ABSTRACT

Web services are very dynamic, they are all around us and we use them every day without even knowing it. In this paper, the authors define a formal model for dynamic Web services composition and they investigate how it can be used to specify and analyse backward recovery procedures, updating partner services in case of failure. First, they propose Recursive and Dynamic Timed Automata (RDTA) model, interpreted over composite service configurations, which provide a natural way to design stateful and dynamic Web services. Then, the authors define the concurrent semantics of this timed automata extension in terms of real-time rewrite theories. Analysis of the model is carried out in the Real-Time Maude system, based on checking relevant properties.

Keywords: Dynamic Web Services, Formal Modelling and Analysis, Real-Time Maude, Timed Automata, Timed Rewriting Logic

1. INTRODUCTION

One of the emerging issues when using Web Services (WSs, for short) is their dynamic composition that serves anywhere, at any time and it is frequently guided by users’ needs and preferences. WSs composed dynamically must properly react to changes and failure. In (Colombo & Pace, 2013) the recovery and adaptation techniques are classified into backward and forward recovery approaches. The backward recovery restores first the safe state and then follows the execution, while in forward recovery approach, one must first correct the erroneous state and then continue the execution.

DOI: 10.4018/ijertcs.2014070103
Ordinary WS composition languages provide only some constructs to implement temporal constraints (Diaz et al., 2005; Kallel et al., 2009), they do not evaluate dynamic composition process. Besides, most of existing approaches attempting to deal with dynamic aspects, propose hybrid models. Indeed, they combine informal or semi-formal models with formal ones, as for instance, those integrating Ontological, Artificial Intelligence and Aspect Oriented techniques in order to discover, classify and compose dynamically WSs (D’Mello et al., 2011). However, modelling and analysing dynamic WSs is a challenging task, it requires suitable constructions to model and simulate failure and recovery actions. Formal methods are a good candidate to provide a guided development process of dynamic and complex WSs. They define mathematical models having a well-defined semantics, they make it possible application of rigorous analysis procedures.

Timed automata are one of the most known formalism designed to specify and analyse WSs based systems. Time constraints in the given model may traverse all its elements (Waez et al., 2013). Thus, timed automata in their basic form have been applied to simulate, validate and check compatibility of time constrained WSs (Diaz et al., 2005; Diaz et al., 2006; Dong et al., 2006; Guermouche & Godart, 2009). However, when dealing with dynamic aspect, the timed automata formalism remains limited because it does not allow specification of real-time systems having evolved configurations, no update of partner timed automata is permitted in case of failures.

In this paper, we propose a more elaborate framework based on timed automata: Recursive and Dynamic Timed Automata (RDTA), to deal with formalization and analysis of dynamic WSs. The proposed timed automata extension considers backward recovery procedures, updating partner automata in case of failure. Thus, we enrich timed automata with partial and recursive call switches to tackle failure and unavailability of responses. Besides, we give some real-time theories to define the semantics of the proposed extension.

The existing recursive extensions of timed automata and their supporting tools have restrictive specification languages, they impose constraints about the clocks number and their passing mode when executing recursive calls. The use of Real-Time Maude language to define RDTA semantics, allows modelling Real-Time systems having advanced data types and new communication models, since that real-time Maude emphasizes ease and expressivity of specification.

Real-time rewriting logic (Ölveczky & Meseguer, 2007b, c, a) is an important extension of rewriting logic, dedicated to specify and analyse real-time systems. Real-time rewriting theories soundly and naturally depict temporal behaviour. Expressiveness of these theories has provided formal communication models and advanced data types for realistic systems.

The main objective of this work is twofold. On the one hand, we define
Joint Validation of Application Models and Multi-Abstraction Network-on-Chip Platforms
www.igi-global.com/chapter/joint-validation-application-models-multi/65608?camid=4v1a

Fighting the Problem of Unsolicited E-Mail Using a Hashcash Proof-of-Work Approach
www.igi-global.com/chapter/fighting-problem-unsolicited-mail-using/6051?camid=4v1a