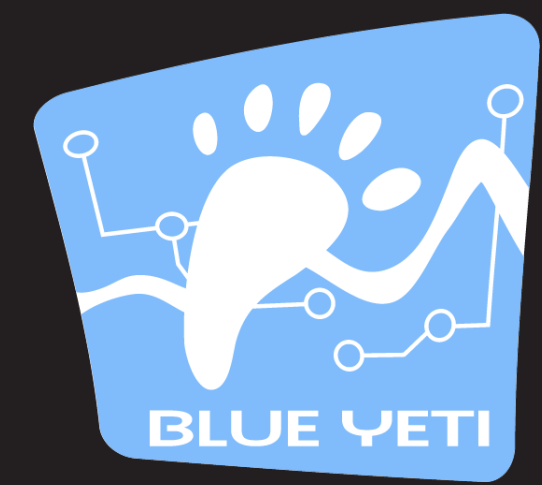


OSSIA: Towards a unified interface for scoring time and interaction

Jean-Michaël Celerier Pascal Baltazar Clément Bossut Nicolas Vuaille
Jean-Michel Couturier Myriam Desainte-Catherine

Laboratoire Bordelais de Recherche en Informatique (LaBRI)



Introduction

- OSSIA : Open Scenario System for Interactive Applications
- Part of the interactive scores research
- Implementation of the OSSIA API and the i-score[1] software
- Research in hardware implementation using FPGAs

Interactive scores

- Scores whose structure depends on interactivity
- Described formally by Antoine Allombert[2], Mauricio Toro[3], and Jaime Arias[4]
- Allows the authoring and execution of hierarchical scores
- Originally, worked with interaction points triggered manually
- We present the implementation of conditionals and loops in interactive scores using a novel graphical and theoretical formalism.

Relevant works of art

- Pierre Boulez : *Third sonata for Piano*. The performer has the freedom to arrange the order of the movements during the interpretation.
- André Boucourechliev : *Archipels*. Performers can associate the different components of the musical interpretation (pitches, durations, dynamics...) as they wish in some parts of the music.
- Multiple interactive audio-visual experiments : *Concert Prolongé*, *Mariona*, *Wandering Listener*.

