Formal Treatment of Data Structures in Concurrency Models

Nicolas Guelfi

L.G.L.
Software Engineering Laboratory
Swiss Federal Institute of Technology
IN (Ecublens)
CH-1015 Lausanne
Switzerland

voice : 41.21.693.67.68
fax : 41.21.693.50.79
e_mail : Nicolas.Guelfi@di.epfl.ch

Fabrice Mourlin

L.R.I.
Laboratoire de Recherche en Informatique
U.R.A 410 du CNRS
Bât. 490, University of Paris XI
91405 Orsay
France

voice : 33.1.69.41.65.91
fax : 33.1.69.41.65.86
e_mail : Fabrice.Mourlin@lri.fr

Abstract *

The problem of real size system modelling with basic concurrency models like transition systems, Petri nets or process algebra is that there is no possibility to model easily the system data structures. Algebraic specification is a method well adapted to this last aspect and has been chosen to be mixed with the two major concurrency models which are process algebra and Petri nets. We present in this paper how the 'mixing' is done and what are the consequences concerning concurrency through the case of Lotos and Algebraic nets. Our approach is built on a simple case study which is given in several versions with both formalisms. The first one uses the basic models without algebraic specifications to highlight the data structures modelling problem. The second one is used to present how algebraic specifications are integrated in each concurrency model and the last one focuses on the concurrency aspects. Several propositions are then given in order to enhance the mixing of algebraic specifications and Petri nets or process algebra.

Keywords : Petri net, process algebra, data types, algebraic specification, algebraic net, Lotos.

1. Introduction

The problem of real size system modelling with basic concurrency models[1] (BCM) like transition systems, Petri nets or process algebra is related to the modelling of complex systems in a simple way. As the expressive power of the fundamental concepts of these basic concurrency models is very poor (this does not mean not expressive) then complex systems need complex models. In fact BCM are well adapted to the modelling of simple systems especially because no classification nor modular constructs are directly available in the CM. For examples, in order to describe the set of natural numbers we have to model explicitly each value, if a set of similar events is to be modelled then all events must be defined.

When complex systems are developed with CM, it could be interesting to describe 'easily' all the possible events, their relations with the system state and their order relation during execution. That's why the many extensions of BCM have focused on notions like : data structures (data classification, data composition) and structuring primitives (modularity, genericity, composition, inheritance, ...). The case of data structures is especially interesting because is has lead to similar extensions in different BCM and concerns the system state modelling as well as the system events. More precisely, process algebra and Petri nets have been extended in LOTOS and Algebraic nets especially by using algebraic specification of data structures.

Algebraic specification allows the abstract specification of data structure in a property oriented way. Syntactically a type is defined by some set names and operation signatures with a set of logical formula. The formula express the minimal properties that must be verified by the operations. Semantically, a type is usually an algebraic structure made of mathematical sets and functions associated to the syntactic set names and functions signatures. Furthermore, this algebraic structure must have the properties logically specified to be considered as a model for the type.

This paper presents the evolution of the Petri nets and Process algebra BCM to high level CM using algebraic specifications for the definition of data structures. After a short introduction, the second part presents the Petri net and process algebra in order to show the difficulties of specifying systems using complex data structures. The third part presents the algebraic specifications for the specification of data structures while the fourth part introduces algebraic nets and Lotos. The last part is devoted to the comparative study of Algebraic nets and Lotos using a rather simple case study chosen for its capability to illustrate the modelling concepts rather than the expressive power of the respective formalisms.

** This work has been sponsored partially by the Esprit Long Term Research Project 20072 ``Design for Validation'' (DeVa) with the financial support of the OFES (Office Fédéral pour l'Education et de la Science).*

[1] we should use the term meta-model to indicate the language definition (like place, transition, arcs, labels and association rules)