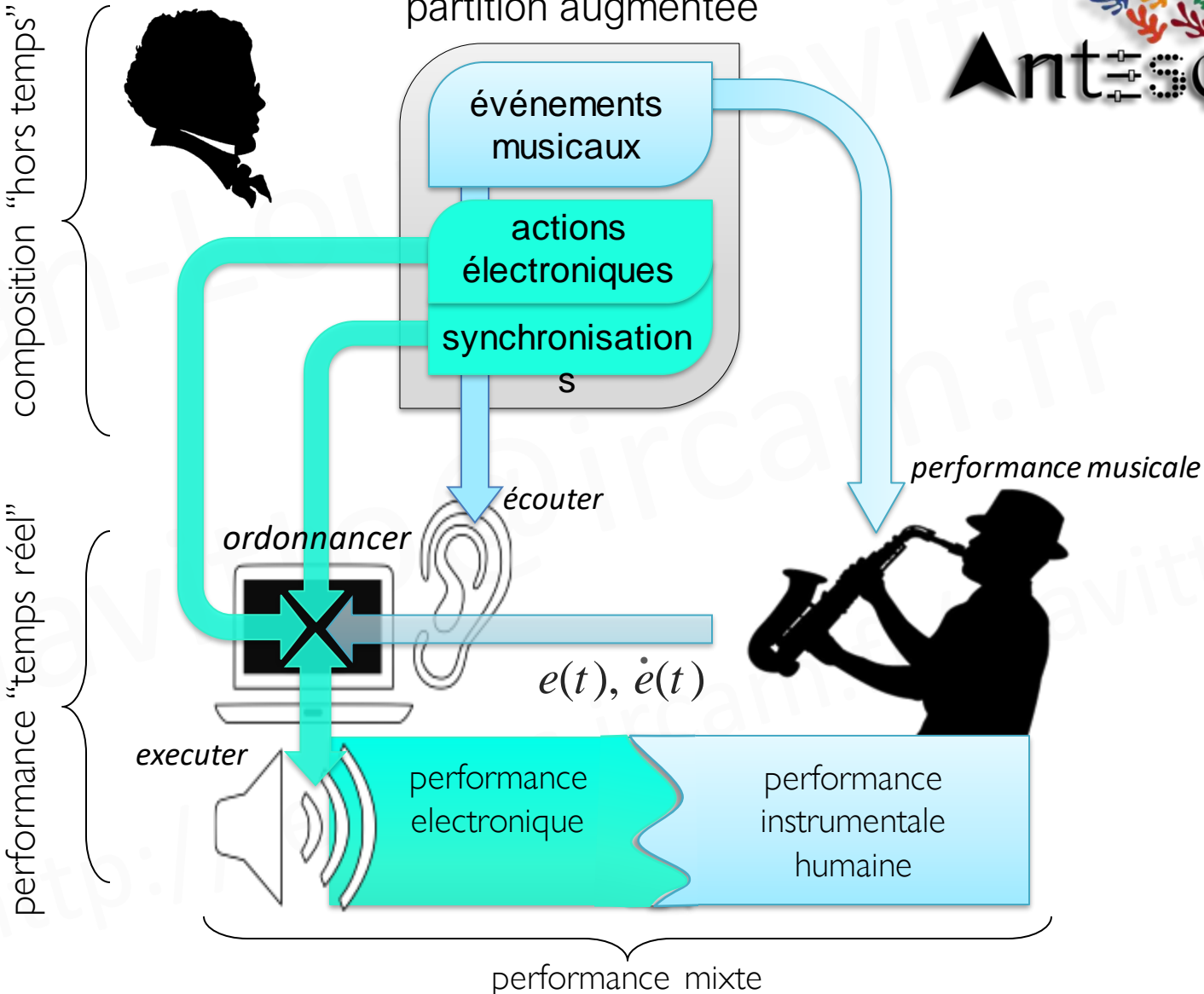
Arshia Cont, Philippe Cuvillier & Jean-Louis Giavitto, José Echeveste



L'approche Antescofo pour la musique mixte

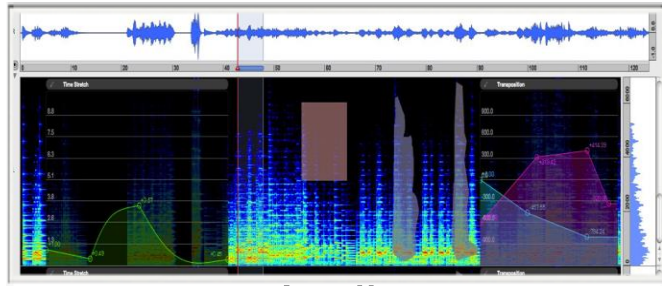


Antescofo

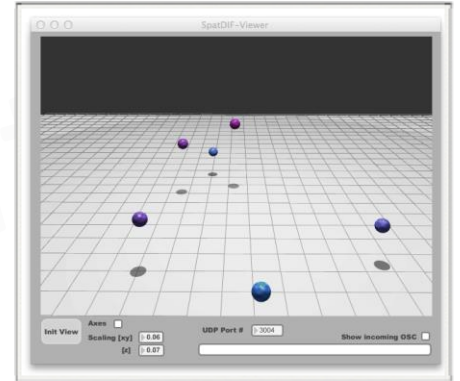




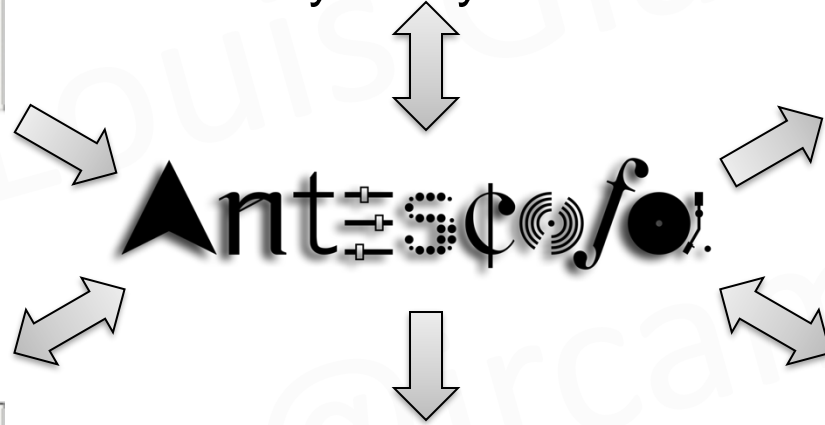
Gestur
e



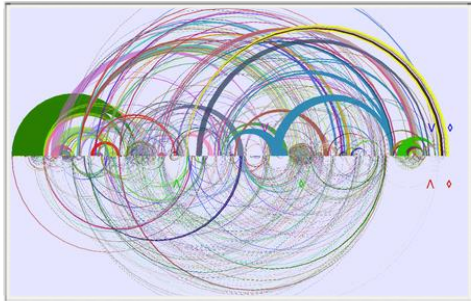
Audio
Analysis/Synthesis



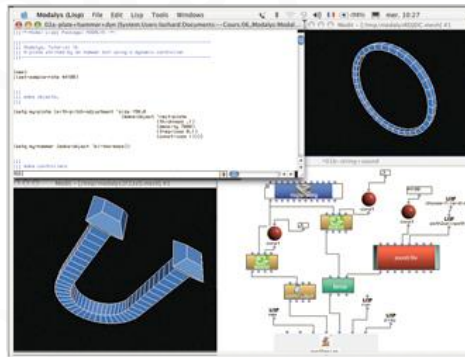
Spatialization



Antescofo



Improvisatio
n



Physical Models



MIDI /
Control

More than 100 creations

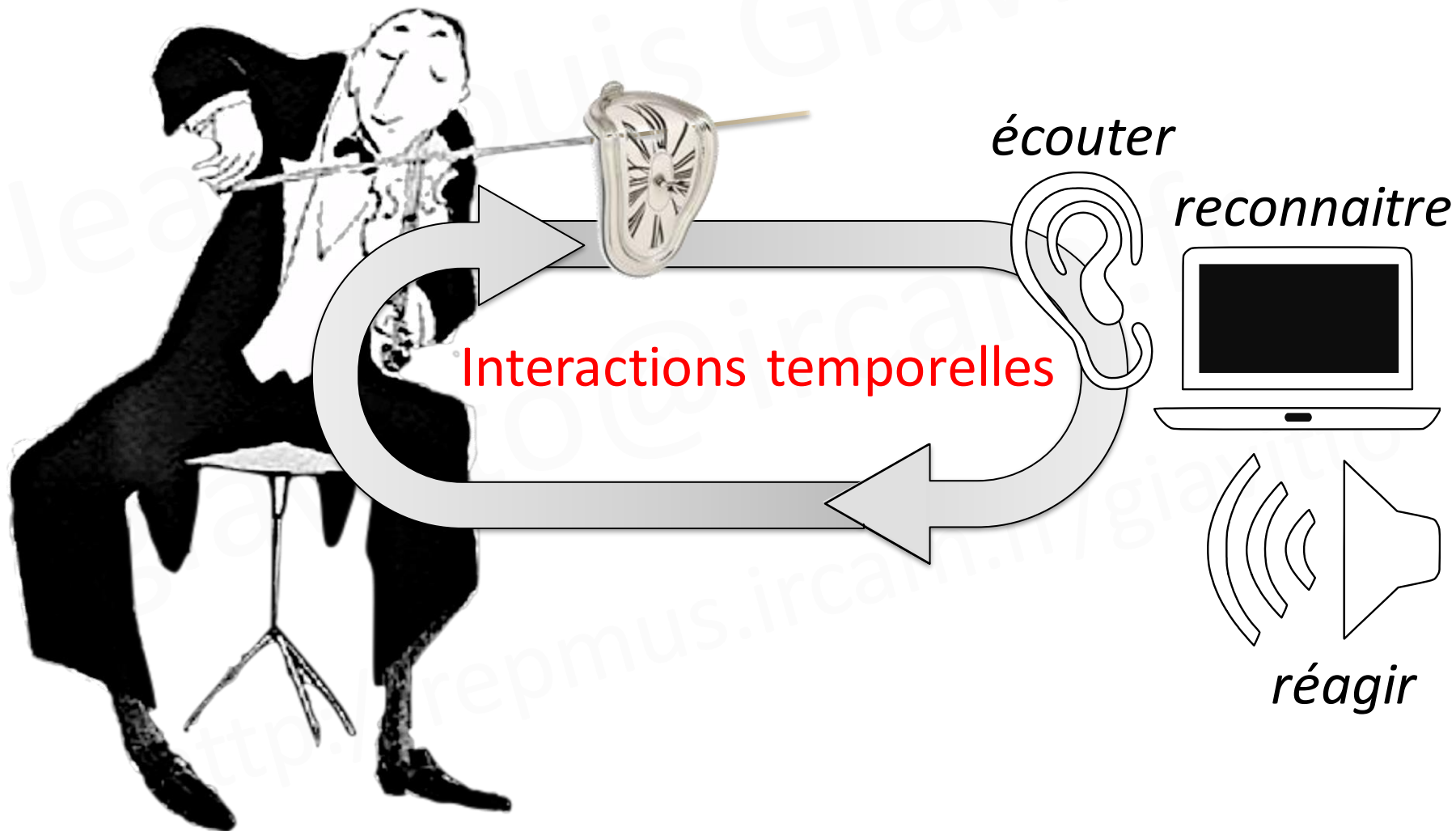
New York Philharmonics, Chicago Symphony, Los Angeles Philharmonics, Berlin Philharmonics, BBC Orchestra...

