Exécution répartie de partitions interactives

Jean-Michaël Celerier

Laboratoire Bordelais de Recherche en Informatique, Blue Yeti

Problématique

Étant donné une partition interactive, quelles sémantiques peut-on introduire pour permettre l'exécution de sous-parties de cette partition sur des machines distinctes, en permettant de nouvelles formes d'écriture.

Partitions interactives

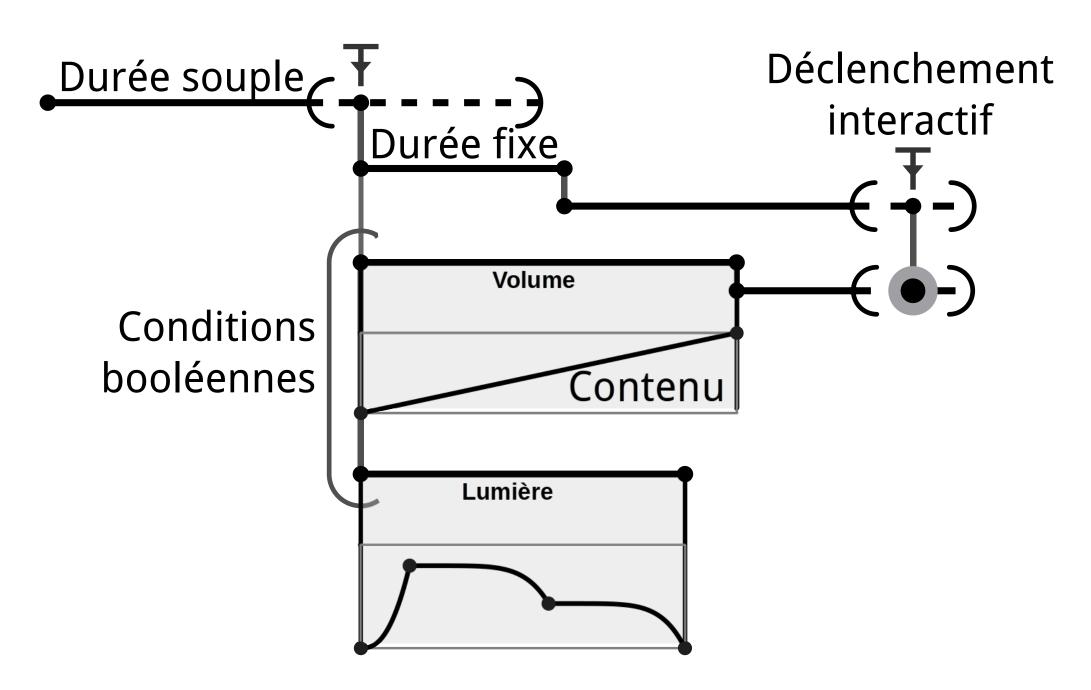


FIGURE 1: Syntaxe d'une partition interactive

Possibilités d'écritures forment un langage de programmation structuré axé sur l'organisation temporelle. **Boucles** et **hiérarchie**, calcul instantané ou temporel possible via **Javascript**. Applications : musique interactive, scénographie et spectacle vivant, contrôle de robots. Autres approches graphiques : **OpenMusic** [3], **Antescofo** [5], **INscore** [6]; ainsi qu'approches programmatiques : **Abjad**, **Tuiles réactives**, ...

Existant

Modélisation algorithmique : réseaux de Petri pour synchronisation. Horloges : NTP, PTP, Hybrid Clocks, Ableton Link. Partage de flux : NetJACK Applications musicales réparties : Ohm Studio, Impromptu. Pas d'outil de répartition des structures musicales.

Méthode

Conception en entité-composant-système avec hiérarchies symétriques d'entités et de composants. Plusieurs moteurs opèrent en parallèle, avec une conception modulaire pour étendre le modèle.

Création d'entités sémantiques fortes par héritage, puis d'extensions faibles par composition. Hiérarchie : création automatique de composants enfants à la création de nouvelles entités.

Synchronisation des macro-structures

Programme temporel : **scénario**. Trois modes d'exécution :

