

Modelling chronicle recognition for distributed simulation processing with coloured Petri nets (Poster abstract)

Olivier Bertrand(1), Patrice Carle(1), Christine Choppy(2)

(1) Onera DPRS/TCS, Chatillon, FRANCE

email: {olivier.bertrand,patrice.carle}@onera.fr

(2) LIPN, UMR CNRS 7030, Université Paris XIII, FRANCE

email: {Christine.Choppy}@lipn.univ-paris13.fr

ABSTRACT

The context of our work is the complex systems simulation. Complex systems modelling and simulation require to take into account the interactions between the different components. For instance, an airport ground traffic simulation requires to model the planes with the pilots, the control tower, the weather, etc. together with their interactions, e.g. the communications between the pilots and the control tower (authorizations and acknowledgements), the weather impact on the routing rules, etc.

Air transport systems undergo continuous evolution in order to adapt to future performance, quality and safety requirements. This implies that the operational use of new techniques in navigation, communications, surveillance, regulations and computer based assistance/automation systems. In the context of the future air transportation systems, airports will play a critical role since they have to face serious environmental (noise impact) and operational restrictions which reduce their potential capacity. In order to meet the projected airport demand and to improve efficiency of ground operations while maintaining safety, several new operational concepts are currently under investigation, including new regulations or operational concepts, and advanced equipment or computer-based assistance systems. The French Aerospace Lab (Onera) is participating in this effort through the design of a modeling and simulation infrastructure for the acquisition and analysis of airport operational concepts and equipment.

This simulation is a distributed simulation that is built with the different specialized simulators (by example one simulator manages the different airplane position, another display airplane...). These simulators communicate between them with a computer network. The communication protocol used is High Level Architecture norm. HLA normalizes the communication between the simulation components. The communication between simulation components are exchanges of objects (they have a duration and a life cycle with

creation, update and destruction) and interactions (message with no duration). The simulation architecture is in Figure 1. The different circles represent the different simulators. In our example we mainly use the communication emitted by the TR simulator. The TR simulator manages the airplanes in the simulation.

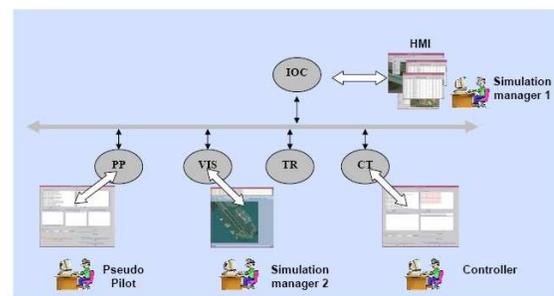


Figure 1: Future Airport simulation architecture

The originality of this simulation is to provide an open and scalable HLA simulation supporting an easy plug of additional components in order to analyse and compare the efficiency of several airport configurations. We chose the Future Airport application because it is representative of the HLA simulations that are used at Onera. Indeed, it is a complex system simulation for the evaluation of the performances of the simulated system. The analysis of this kind of simulation requires the availability of a search process for undesired behaviours (in this example, collisions or simultaneous uses of a runway), and of a measurement process for the realization of some behaviours (for instance, the time elapsed between the landing of a plane and its stop at its gate). Moreover, this Future Airport application is designed to be extensible, and the analysis components should be easily extended without hindering the extension possibilities of the simulation. The chosen approach is to use an additional component in the simulation that can detect the searched activities from observations of the simulation communication.

For this purpose, we chose a temporal approach that provides a better expression power, and the possibility to detect behaviour realisations through the observation of their characteristic traces. We consider each interaction as an event. The characteristic traces are expressed through event

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIMUTools 2009, Rome, Italy

Copyright 2009 ICST, ISBN 978-963-9799-45-5.

correlations. We chose to use *chronicles* for a fine detection of event correlations that represent behaviour realisations. In this approach, events cannot occur simultaneously, and durations are not associated with events, events are thus considered as instantaneous. The event timestamps are achieved by the observation process, and are therefore expressed in terms of the observer time that is independent of the observed system own time. Hence chronicles do not express absolute dates, but delays between two events that are relative to the observer.