Tesla ou l'effet d'étrangeté Sasha Blondeau (2014)

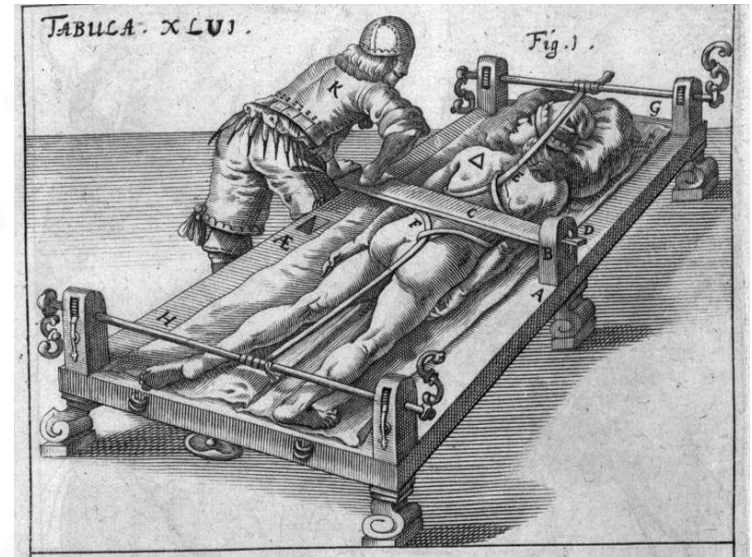
alto: Christophe Desjardins, real-time electronic: Antescofo

Musique mixte

retrouver la performabilité



le temps des physiciens,
le temps des informaticiens



le temps des musiciens ?

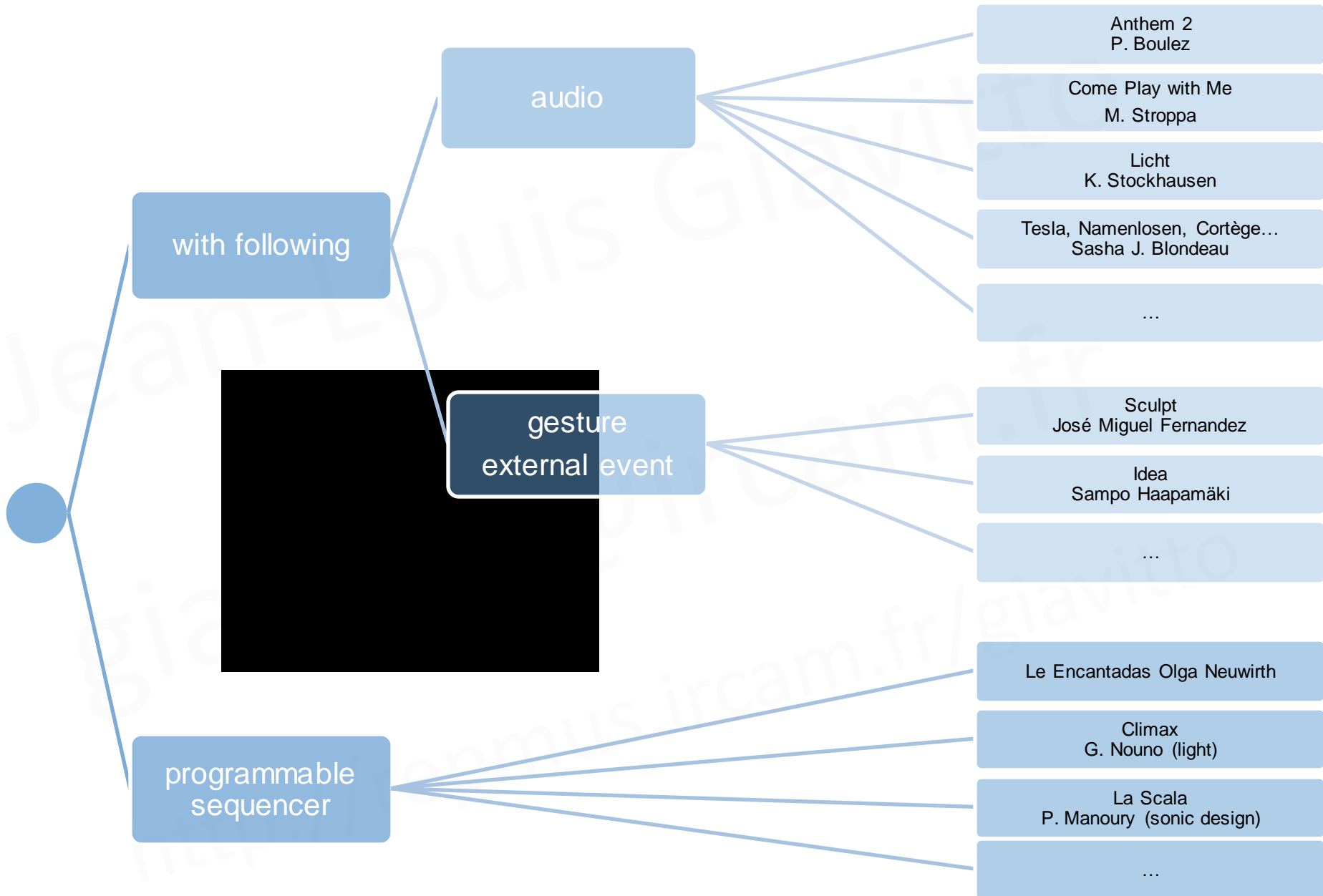
*Tempus ex Machina: A composer's reflections on musical time*¹

Gérard Grisey

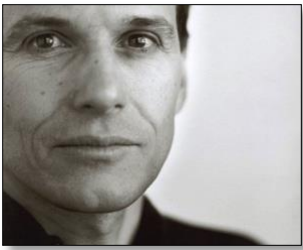
Dept. of Music, University of California, Berkeley, California, U.S.A. (composer)

This article is divided into three sections (the skeleton of time, the flesh of time, and the skin of time) and clearly distinguishes conceptual (or chronometric) time from perceptual (or psychological) time. The author firstly liquidates a few theoretical avatars such as rhythmic symmetry and then proposes a scale of complexity for duration that goes from

A growing repertoire



QUELQUES EXEMPLES



Marco Stroppa

Antescofo library for Totem control

```
; SOURCE Object Definition

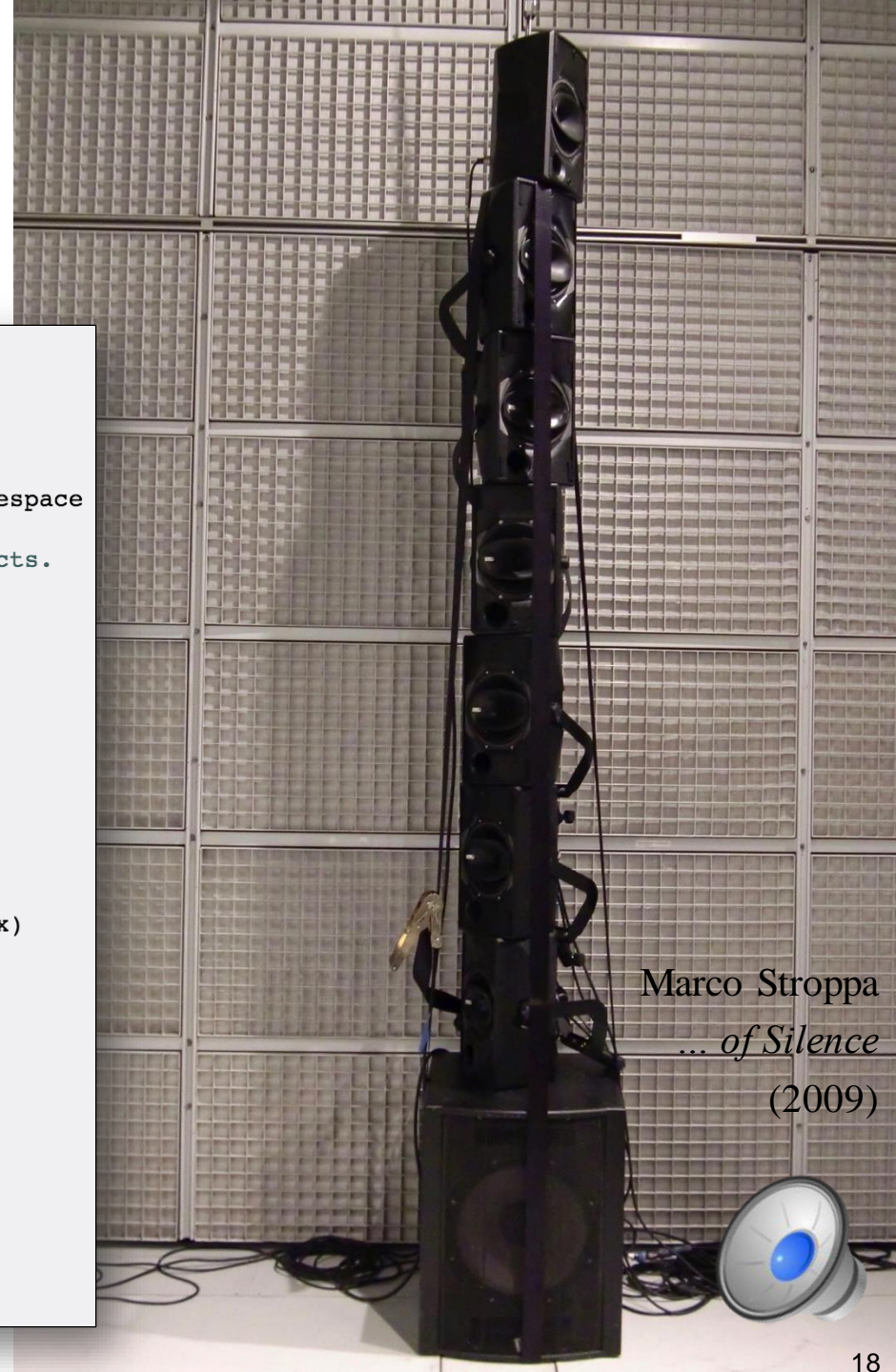
@obj_def source($idn, $npoints, $namespace)
{
  ; local state
  @local $coordinates, $idnum, $last_action, $prefix_namespace

  ; A broadcasted signal to all instances of source objects.
  @broadcast reset() { abort $last_action }

  // ...

  ; igoto method will create a curve that goes
  ; from $coordinates to $destination in $dur time
  @proc_def igoto($destination, $dur, $itp)
  {
    curve FlyingEngine @Grain := 0.05s,
      @Action := {
        $coordinates := $x
        @command($prefix_namespace+$idnum+"-spat") ($x)
      }
  }
  {
    $x
    {
      { ($coordinates) } @type $itp
      $dur { ($initlevels($destination)) }
    }
  }
}

// ...
}
```



Marco Stroppa
... of Silence
(2009)

Une approche algébrique

```
@obj_def source($idn, $npoints, $namespace)
{
  ; Create local variables that are specific to each instance of the object
  @local $coordinates, $idnum, ...

  ;; A broadcasted signal to all instances of source objects. Useful in panics!
  @broadcast reset() {...}

  @abort {...}

  @fun_def space_jump($destination) {...}

  @proc_def goto($destination, $dur) {...}

  @proc_def igoto($destination, $dur, $itp) {...}

  @proc_def goto_rt($destination, $dur)
  {
    | _ := $THISOBJ.igoto_rt( $destination, $dur, "sine_in_out" )
  }

  ;-----
  ; Random Granular Spatial Movement Process

  @fun_def make_unique_random_config($configs) {...}
  @proc_def xRandomSpace_i($configs, $grainrange_min, $grainrange_max, $stay_percent, $itp)
}
}
```