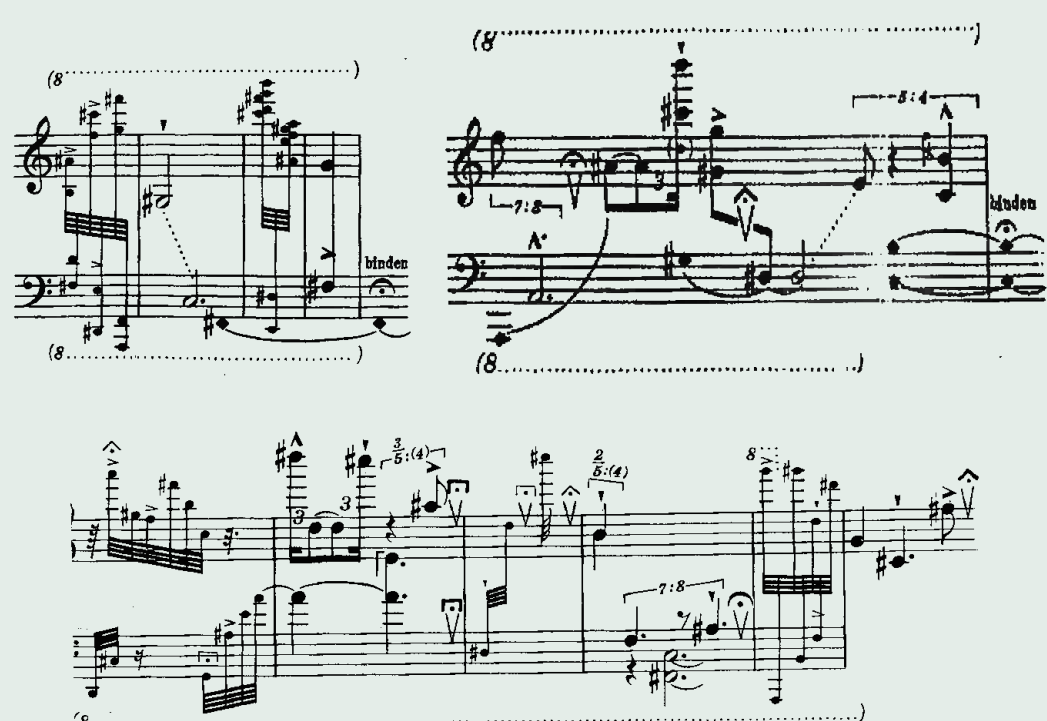


FIGURE 1: Parts of Karlheinz Stockhausen's *Klavierstück XI* : the progression in the song depends on a set of rules on the arrangement of 19 musical sequences. Performers can play them in any combination they wish, and the music ends when they look at a sequence already played twice.

Implementation

- Conception of a graphical formalism able to capitalize on the extensive formal research
- As few concepts as possible to describe time :
 - Span of time : **Constraint**
 - Punctual element : **Event**
 - Synchronization : **TimeNode**
 - Hierarchy, looping, and parameter control : **Processes**
- The software is based on the idea of a time-line similar to other common music production software.
- For now, mostly parameter-centric, and time-oriented (versus more flow-oriented paradigms like OpenMusic or Max/MSP)