Chronicles are used to model activities to be detected, for instance because they are undesired or dangerous (an example is an airport monitoring). An activity is described by a combination of event occurrences. When these events are identified within a flow of events and in the described combination, then this activity is detected. A chronicle describes relationships between the events of a sequence ordered with respect to time. The goal is to identify all instances of the chronicle schema within an observed event flow (where events are ordered and time-stamped). The chronicle identification is achieved through the matching between events of the flow and events in the chronicle description, while flow events that do not contribute to the chronicle recognition are simply ignored. In addition, it may be of interest to save the piece of information stating which events in the flow contributed to the chronicle recognition, because it may help to find which are the causes of the observed events.

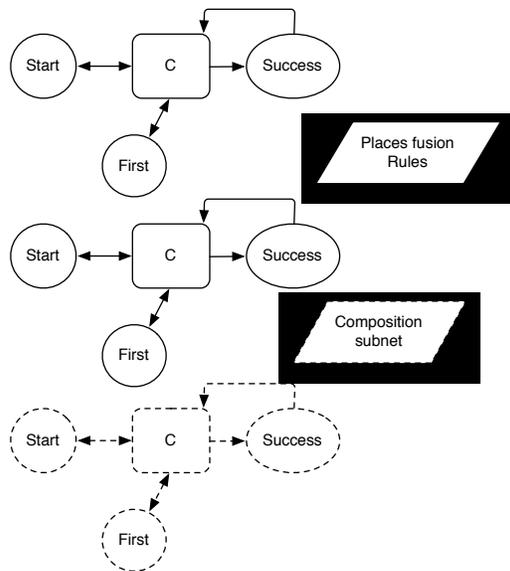


Figure 2: Composition schema for the Petri nets

The relationships between the events may be logical or temporal, and are expressed in a chronicle language [3]. The basic principles of our modelling with coloured Petri nets are that we associate a transition to each event that should be detected (and this transition is fired whenever this event occurs), and places are used to store (in a single token) a list of all chronicle (partial or total) instances identified. In addition, some control places are used to ensure that the desired sequencing is followed, or to model discrete time.

For each operator expression of the chronicle language we provide a coloured Petri net modelling its recognition, and we provide a composition schema of these nets (cf. Figure 2).

We developed a tool [2] that uses chronicle recognition to facilitate the exploitation and the supervision of HLA distributed simulations. This analysis tool processes the normalised exchanges between simulators of an HLA simulation. The facts that chronicles are used, and that the simulators exchanges are used as event sources leads to building non intrusive analysis tools for the simulators.

In [1] we compare various kinds of automata (counter automata, duplicating automata) and coloured Petri nets for expressing and recognizing some basic chronicles operators, and found that coloured Petri nets comply with the specific needs associated with chronicle recognition. In particular, with coloured Petri nets we can model multiple occurrences recognition, and trace the events that contributed to a recognition while providing a modular and visual presentation. The goal is to perform analysis of system simulations of type High Level Architecture (HLA), to be used e.g. in airport control, and we started to perform off line analysis of distributed HLA simulations using chronicles [2]. In this work, we pursued to reach an extensive expression of the chronicle language operators, and then worked out coloured nets that could be composed to express complex chronicle expressions. The composition could be achieved either via transition substitution (a first approach is in [3]), or via place fusion, and in both cases the algebraic properties of the operators are reflected. Our model thus encompasses all logical and temporal operators, including the absence operator.

This work thus provides elements to model complex chronicles recognition, and to define a semantics of the chronicle language in terms of the used subset of coloured nets.

In a near future, we plan to model the transition firing with Petri net to comply the “classical” rules of Petri nets.

Keywords

chronicle recognition formal modelling, coloured Petri nets, distributed simulation processing, HLA, Airport simulation

1. REFERENCES

- [1] O. Bertrand, P. Carle, and C. Choppy. Chronicle modelling using automata and colored Petri nets. In *The 18th International Workshop on Principles of Diagnosis (DX-07)*, pages 229–234, 2007.
- [2] O. Bertrand, P. Carle, and C. Choppy. Reusing simulation through processing tools. In *Simulation Interoperability Workshop*, june 2008.
- [3] O. Bertrand, P. Carle, and C. Choppy. Towards a coloured Petri nets semantics of a chronicle language for distributed simulation processing. In *CHINA 2008 Workshop (Concurrency metHods: Issues aNd Applications)*, pages 105–119, June 2008.