- définir une source
 - d'où vient le son
 - position (diffusion) dans le totem
- sauter à un endroit
- aller (continument) vers un autre endroit
- mouvement
 - position de départ
 - position d'arrivée
 - durée
 - dynamique

```
bouge_ici(source1) || delay(5) (bouge_la_bas(source2))
```

```
(faire_en_parallelèle
  ((bouge (ici)) source1)
  ((delay (beat 5)) ((bouge (la_bas)) source2))
)
```



704 EF_Process.asco.txt (3.72%)

82 EF_init.txt

1469 EF_Original_Score.tx

7376 EF_T5_Score.txt

11497 EF_T7_Score.txt

21128 total



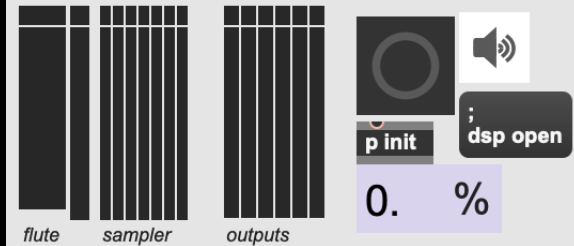
Explosante-Fixe Pierre Boulez

Computer Music Design: Andrew Gerzso

Max7 port & Antescofo integration :
Augustin Muller - 2017

with elements from Gilbert Nouno & Carlo Laurenzi

augustin.muller@ircam.fr



p DSP_new

p Transport&Score

p CTL_CONFIG

p midifaders

p readme

p CONFIG_MAX_PREF

Reset

test tone

loadbang

ReLoadBang

close

```

; APPEL DES CHACHA (accords répétés avec accélération/décélération)

@proc_def ::ChaCha($pitch_list, $period_list, $accvel, $noaccvel, $chan, $dur)
;liste des notes, arguments pour periodvar ([start, dest, temps]), veloaccents, elofond, proba des accents, chan, dur
{
  @local $accproba := 1/$pitch_list.size(), $P := []

  if ($ChaCha) ; update
  {
    forall $p in $pitch_list
    {
      @local $i := $pitch_list.occurs($p)
      if (0 <= $i && $i < $ChaCha.$P.size())
      {
        _:=( $ChaCha.$P[$i]).EFL_Pitch([($pitch_list[$i]))
      }
      else
      { ; ajout de voix FAIRE LA SUPPRESSION
        $q := obj::EFLoop("seq", [$p], [1], $period_list[0], $chan, $dur, $accvel, $noaccvel, $accproba)
        $ChaCha.$P:= @push_back($ChaCha.$P, $q) ;
      }
    }
    _ := @EFL_Accents(::ChaCha, $accvel, $noaccvel, $accproba)
  }
  else ; appel du proc global
  {
    $ChaCha := $MYSELF
    $P := [ obj::EFLoop("seq", [$p], [1], $period_list[0], $chan, $dur, $accvel, $noaccvel, $accproba) | $p in $pitch_list ]
  }

  _ := ::PeriodVar(::ChaCha, $period_list) ; courbe partagée pour tous les chacha
}

```

```

obj_def EFLoop($mode, $pitch_in, $cycle, $base_period, $chan, $dur, $accvel, $noaccvel, $accproba)
{
  @local $current_period, $current_pitch, $rand, $vel, $next_period, $next_pitch, $body, $curve, $x, $go, $AccentPitch, $IsAccent, $count
  @init {
    $body := 0
    $count := 0
    switch ($mode) ; on regarde le mode, et on choisit la fonction next_pitch en conséquence; possible de choisir la fonction depuis l'extérieur
    {
      case "seq" : ; Séquenceur : Cas de Chug, ChaCha, Repeat, Seq ... on répète un accord avec un rythme donné (séquence rythmique et tempo)
        $current_pitch := $pitch_in ; déjà un tableau
        $next_pitch := @pitch_chords()
        $current_period := @increment_period_cycle(0, $THISOBJ);
        $next_period := @increment_period_cycle($count) ; currification
      case "ranrep" : ; cas des ranreps : séquenceur aléatoire, on tire au sort des notes et des périodes (mode arpeggio quoi)
        $current_pitch := @random_pitch_from_tab($THISOBJ)
        $next_pitch := @random_pitch_from_tab()
        $current_period := @random_period_from_tab($THISOBJ)
        $next_period := @random_period_from_tab()
      case "slant" : ; cas des slant ; même gestion rythmique par contre on fait des gammes
        $current_pitch := [$pitch_in[0]]
        $next_pitch := @transpose_pitch()
        $current_period := @increment_period_cycle(0, $THISOBJ)
        $next_period := @increment_period_cycle($count)
      case "wobble" : ; cas des wobble ; on joue des accords par contre la gestion des rymes est différente (variation autour d'une période de base)
        $current_pitch := $pitch_in ; déjà un tableau
        $next_pitch := @pitch_chords()
        $current_period := @compute_period_wobble($THISOBJ)
        $next_period := @compute_period_wobble()
    }
  }

  @broadcast EFL_Sync($ancestor) ; restart de tous les EFLoop
  @broadcast EFL_Period($ancestor, $ch_period) ; variation globale de la période
  @broadcast EFL_Accents($ancestor, $ch_accvel, $ch_noaccvel, $ch_accodds) ; modification des vélocités (accents, bckgnd, proba)
  @method_def EFL_Rythm($ch_cycle) ; changements de rythme par instance pour chugs
  @method_def EFL_Pitch($ch_pitch) ; changements de pitch par instance

  @whenever ($go) ; Corps de l'objet, la boucle a proprement parler. les changements de période interviennent à la loop suivante
  {
    $body := 0
    $count := 0
    abort $body
    $body := {
      loop ef_loop ($current_period) ms
      {
        if (@rand(1) <= $accproba) { $vel := $accvel}
        else { $vel := $noaccvel}
        if ((0 < $vel) && (0 < $current_pitch))
        {
          ::MakeChord($current_pitch, $vel, $chan, $dur)
          $current_period := $next_period($THISOBJ)
          $current_pitch := $next_pitch($THISOBJ)
          $EFLSync := $THISOBJ
          $count := ($count+1)
        }
      }
    }
  }

  @whenever ($current_pitch == [0]) ; le pitch 0 arrête l'instance
  @whenever ($vel == $accvel) ; contrôle des accents pour le pitch courant

  @init { $go := true } ; il faut le mettre après ;
}

```



Augustin Muller – IRCAM

Christopher Trapani

real-time rhythmic canon à la Nancarrow



start

Clarinet in B \flat

Cl.

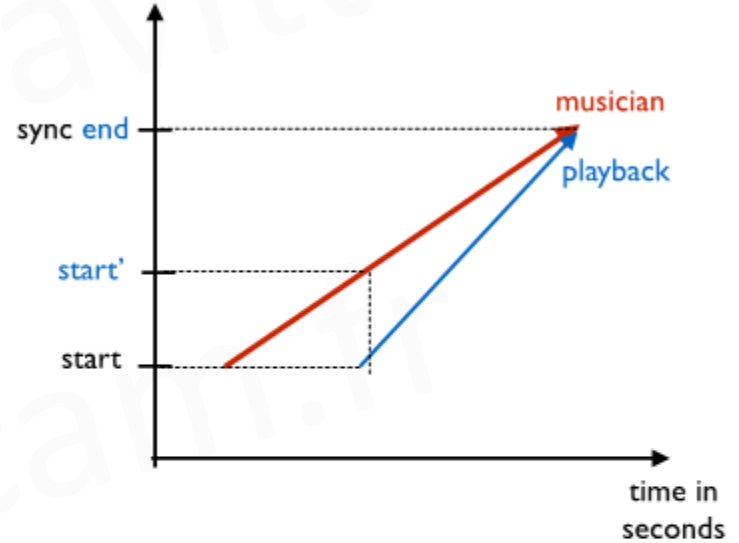
start' [playback]

Cl.

Cl.

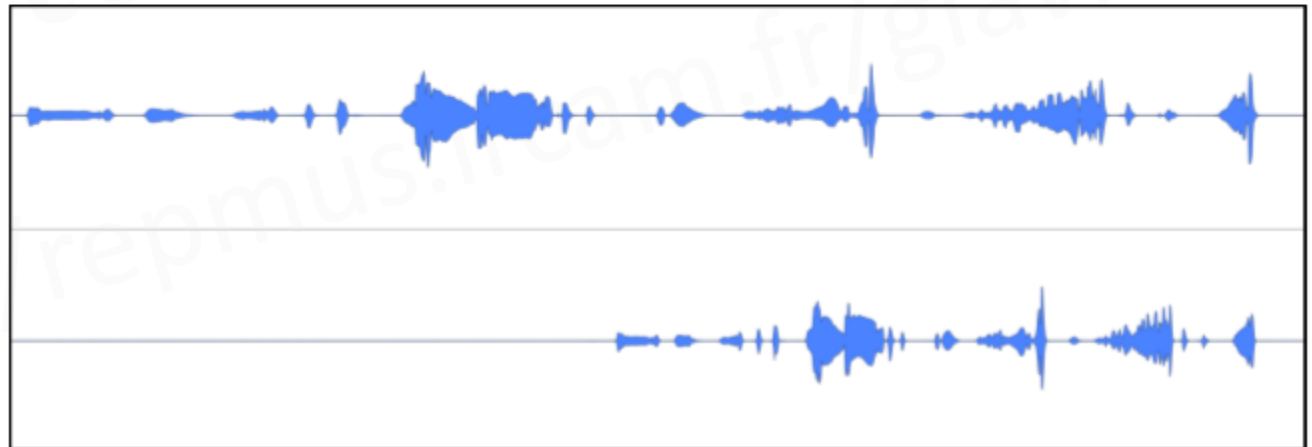
sync end

position in beats
in the score



musician

playback



Christopher Trapani

real-time rhythmic canon à la Nancarrow

SCORE in C

Sketch — ~~7 Nov 2012~~ 3 DEC

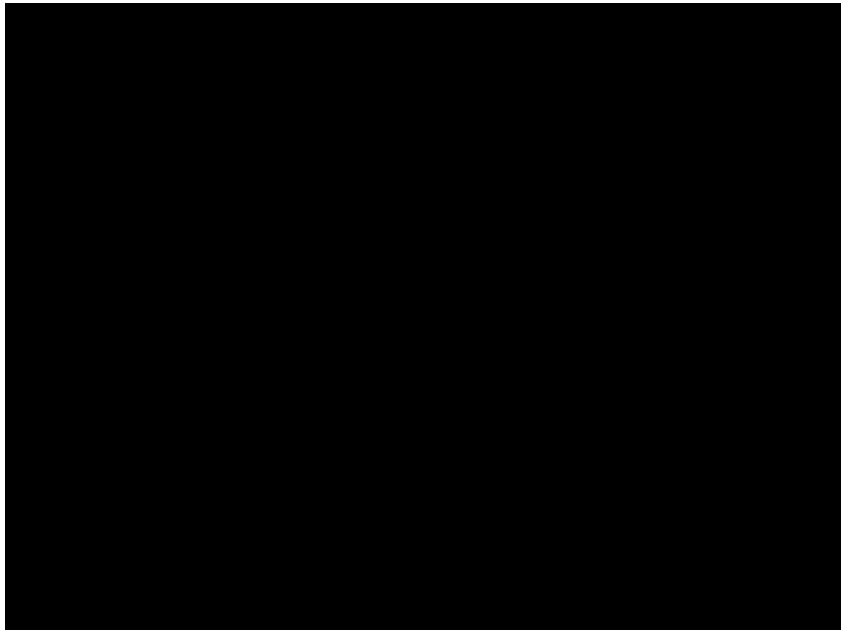
CHRISTOPHER TRAPANI

markers to be placed by Antescofo

⊗ possible convergence points

É

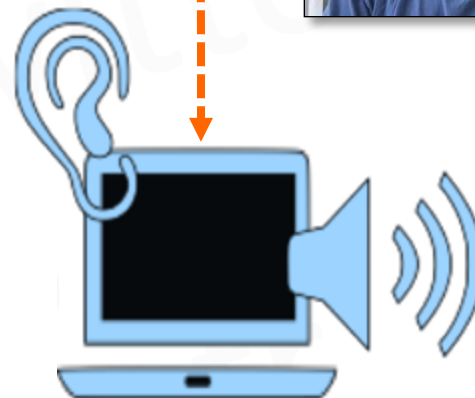
The musical score is for Clarinet in Bb, written in 3/4 time with a tempo of ca. q=108. It consists of seven systems of music, each starting with a measure number and a convergence point marker (A through G). The score includes various musical notations such as dynamics (f, mp, p, pp, sfz, mf, n), articulation (accents, slurs, gliss., pizz.), and performance instructions (sub., tong., slap). Convergence points are marked with boxes and letters: 1 (red), 2 (green), 3 (blue), 4 (purple), 5 (yellow), 6 (pink), and 7 (grey). Arrows and lines connect these points to specific measures, indicating the 'point d'entrée défini' for each section. For example, 1B -> A is marked in red at the top right, and 2 -> B is marked in purple in the middle. The score also includes time signatures like 2/4 and 3/4, and various rhythmic values like eighth and sixteenth notes.