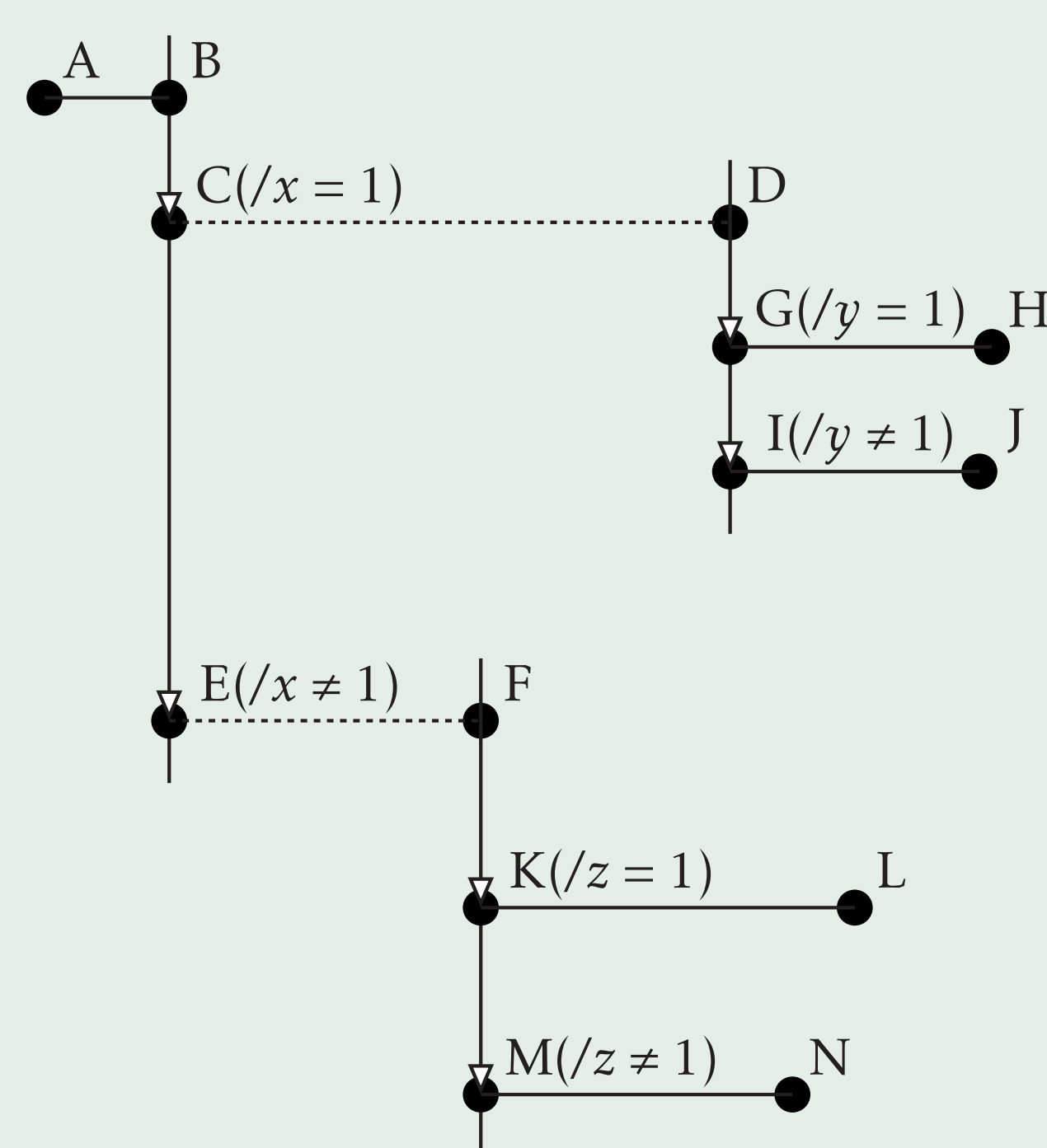


FIGURE 2: Nested if - then - else using flexible constraints.