- Indépendant
- Partagé
- Mixte

Synchronisation des micro-structures

- Mode synchrone
- Mode asynchrone

Expressions et consensus

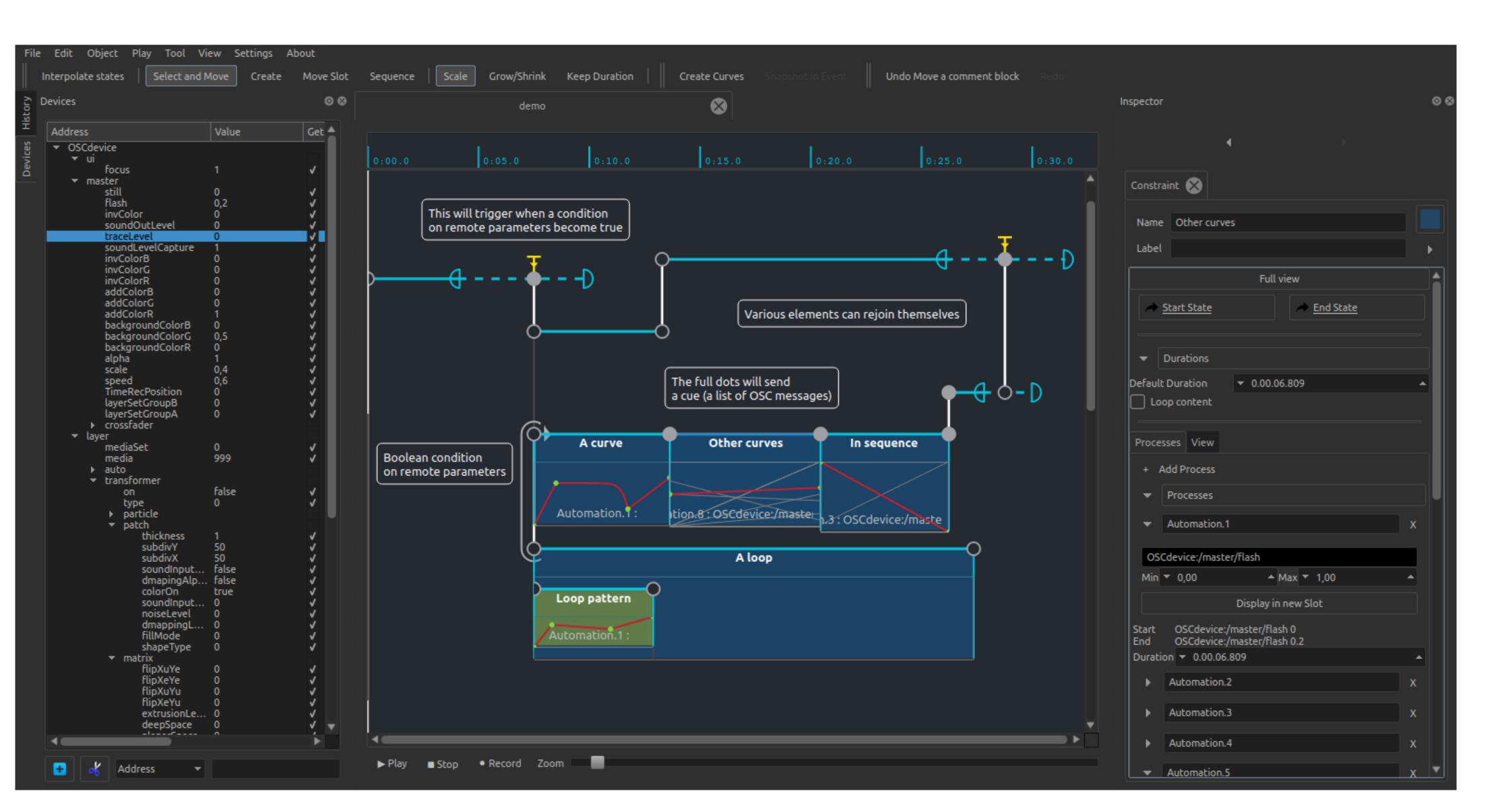


FIGURE 2 : Un scénario d'exemple i-score

Résultats

Plusieurs moteurs sont implémentés de cette manière : il est en pratique souvent nécessaire d'avoir des graphes miroirs du graphe principal avec des données supplémentaires.

Optimisation pour temps fixe Implémentation

- · Scripting intégré : intégration d'un langage de haut-niveau interprété (QML) pour prototyper plus simplement sur le logiciel.
- Implémentation audio plus poussée en utilisant libaudiostream[8] qui crée un graphe semblable à celui d'i-score.
- Travail en cours sur modèles et processus spatiaux : écriture avec des données multi-dimensionnelles.

Informations complémentaires

Articles sur ce sujet :

- Modèles formels sur lesquels se base i-score :
 [1, 2].
- Paradigme graphique OSSIA : [4].
- i-score peut être téléchargé librement sur
- ·www.i-score.org

Références

- [1] Antoine Allombert, Gérard Assayag et Myriam Desainte-Catherine. "A system of interactive scores based on Petri nets". In: 4th Sound and Music Computing Conference (SMC07). 2007, p. 158–165. (Visité le 02/11/2015).
- [2] Jaime Arias, Myriam Desainte-Catherine et Camilo Rueda. "Modelling Data Processing for Interactive Scores Using Coloured Petri Nets". In: IEEE, juin 2014, p. 186–195. ISBN: 978-1-4799-4281-7. DOI: 10.1109/ACSD.2014.23. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7016342 (visité le 02/11/2015).
- [3] Jean Bresson, Carlos Agon et Gérard Assayag. "OpenMusic: Visual Programming Environment for Music Composition, Analysis and Research". In: Proceedings of the 19th ACM International Conference on Multimedia. MM '11. Scottsdale, Arizona, USA: ACM, 2011, p. 743–746.
- [4] Jean-Michaël Celerier et al. "OSSIA: Towards a unified interface for scoring time and interaction". In: TENOR: First International Conference on Technologies for Music Notation and Representation, Paris, France. 2015. (Visité le 02/11/2015).
- [5] Arshia Cont. "ANTESCOFO: Anticipatory Synchronization and Control of Interactive Parameters in Computer Music." In: International Computer Music Conference (ICMC). 2008, p. 33–40.
- [6] Dominique Fober, Yann Orlarey et Stéphane Letz. "An environment for the design of live music scores". In: Proceedings of the Linux Audio Conference, CCRMA, Stanford University, California, US. 2012, p. 47–54.
- [7] Chris LAFFRA et al. *Object-oriented programming for graphics*. Springer Science & Business Media, 2012.
- [8] Stéphane LETZ. "Spécification de l'extension LibAudioStream". In : (2014). URL : https://hal.archives-ouvertes.fr/hal-00965269/(visité le 22/07/2015).