José-Miguel Fernandez *gesture-driven synthesis*



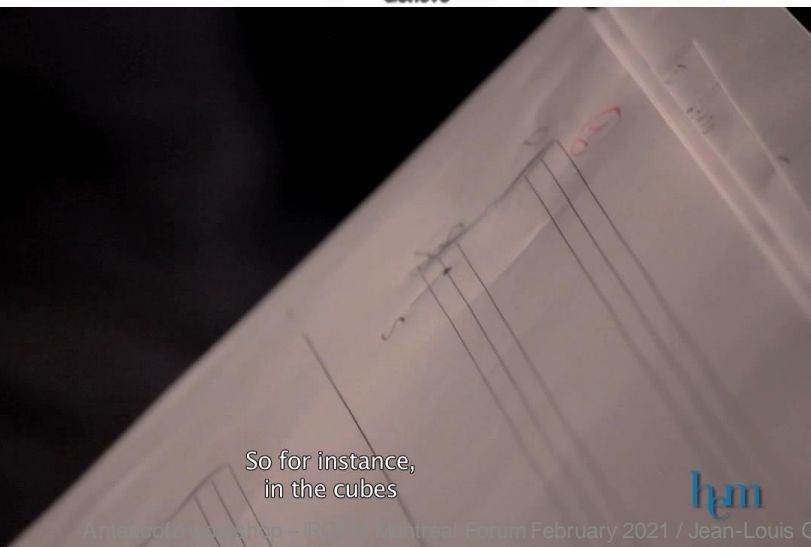
OSC or setvar



GeKiPe (Gest Kinect Percussion)
Philippe Spiesser (percu)
Alexander Vert (composition)
Jose Miguel Fernandez (RIM)



Hypersphère Jose Miguel Fernandez
séance de travail IRCAM 26/2/16



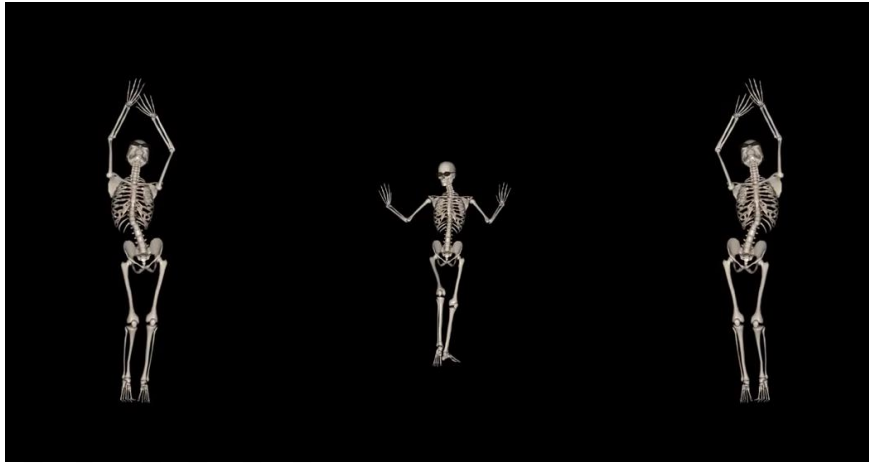
So for instance,
in the cubes



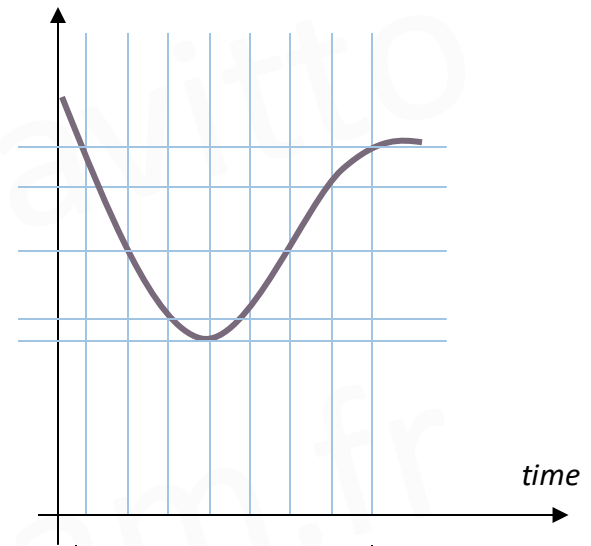
Suivi du chef d'orchestre (Jean Deroyer)
synchronisation de l'électronique (ISMM, RIM : Serge Lemouton)
IDEA, Sampo Haapamäki

5'02, 7'14

Using a gesture follower

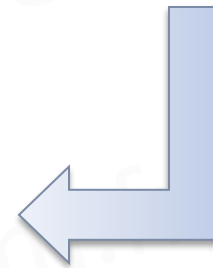
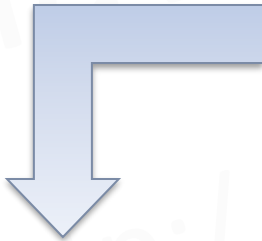


data →



```
@tempovar $t(60, 1)
```

```
loop 0.1 @tempo $t  
{ send X $t.tempo }
```

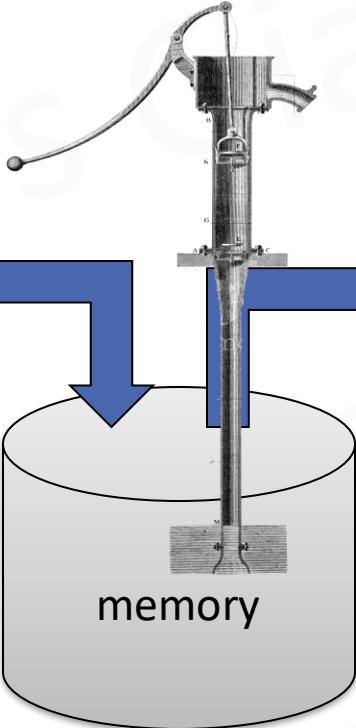


Recorder and player



record

a change
in time



play

a change
in time



*process
in time*

data in memory

*process
in time*

Souvenirs du futur 2023 — ...

- développement d'une version open-source modulaire du langage
 - la machine d'écoute sera un plugin
 - pluginification de la machine d'écoute
 - pluginification de Gesture Follower (GF) : « écoute » du geste
 - autre machine d'écoute
 - scriptabilité du langage : message Max \equiv message OSC \equiv API
- Ponts : AntesCollider, OpenStageControl, ...
- Intégration à **Faust**
 - faire du son dans Antescofo
 - couche *architecture* de Faust pour le portage sur les arch. audio
- Compilation
 - des expressions instantanées
 - vers la génération de code temporisé
- Extension des patrons temporels (contraintes)

QUELQUES DICHOTOMIES

Quelques dichotomies structurantes...

- symbolique vs. analogique *ou signal/symbole*
- composition vs. performance
- hors-temps vs. temps-réel
- partition vs. instruments
- synthèse vs. échantillon
- analyse vs. synthèse
- modélisation physique vs. phénoménologique

Quelques dichotomies structurantes... dont il faut se méfier

- symbolique vs. analogique *ou signal/symbole*
- composition vs. performance
 - improvisation
- hors-temps vs. temps-réel
 - antescofo, OM#
- partition vs. instruments
 - MusicN
- synthèse vs. échantillon
 - Catarth
- analyse vs. synthèse
 - inpainting
- modélisation physique vs. phénoménologique
 - ... informé par la physique

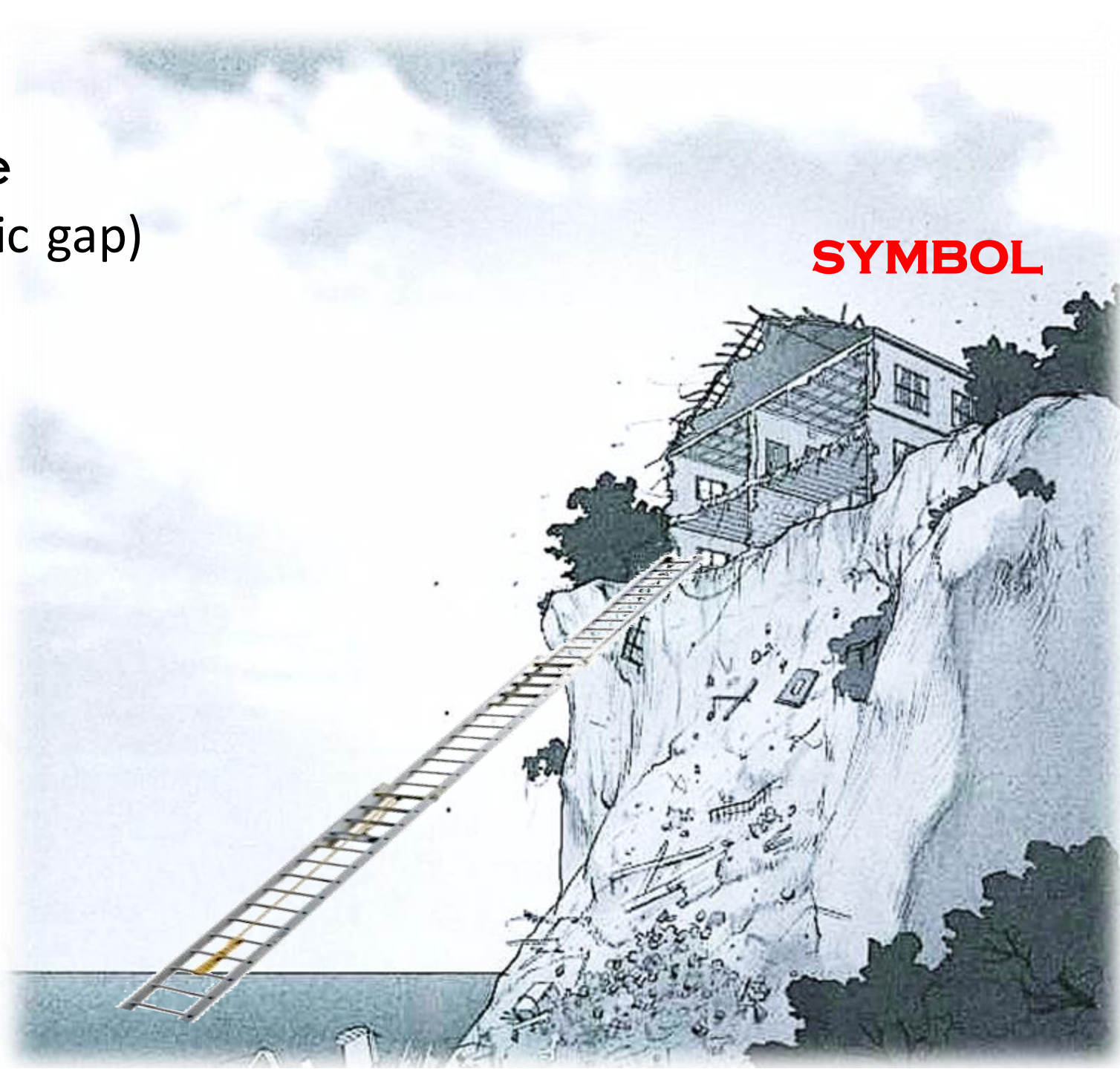
Le fossé sémantique (the semantic gap)