Flow control

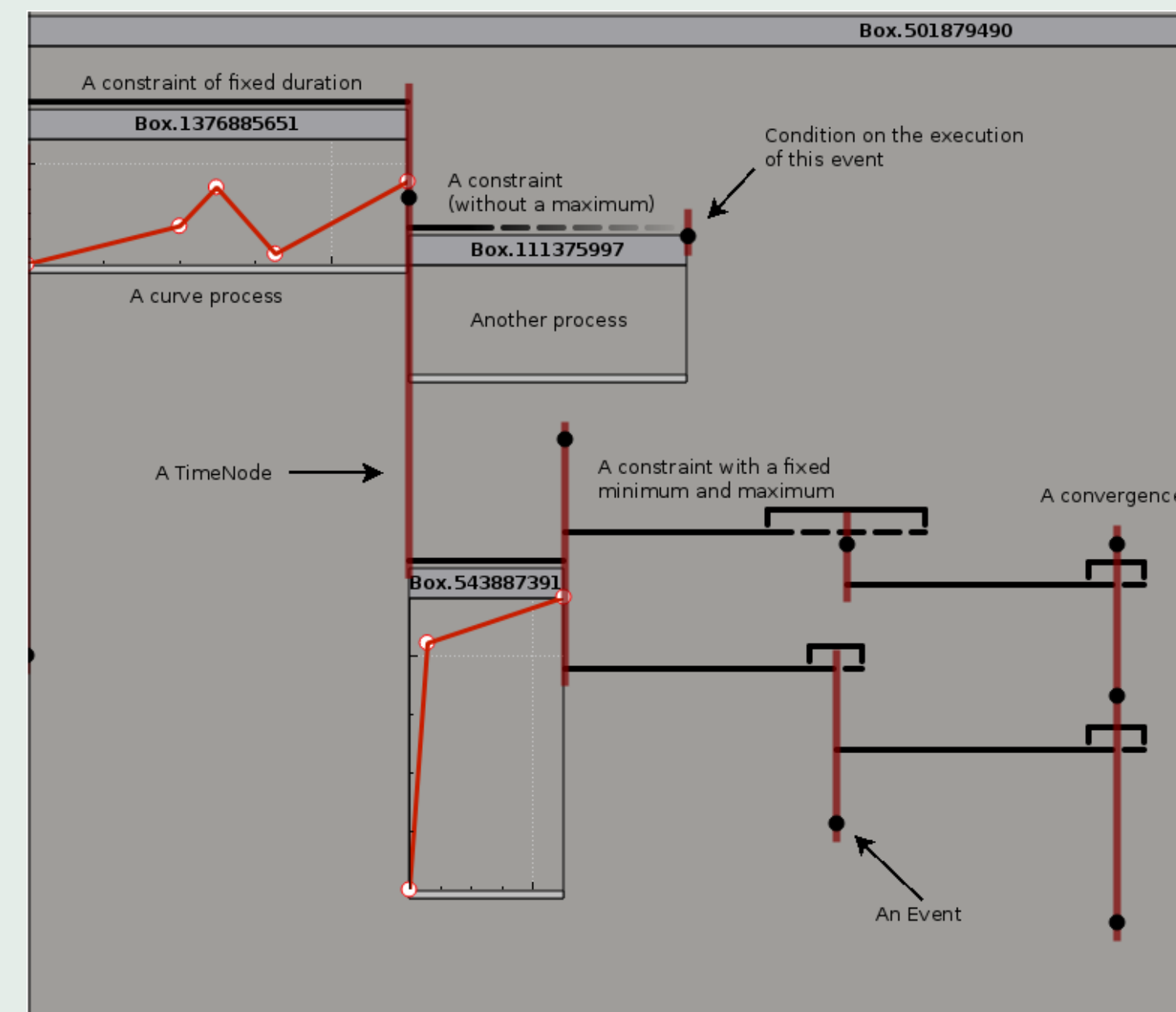
Conditions

- Each event can have a condition on an external parameter
- It is possible to make standard programming constructs (Fig. 2)
- The condition can be validated when all the constraints that finish on the event enter their evaluation range

A proposition for loops

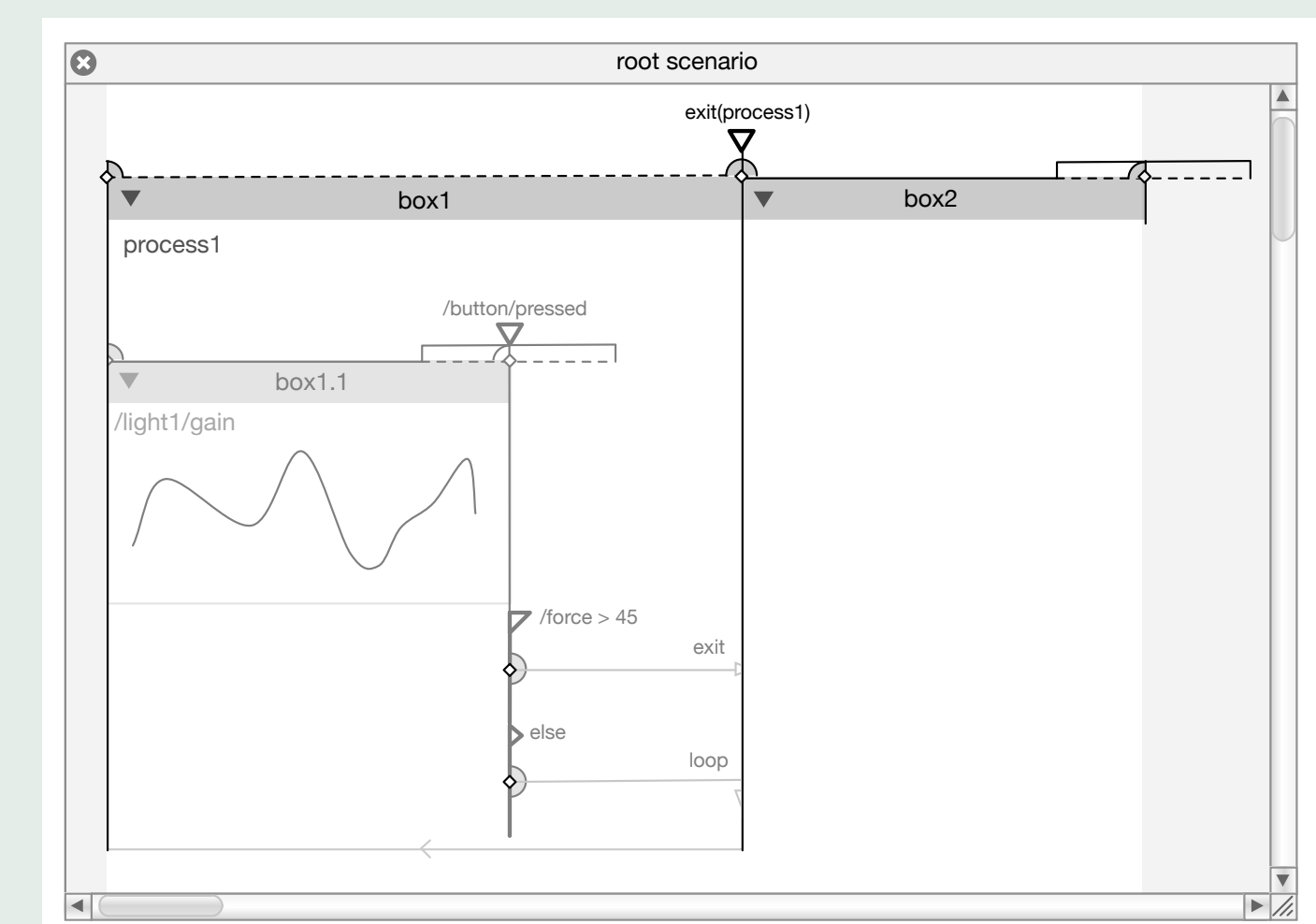
- Implemented as a process
- Requires a pattern (the looping data) and an event (the predicate)
- Clean separation of the duration of the loop and the data of the loop

