

STRUCTURED INTERACTIVE MUSICAL SCORES

MAURICIO TORO-BERMÚDEZ¹

¹ Université de Bordeaux 1, Laboratoire Bordelais de Recherche en Informatique, Bâtiment A30.
351, cours de la Libération F-33405 Talence cedex, France.

URL: <http://www.labri.fr/perso/mtoro/>

ABSTRACT. Interactive Scores is a formalism for the design and performance of interactive scenarios that provides temporal relations (TRs) among the objects of the scenario. We can model TRs among objects in Time Stream Petri nets, but it is difficult to represent global constraints. This can be done explicitly in the Non-deterministic Timed Concurrent Constraint (ntcc) calculus. We want to formalize a heterogeneous system that controls in one subsystem the concurrent execution of the objects using ntcc, and audio and video processing in the other. We also plan to develop an automatic verifier for ntcc.

Introduction

Interactive Scores (IS) are currently used for the design and performance of Electroacoustic music [All08a] and live spectacles [Bal09] (e.g., interactive theater plays and interactive museums). Both applications are based on Petri nets [All08b]. The main purpose of IS is to provide temporal relations; for instance, precedence between two objects and relations between their durations. Recently, we extended IS to support conditional branching together with temporal relations [TB10]. It is now possible to represent loops and choices.

We can model temporal relations in *Time Stream Petri nets (TSPN)* [Sen95], but it is difficult to represent global constraints involving (possibly) all the objects of the scenario. Instead, in *Concurrent Constraint Programming (ccp)* [Sar92] there are agents that reason about partial information contained in a constraint *store*; thus, global constraints are inherent in ccp. However, there is not discrete time in ccp, which makes it difficult to represent reactive systems.

There are some IS models based on extensions of ccp with discrete time. An example is the *Non-deterministic Timed Concurrent Constraint (ntcc)* model of IS [Nie02, All06]. Ntcc is an extension of ccp for non-determinism, asynchrony and discrete time. In the declarative view, ntcc processes can be interpreted as *linear temporal logic* formulae [Pnu77]. The ntcc model includes an inference system in this logic to verify properties of ntcc models. This inference procedure was proved to be of exponential time complexity. Nevertheless, we believe practical automatic verification could be envisioned for useful subsets of ntcc via model checking (see [Fal06]). At present, there is no such automatic verifier for ntcc.

1998 ACM Subject Classification: D. Software, D.1 Programming techniques, D.1.3 Concurrent programming, D.1.5 Logic programming.

Key words and phrases: ntcc, ccp, interactive scores, temporal relations, faust, ntcrt, heterogeneous systems, automatic verification.

Automated verification for `ntcc` will provide information about the correctness of the system to computer scientists, and will provide important properties about the scenario to its designers and users; for instance, reachability and liveness. We plan to augment `ntcc` models of IS with these features.

1. Current and future work

Functional AUdio Stream (Faust) [Orl04] is a programming language for signal processing with formal semantics and *Ntccrt* [TB09] is a real-time capable interpreter for `ntcc`. We implemented a signal processing prototype where Faust and Ntccrt interact together. In the future, we want to define formal semantics to describe a heterogeneous system that includes three subsystems: (i) one based on `ntcc` to control discrete events from the user and to synchronize the objects of the scenario, (ii) another one based on Faust to process audio and video, and finally (iii) one in charge to load and play audio and video files.

At the time of this writing, there are no formal semantics of a heterogeneous system that synchronizes concurrent objects, handles global constraints, and controls audio and video streams. Modeling this kind of systems will be useful in other domains such as *machine musical improvisation* [Ass04] and *music video games*. An advantage over the existing implementations of these systems will be verification.

In the proof system of `ntcc`, we can prove properties like “10 time units (TUs) after the event e_A , during the next 4 TUs, the stream B is the result of applying a *gain filter* to the stream A ”. However, real-time audio processing cannot be implemented in Ntccrt because it requires to simulate 44100 TUs per second to process a 44.1 kHz sound. If we replace some `ntcc` processes by Faust plugins, we can execute such system efficiently, but we cannot verify that the properties of the system hold.