SYMBOL

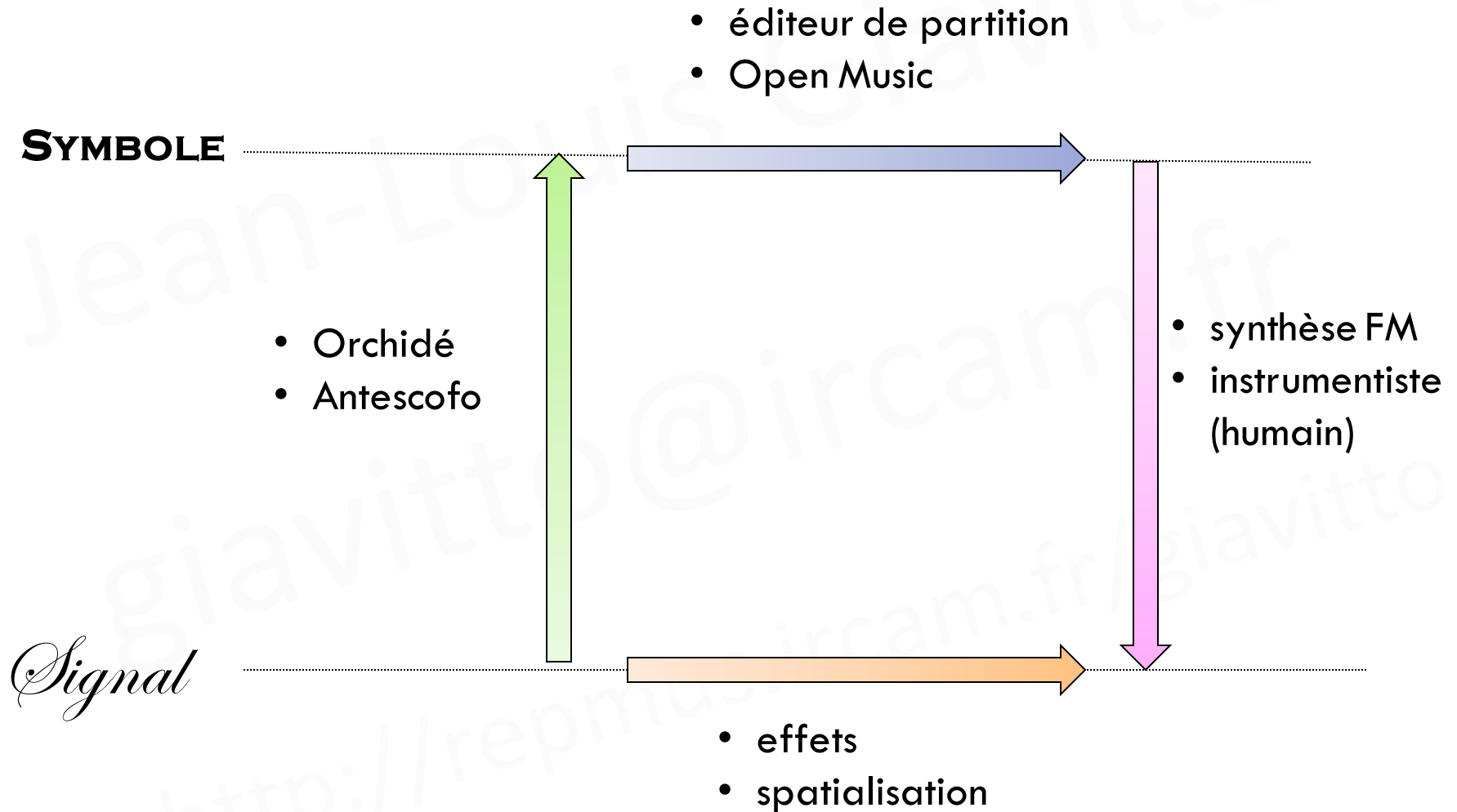
Jean

Signal

Signal

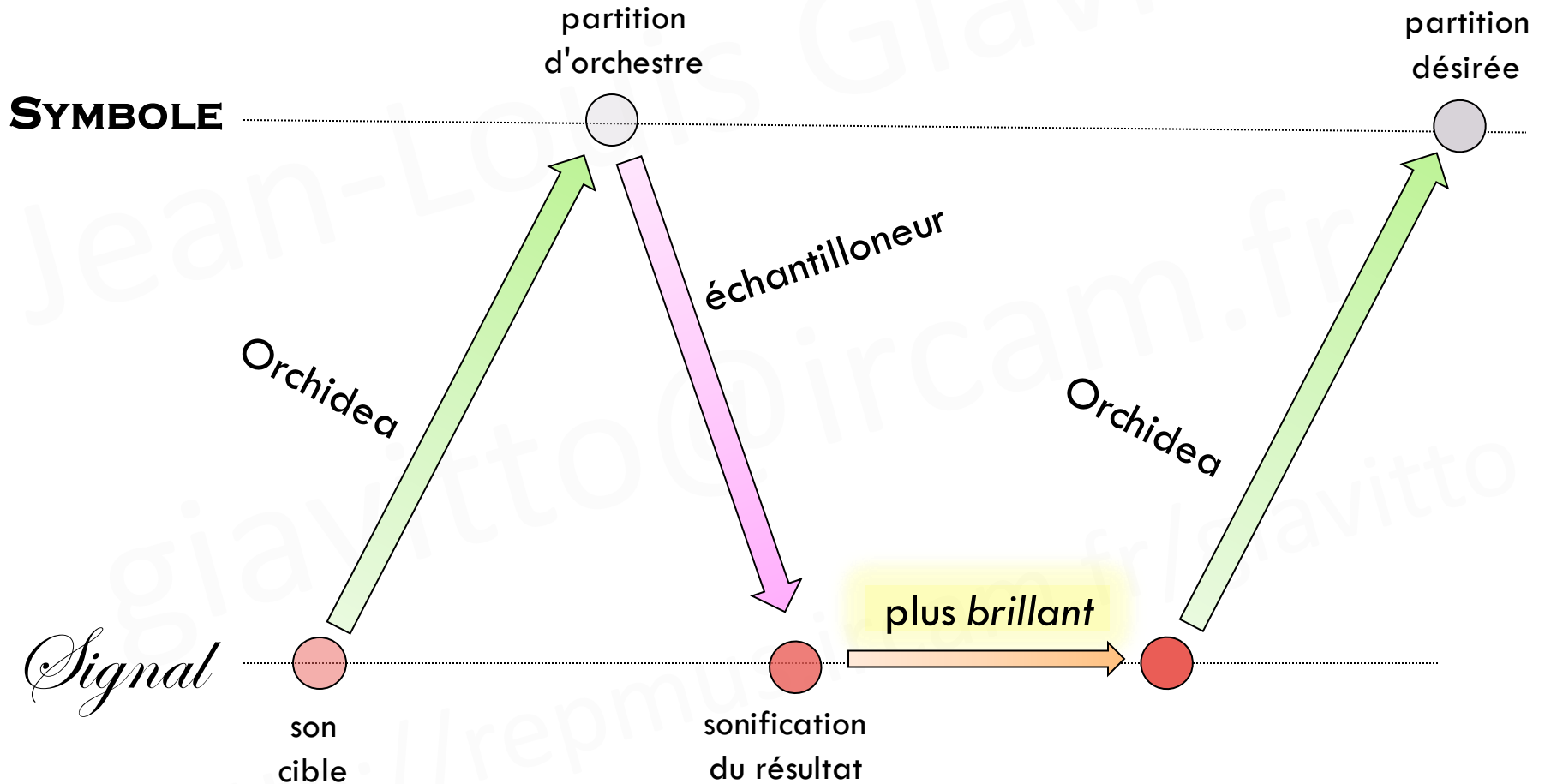


La dualité signal/symbole



La dualité signal/symbole

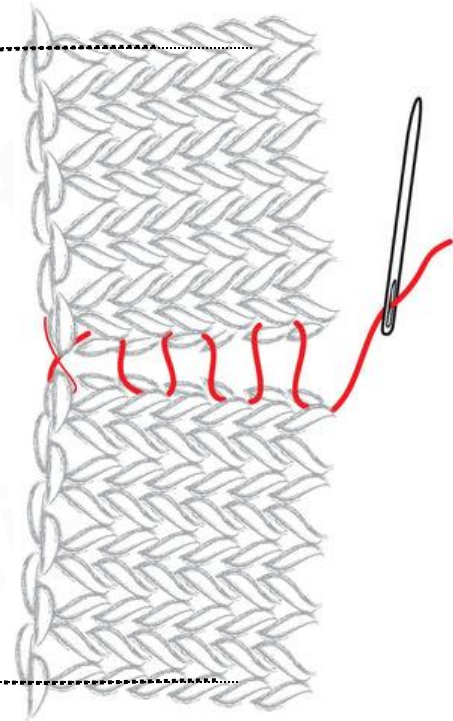
un exemple en **Orchidea** (Carmine Cella + Daniele Ghisi)