Examples of scores



LEFT: Example of an interactive scenario with multiple conditions and flexible time in the current development version of i-score. The software implementation is still under development : the conditions on events are present but not displayed. The software can already be tested at <http://www.i-score.org>.

RIGHT: Prototype of a user interface to represent looping in the scores, with a higher-level representation of conditions. This allows to write arbitrarily complex scores very easily, and to conceive full-blown multimedia installations within the software. It alleviates the need for an external tool to handle basic algorithms.



Open towards extensibility

- To allow testing with different back-ends, the software depends on a C++ interface, the OSSIA API, which allows to write such scenarios directly from code. It is a kind of domain-specific language.
- This interface is currently getting a canonical implementation using the Jamoma frameworks, but it should be easily translatable to other programming back-ends.
- i-score is geared towards the inclusion of new kinds of temporal processes. It revolves around a plug-in API which allows to write new processes and adapt to different use cases.

Conclusion

- Presentation of a temporal paradigm able to handle :
 - Traditional scores
 - Museography
 - User interface control
 - Representation of distributed systems
- Extensions being researched :
 - Interoperability with sound and space-centric environments
 - Distributed operation, for edition as well as execution of scores
 - Incorporation of a scripting API in a high-level language
 - Formal verification in the case of loops

Bibliography

- [1] P. Baltazar, T. de la Hogue, and M. Desainte-Catherine, "i-score, an interactive sequencer for the intermedia arts," in *Submitted to the joint ICMC/SMC conference*, (Athens, Greece), 2014.
- [2] A. Allombert, *Aspects temporels d'un systeme de partitions musicales interactives pour la composition et l'interprétation*. PhD thesis, Université de Bordeaux, 2009.
- [3] M. Toro, M. Desainte-Catherine, and C. Rueda, "Formal semantics for interactive music scores : a framework to design, specify properties and execute interactive scenarios," *Journal of Mathematics and Music*, vol. 8, no. 1, pp. 93–112, 2014.
- [4] J. Arias, M. Desainte-Catherine, and C. Rueda, "A framework for composition, verification and real-time performance of multimedia interactive scenarios," in *15th International Conference On Applications Of Concurrency To System Design (ACSD'15)*, 2015.