There are two open issues: (i) how to prove that a Faust plugin that replaces a `ntcc` process respect the temporal properties proved for the process, and (ii) whether an implementation of Interactive Scores in Ntccrt can be as efficient as the existing Petri nets implementation, or as one using synchronous languages such as *Signal* [Gau87], although the performance results from Ntccrt are promising¹.

Acknowledgement

I wish to acknowledge fruitful discussions with my supervisors Myriam Desainte-Catherine and Camilo Rueda. I also want to thank them for helping me writing this article and for guiding my current research.

References

- [All06] Antoine Allombert, Gérard Assayag, M. Desainte-Catherine, and Camilo Rueda. Concurrent constraint models for interactive scores. In *Proc. of SMC '06*. 2006.
- [All08a] Antoine Allombert, G. Assayag, and M. Desainte-Catherine. Iscore: a system for writing interaction. In *Proc. of DIMEA '08*, pp. 360–367. ACM, New York, NY, USA, 2008.
- [All08b] Antoine Allombert, Myriam Desainte-Catherine, J. Larralde, and Gérard Assayag. A system of interactive scores based on qualitative and quantitative temporal constraints. In *Proc. of Artech 2008*. 2008.

¹We ran a prototype of a score with conditional branching in Ntccrt. The score contains 500 temporal objects. The average duration of each time-unit was 30 ms, which is compatible with real-time interaction.

- [Ass04] G. Assayag and Sholomo Dubnov. Using factor oracles for machine improvisation. *Soft Comput.*, 8(9):604–610, 2004.
- [Bal09] P. Baltazar, A. Allombert, R. Marczak, J.M. Couturier, M. Roy, A. Sèdes, and M. Desainte-Catherine. Virage : Une reflexion pluridisciplinaire autour du temps dans la creation numerique. In *in Proc. of JIM*. 2009.
- [Fal06] Moreno Falaschi and Alicia Villanueva. Automatic verification of timed concurrent constraint programs. *Theory Pract. Log. Program*, 6(4):265–300, 2006.
- [Gau87] Thierry Gautier, Paul Le Guernic, and L oic Besnard. Signal: A declarative language for synchronous programming of real-time systems. In *Proc. of FPCA '87*. 1987.
- [Nie02] M. Nielsen, C. Palamidessi, and F. Valencia. Temporal concurrent constraint programming: Denotation, logic and applications. *Nordic Journal of Comp.*, 1, 2002.
- [Orl04] Y. Orlarey, D. Fober, and S. Letz. Syntactical and semantical aspects of faust. *Soft Comput.*, 8(9):623–632, 2004.
- [Pnu77] A. Pnueli. The temporal logic of programs. In *In Proc. of the 18th IEEE Symposium on the Foundations of Computer Science (FOCS-77), pages 46-57. IEEE, IEEE Computer Society Press, 1977*. 1977.
- [Sar92] Vijay A. Saraswat. *Concurrent Constraint Programming*. MIT Press, 1992.
- [Sen95] Patrick Senac, Pierre de Saqui-Sannes, and Roberto Willrich. Hierarchical time stream petri net: A model for hypermedia systems. In *in Proc. of the 16th International Conference on Application and Theory of Petri Nets*, pp. 451–470. Springer-Verlag, London, UK, 1995.
- [TB09] Mauricio Toro-B., Carlos Ag on, G erard Assayag, and Camilo Rueda. Ntcrcrt: A concurrent constraint framework for real-time interaction. In *Proc. of ICMC 09*. 2009.
- [TB10] Mauricio Toro-B., Myriam Desainte-Catherine, and P. Baltazar. A model for interactive scores with temporal constraints and conditional branching. In *Proc. of Journ ees d'informatique musical (JIM) '10 (to appear)*. 2010.