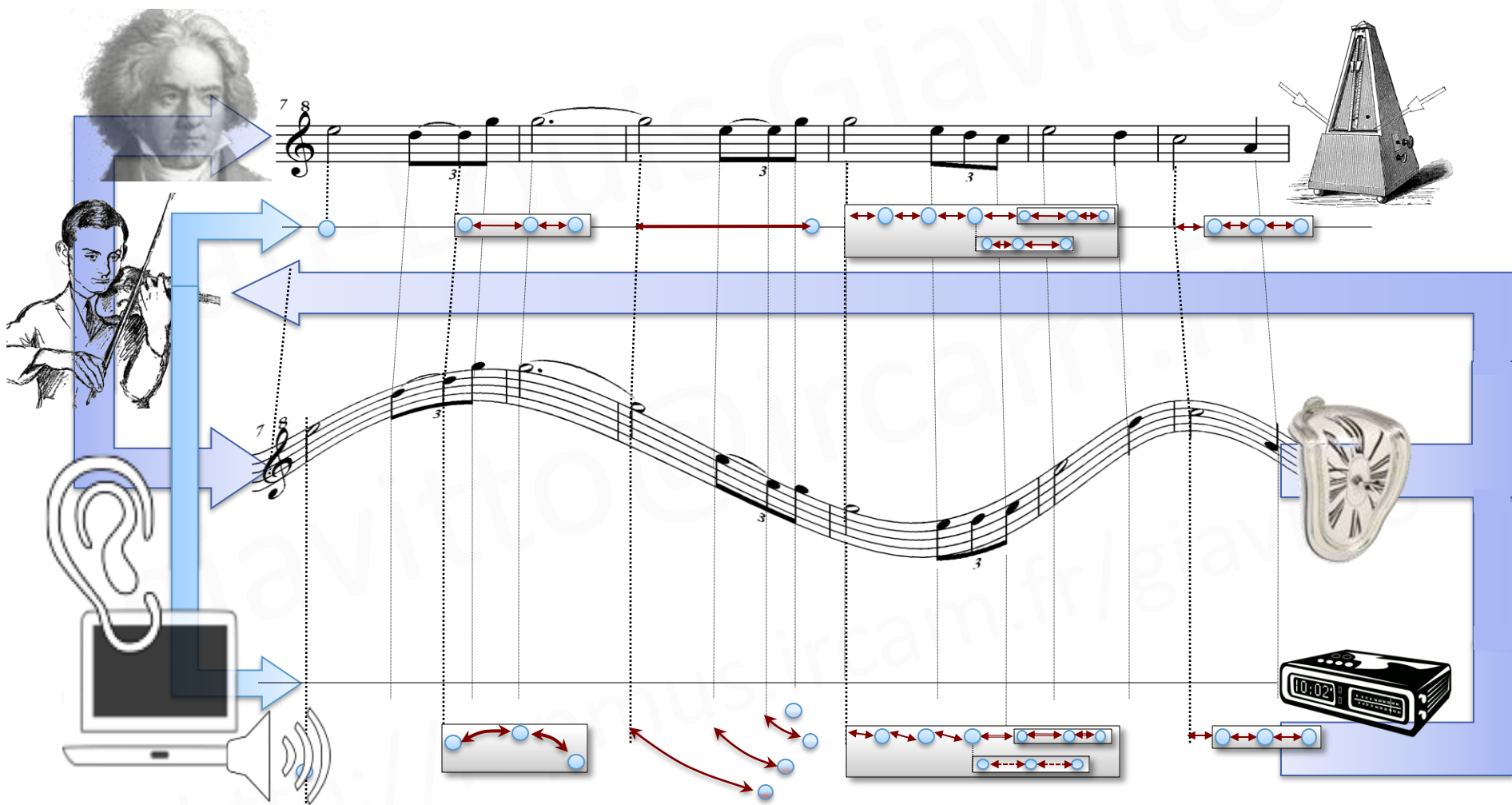
La dualité signal/symbole... à intriquer tricoter

SYMBOLE

Signal



Les multiples temps d'une partition (mixte) sont intriqués



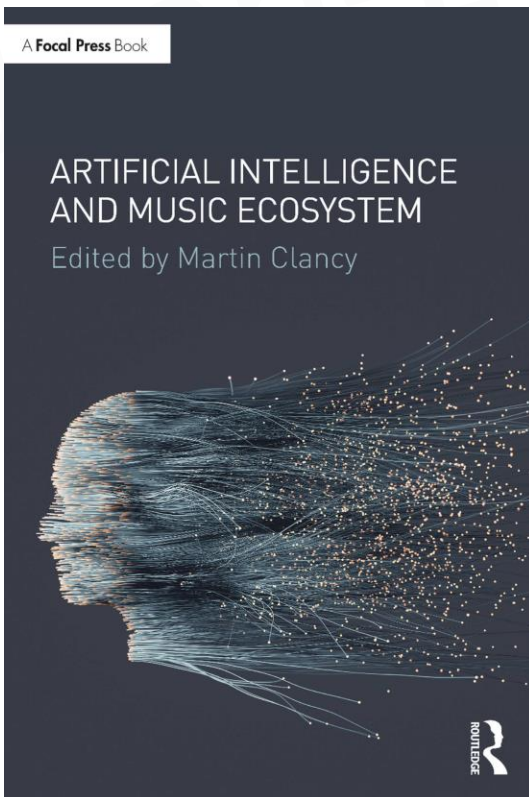
ÉCRITURES DE L'ÉLECTRONIQUE ?



The Pure Data Environment

Shortly after leaving IRCAM to join the University of California–San Diego (UCSD), I began working on and distributing a new software environment called Pure Data. I could easily have continued to contribute to Max and/or Max/MSP, but I had ideas that did not fit easily into that paradigm, and seeing that personal computers were now fully audio and video capable, I felt that a fresh start was needed. Among the many people I worked with on Pure Data were the composers Rand Steiger, Andrew May, Michael Theodore, and Kerry Hagan, performers Elizabeth McNutt and Juliana Snapper, visual artist Vibeke Sorensen, and fellow researcher/designer Mark Danks. My primary purposes in writing Pure Data were to first expand the available media from audio and MIDI to video, computer graphics, and network communications and, secondarily, to rethink the role of musical score and real-time process in music software. In the first aim, my work finds parallels in the later development of Max/MSP and Jitter. It now seems clear (if perhaps for reasons that are not yet understood) that software written to address real-time problems in music performance can also be well suited to all sorts of other real-time computing problems as well, notably live video.

My second purpose is more abstract and still largely unachieved. Although at the outset I aimed to make an environment for composers to use, I ended up making what amounts to a musical instrument: something that you tinker with during production but which is played in a concert. While there are ways to sequence predetermined actions in Max and Pure Data, the true power of both environments lies in their reactivity and the flexibility with which that reactivity can be specified. What I did not do was write an editor for entering musical scores. To do so in a way the composers around me would use, I would have needed to build Western musical culture into the software, in the form of staves, clefs, quarter notes, pitch names, etc. This process struck me as privileging old-fashioned ideas about musical composition, which would not only be a snub to non-Western musical practices but would also hamstring even strictly Eurocentric composers. After all, the whole purpose of electronic music was to emancipate ourselves from the limitations of traditional instruments, and in that respect, making a musical score editor would be a giant step backwards. Besides, I could not read music, so I would have had a colossal task in front of me.



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)
...	...

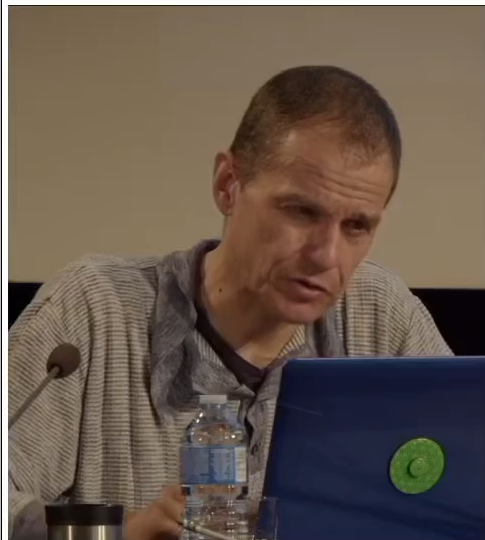
- **Partition \neq Programme**

- G. Mazzola : un « geste surgelé »
- « dégeler n'est pas exécuter »

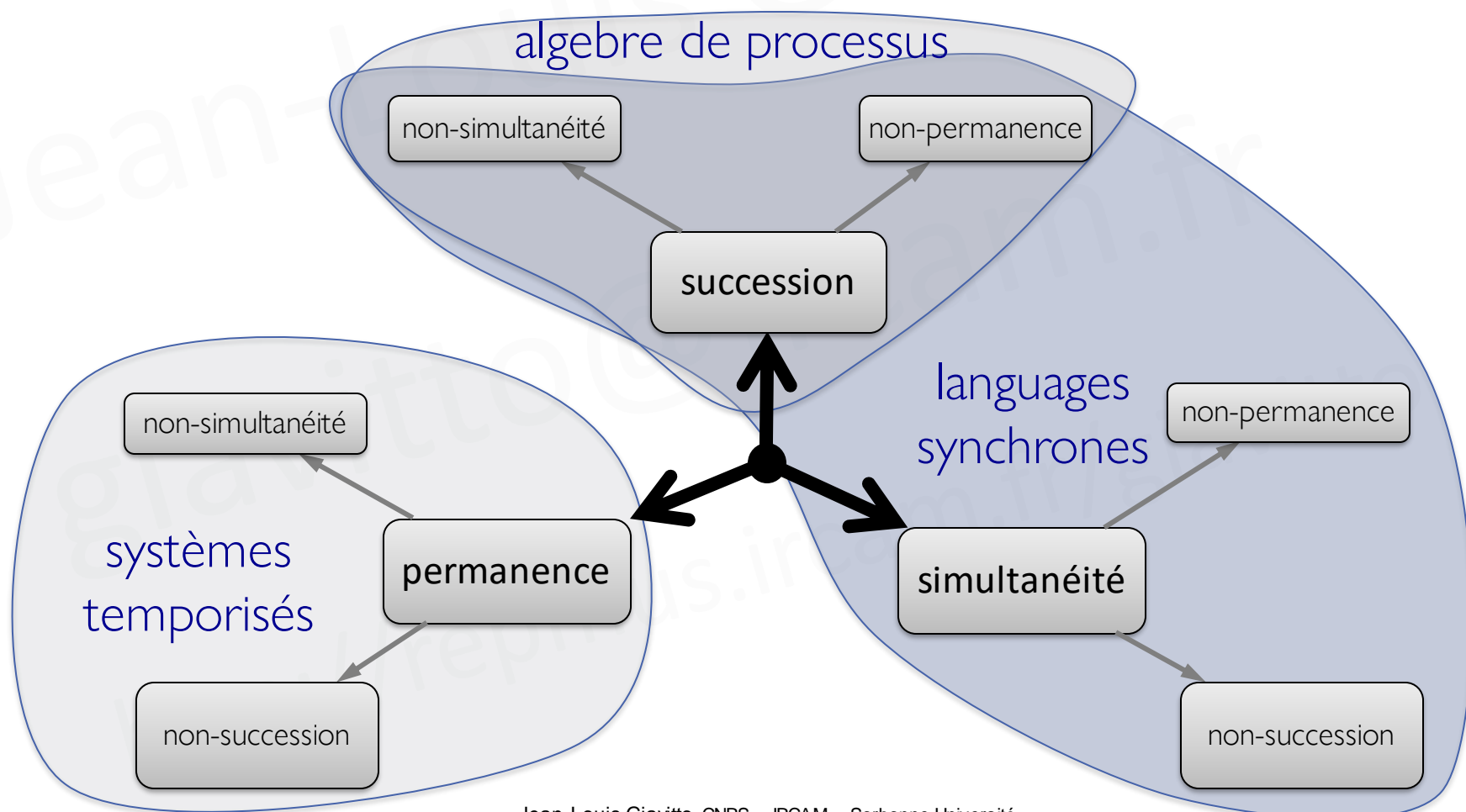
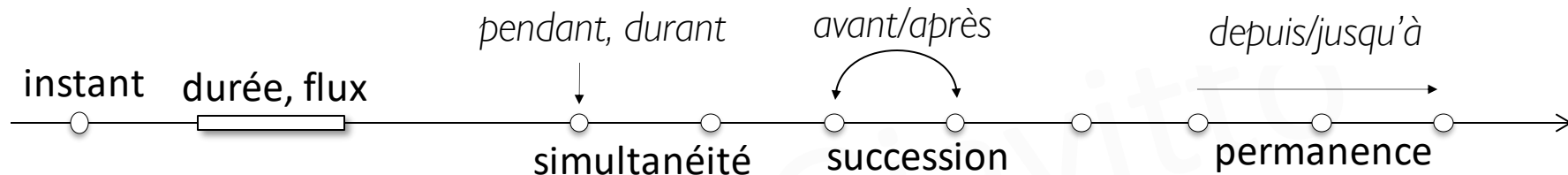
Marco Stroppa, 9 mai 2023, intervention dans le cadre du Coursus
A sense of synthesis (le sens de la synthèse):
around the compositional environment of Lance dei Crepuscoli

B. The Venice Time, 1980-83 (A. Vidolin)

- c. **Structured programming**
[Assembler/Fortran]
(procedural abstraction)
(data abstraction)



Instant et Durée : Simultanéité, Succession & Permanence



Pertinent aussi en musique !

