Verification of EB³ Specifications Using CADP

Dimitris Vekris¹,⋆, Frédéric Lang², Catalin Dima¹, and Radu Mateescu²

¹ LACL, Université Paris-Est
61, av. du Général de Gaulle, F-94010 Créteil, France
{Dimitrios.Vekris,Catalin.Dima}@u-pec.fr

² Inria Grenoble Rhône-Alpes and LIG – CONVECS Team
655, av. de l’Europe, Montbonnot, F-38334 Saint Ismier, France
{Frederic.Lang,Radu.Mateescu}@inria.fr

Abstract. EB³ is a specification language for information systems. The core of the EB³ language consists of process algebraic specifications describing the behaviour of the entities in a system, and attribute function definitions describing the entity attributes. The verification of EB³ specifications against temporal properties is of great interest to users of EB³. In this paper, we propose a translation from EB³ to LOTOS NT (LNT for short), a value-passing concurrent language with classical process algebra features. Our translation ensures the one-to-one correspondence between states and transitions of the labelled transition systems corresponding to the EB³ and LNT specifications. We automated this translation with the EB³2LNT tool, thus equipping the EB³ method with the functional verification features available in the CADP toolbox.

1 Introduction

The EB³ method [10] is an event-based paradigm tailored for information systems (ISs). EB³ has been used in the research projects SELKIS [19] and EB³SEC [17], whose primary aim is the formal specification of ISs with security policies. In the EB³SEC project, real banking industry case studies have been studied, describing interaction with brokers, customers and external financial systems. The SELKIS project deals with two case studies from the medical domain. The first one draws data records from medical imaging devices. The access to these records is done via web-based applications. The second one deals with availability and confidentiality issues for medical emergency units evolving in a great mountain range, like the Alps in that case.

A typical EB³ specification defines entities, associations, and their respective attributes. The process algebraic nature of EB³ enables the explicit definition of intra-entity constraints, making them easy for the IS designer to review and understand. Yet, its particular feature compared to classical process algebras,

⋆ Partially supported by SELKIS ANR Project. The LACL group is grateful to CONVECS for its warm welcome to INRIA Grenoble in 2012 and wishes to thank all its members for useful advice and discussions.
such as CSP [15], lies in the use of attribute functions, a special kind of recursive functions evaluated on the system execution trace. Combined with guards, attribute functions facilitate the definition of complex inter-entity constraints involving the history of events. The use of attribute functions simplifies system understanding, enhances code modularity, and streamlines maintenance. However, given that ISs are complex systems involving data management and concurrency, a rigorous design process based on formal specification using EB$^3$ must be completed with effective formal verification features.

Existing attempts for verifying EB$^3$ specifications are based on translations from EB$^3$ to other formal methods equipped with verification capabilities. A first line of work [13,14] focused on devising translations from EB$^3$ attribute functions and processes to the B method [2], which opened the way for proving invariant properties of EB$^3$ specifications using tools like Atelier B [6]. Another line of work concerned the verification of temporal logic properties of EB$^3$ specifications by means of model checking techniques. For this purpose, the formal description and verification of an IS case-study using six model checkers was undertaken in [9,5]. This study revealed the necessity of branching-time logics for accurately characterizing properties of ISs, and also the fact that process algebraic languages are suitable for describing the behaviour and synchronization of IS entities. However, no attempt of providing a systematic translation from EB$^3$ to a target language accepted as input by a model checker was made so far.

In this paper, we aim at filling this gap by proposing a translation from EB$^3$ to LNT [7], a new generation process algebraic specification language inspired from E-LOTOS [16]. As far as we know, this is the first attempt to provide a general translation from EB$^3$ to a classical value-passing process algebra. It is worth noticing that CSP and LNT were already considered in [9] for describing ISs, and identified as candidate target languages for translating EB$^3$. Since our primary objective was to provide temporal property verification features for EB$^3$, we focused our attention on LNT, which is one of the input languages accepted by the CADP verification toolbox [11], and hence is equipped with on-the-fly model checking for action-based, branching-time logics involving data.

At first sight, given that EB$^3$ has structured operational semantics based on a labelled transition system (LTS) model, its translation to a process algebra may seem straightforward. However, this exercise proved to be rather complex, the main difficulty being to translate a history-based language to a process algebra with standard LTS semantics. To overcome this difficulty, we considered alternative memory-based semantics of EB$^3$ [20], which were shown to be equivalent to the original trace-based semantics defined for finite-state systems in [10]. Another important ingredient of the translation was the multiway value-passing rendezvous of LNT, which enabled to obtain a one-to-one correspondence between the transitions of the two LTSSs underlying the EB$^3$ and LNT descriptions, and hence to preserve strong bisimulation. The presence of array types and of usual programming language constructs (e.g., loops and conditionals) in LNT was also helpful for specifying the memory, the Kleene star-closure operators, and the EB$^3$ guarded expressions containing attribute function calls. At last,