- **simultanéité**
 - harmonie, polyphonie
- **succession**
 - mélodie
- **permanence, durée ?**
 - midi !
 - mais pas que !
(rubato, accelerando, deux fois plus vite, etc.)

Déclaratif, dénotationnel / procédural, opérationnel

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

The diagram illustrates a musical score for guitar. It features a single staff with a treble clef. The notation includes a series of notes on the first string (I. Saite) and a trill on the second string (II. Saite). The first string notes are marked with a *pp* dynamic and a *V* (vibrato) symbol. A trill on the second string is marked with *sim.* and *+ II. Saite*. The score concludes with an *arco stop* instruction. The diagram also includes a diagram of the guitar neck and a hand holding the neck, with labels for *Steg* (bridge) and *Griffbrett* (fretboard). A legend at the bottom explains the hand positions and the technique for the trill.

Hälsa aufwärts = rechte Hand
Hälsa abwärts = linke Hand

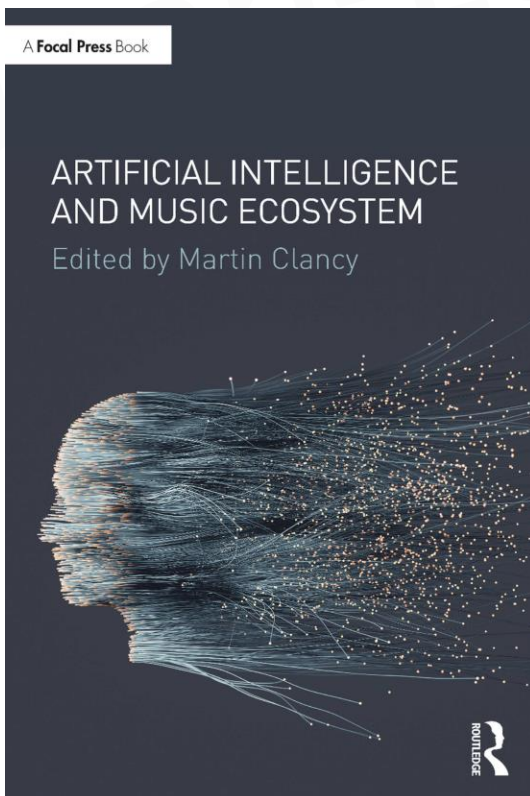
mit Fingerkuppe locker - quasi flageolett -
auf der Saite hin und her fahren.

The Blank Page as a Software Design Ideal

Every software developer has to make trade-offs between ease of use and neutrality. One way to make a program easy to use is to make one gesture do the work of many. The designer predicts which sequences of gestures are more likely than others and provides shortcuts for those sequences. Assuming you have all possible keystrokes bound to some action or another, each new shortcut introduced displaces whatever action was previously bound to the action. In choosing which shortcuts to provide, the software developer nudges the artist towards a particular set of choices.

An example would be Western pitch notation, in which 'C3' has become shorthand for 261.62 Hz. To introduce such a shorthand would be to nudge the artist towards the Western-tempered scale. This design decision privileges Western culture above non-Western ones and is somewhat problematic (although the issue is complicated because the computer itself is a product of the West). Nevertheless, even in the 1980s, when most composers at IRCAM would not have raised cultural hegemony as an issue, the very fact that they were using computers to make music implies a desire to specify things with greater freedom and precision than Western musical practice, for example, by limiting us to that twelve-tone tempered scale had accustomed them to. Although many software designers thought otherwise, my conclusion was that it would be wrong to provide Western pitch names as shorthand for the raw numbers.

To accomplish this, we need simultaneously to eliminate bias (which can be thought of as making our output set cover specific musicians' desired musical outputs at the expense of others) and to make the set as sparse as we can so that our fixed number of points 'covers' as significant a field of possibilities as possible – this is achieved by allowing small changes in input to make significant changes in output. Nevertheless, of course, we somehow have to avoid falling into the trap of introducing biases via the shortcuts mentioned above. Furthermore, here we arrive at what appears to me to be a paradox. To make the software expansive, in the sense that a slight input change, one sparing in its demand on the user, can result in a relatively significant output change sufficient that the set of all possible outputs can be as extensive as possible, we thus ought to automate whatever we can. That is, we ought to make it possible to specify large gestures using compactly contained instructions. However, how is this different from providing the convenient shortcuts – a tendency I have denounced a few paragraphs earlier? It is hard to imagine in this context a way in which the expressive power of an input language could not be a manifestation, at bottom, of the designer's preconceived and presumably heavily biased notions of what good music is.

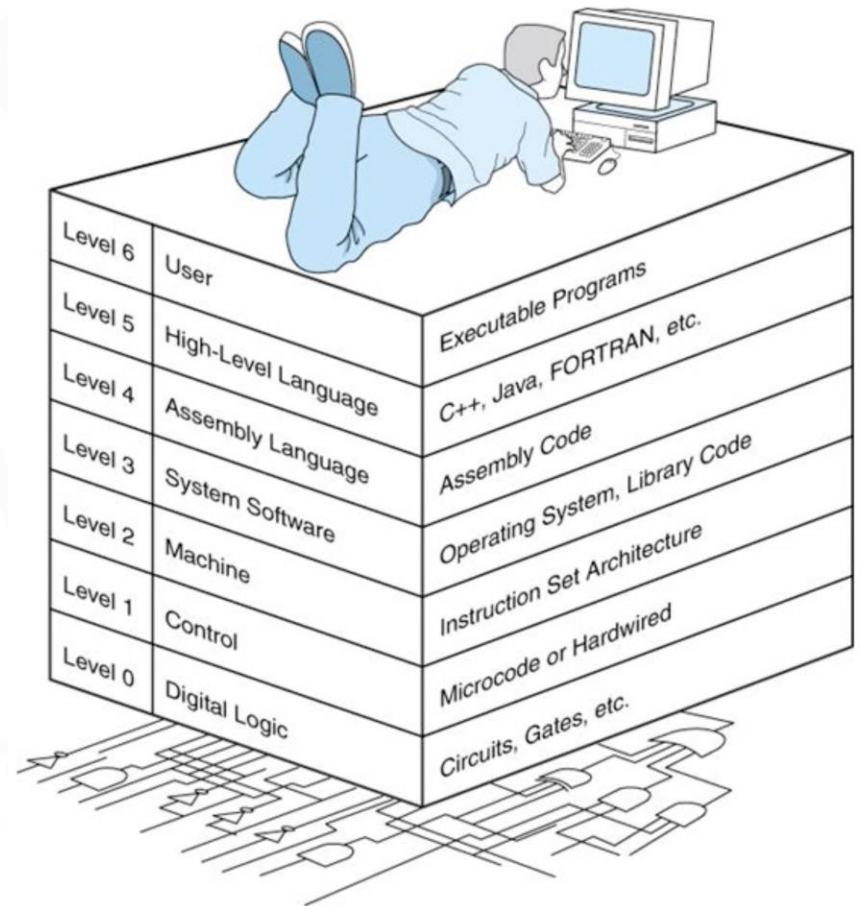
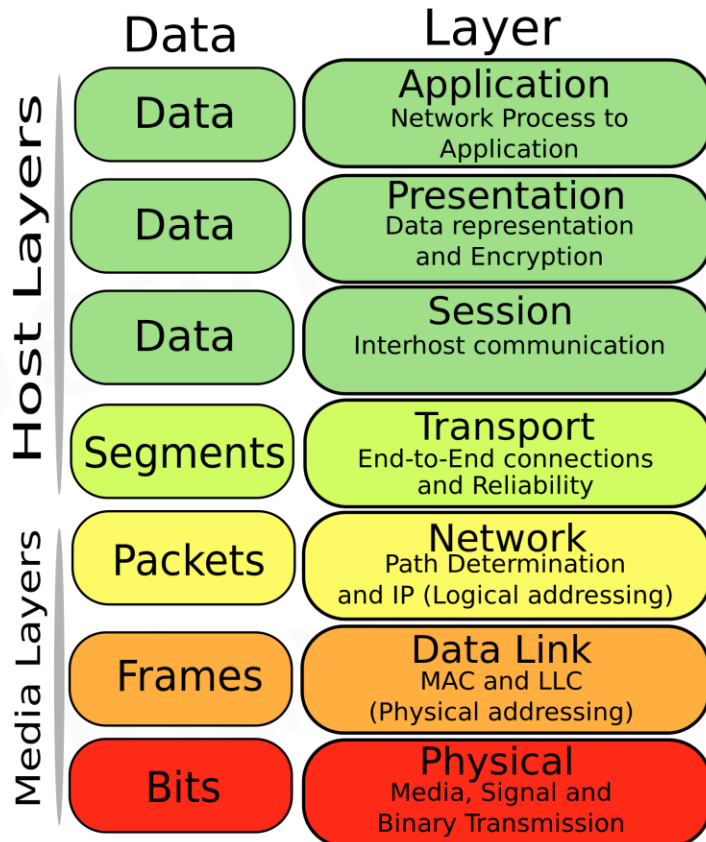


Structuring Computations

- **Macros**
textual substitution (before parsing)
- **Functions**
instantaneous evaluation
- **Processes**
parameterized sequence of actions
- **Actors (\sim concurrent object)**
 - encapsulates a state
 - methods (both actions and expressions)
 - reactions
 - broadcast
- **Functions definition, Process definition, Actors definition, running processes and running Actors are first class values**
they can be argument of a function/method/process call or actor instantiation

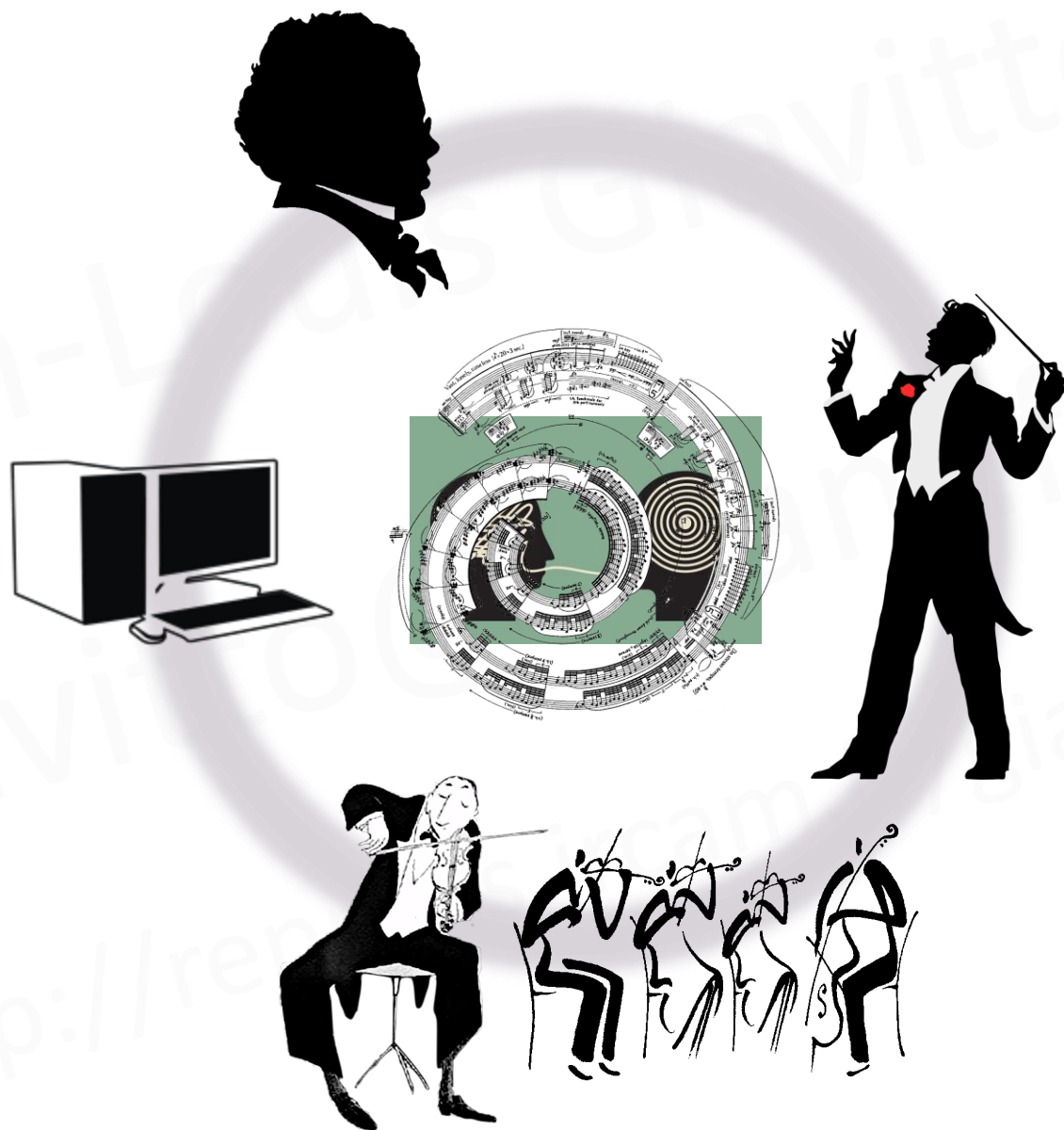
Modèle ISO, une tour de machines abstraites, abstractions...

OSI Model





La partition comme **objet-frontière** ou **objet-intermédiaire**



Jason Freeman (GeorgiaTech)

Shadows, 2015 dynamic non-deterministic score

(excerpt from the score, Jason Freeman)

The computer tracks the choices you make about which fragments to play and how long you play each one. It always highlights the fragment you are currently playing in blue and draws thick black lines to the fragment(s) you can choose to play next.

During your performance, the computer will vary how many choices it gives you about what to play next, as well as how it determines your options. It bases these decisions on which fragments you have played recently and for how long you have played each. At some points in the performance, the computer will push you towards repeating and alternating between the fragments you have played most often. At other points, it will push you to play fragments you have not played recently. Some aspects of the computer's algorithm, such as how many choices to give you and when to change sections, are updated automatically at predetermined times during your performance.



ODEI (performance Les Nuits Sonores, Lyon, 2014)

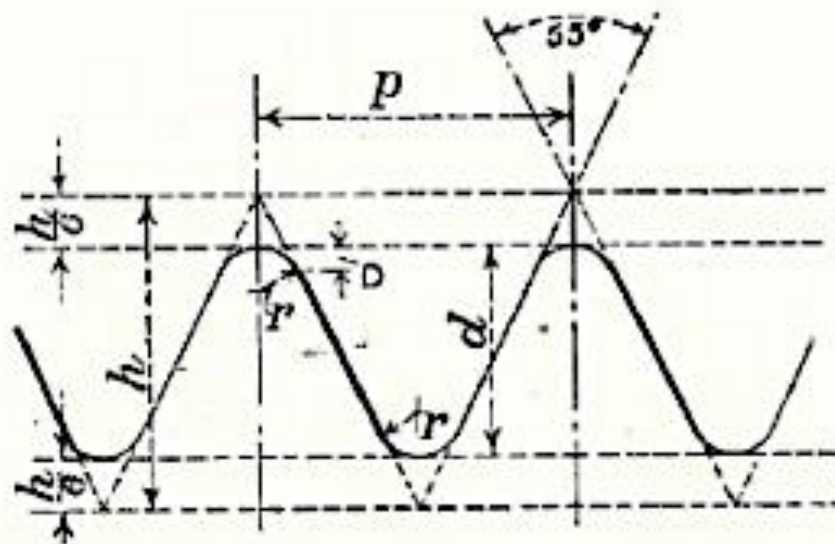


Odei

Improvised electronic music with Antescofo

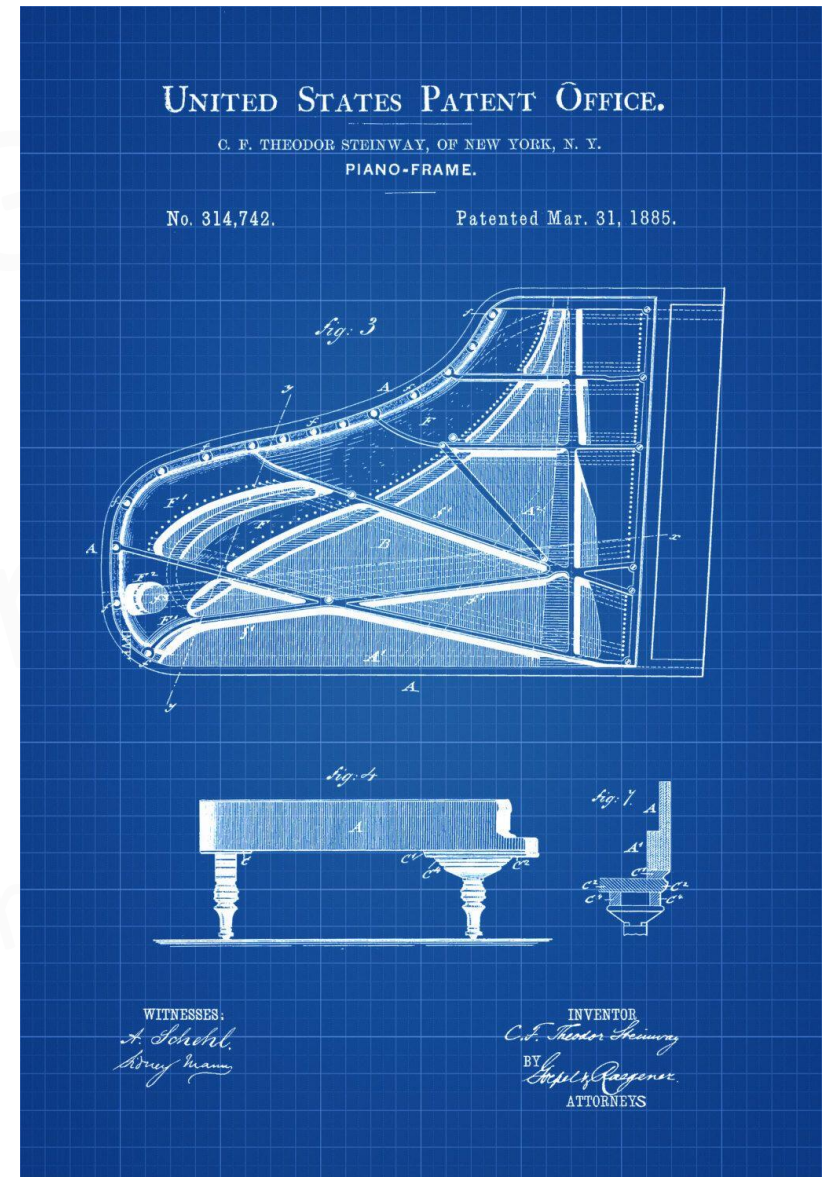
BRITISH STANDARD WHITWORTH SCREW THREADS.

The Whitworth standard screw thread was proposed by Sir Joseph Whitworth and a full table of sizes published in 1841. It was the first recognised screw system and has been employed continuously since its introduction for general engineering purposes. As originally tabulated the sizes commenced from 1/32-inch rising by 1/32nds to 1/4-inch, thence upwards by 1/16ths up to 1 inch, afterwards rising by 1/8ths. The decided merits and general application of the system led to its adoption as a standard by the British Standards Institution. The Whitworth standard pitches are not considered by some authorities to satisfy all requirements, particularly where the screw is subjected to shock or vibration, or where extra strength is required in the core of the screw.



Quel objet-intermédiaire pour la musique avec de l'électronique ?

- les objets techniques *cristallisent* et *capitalisent* un savoir-faire
- ils ont un *impact esthétique* par ce qu'ils facilitent et ce qu'ils rendent difficile
- les objets techniques aujourd'hui sont des *systèmes logiciels*
- ils matérialisent *de facto* un *soft-power*



Can you compose a short tune (say four to eight bars) using ABC notation?
There are many possible ways to compose a short tune in abc notation, but here is one example:

```
X:1
T:Example Tune
M:4/4
L:1/8
K:C
|:G2AB c2BA | G2AB c2c2 | edcB c2BA | G4-G2z2 :|
|:c2de f2ed | c2de f2f2 | g2fe d2cB | c4-c2z2 :|
```

This tune has two four-bar phrases, each repeated once, forming an A part and a B part. The key is C major.

Can you describe the tune in musical terms?

Here is one possible description:

The tune starts with a rising arpeggio of the tonic chord (C major), followed by a descending scale that returns to the tonic. This pattern is repeated, but the second time the scale continues to the higher tonic note. The first phrase ends with a long tonic note.

The second phrase begins with a similar pattern, but using the subdominant chord (F major) instead of the tonic. The second time, the scale continues to the higher subdominant note. The phrase ends with a stepwise descent to the tonic, followed by a long tonic note.

The A part and the B part have a similar structure, but use different chords and ranges. The tune has a simple and catchy melody, with a clear sense of tonality and rhythm.

I like part A, but maybe part B is just a bit too similar to part A, making the song repetitive. How about beginning part B with a descending arpeggio instead of a rising one?

Sure, that would create some contrast and variation. Here is one possible way to do that:

```
X:1
T:Example Tune
M:4/4
L:1/8
K:C
|:G2AB c2BA | G2AB c2c2 | edcB c2BA | G4-G2z2 :|
|:f2ed c2BA | f2ed c2c2 | g2fe d2cB | c4-c2z2 :|
```

That's much better. Now how about making this into a duet, can you add a second staff for this with a bass accompanying the melody?

One possible way to write a second staff for this tune is to use root notes and fifths of the chords, as well as some passing notes, to create a simple accompaniment. Here is an example in abc notation, with the second hand bass on the lower staff:

```
X:1
T:Example Tune
M:4/4
L:1/8
K:C
V:1
|:G2AB c2BA | G2AB c2c2 | edcB c2BA | G4-G2z2 :|
|:f2ed c2BA | f2ed c2c2 | g2fe d2cB | c4-c2z2 :|
V:2 clef=bass
|:C,2C,2 G,2G,2 | C,2C,2 G,2G,2 | F,2F,2 E,2E,2 | C,4-C,2z2 :|
|:F,2F,2 C,2C,2 | F,2F,2 C,2C,2 | G,2G,2 F,2F,2 | E,2C,2 G,2C,2 :|
```

The sheet music of the final tune:

The sheet music consists of two systems. Each system has a treble clef staff (melody) and a bass clef staff (bass). The first system shows the first phrase of the tune, and the second system shows the second phrase. The bass line uses root notes and fifths of the chords, with some passing notes.

