Towards an Agent Based Approach for Verification of OWL-S Process Models

Alessio Lomuscio and Monika Solanki

Department of Computing, Imperial College London, UK
{a.lomuscio,m.solanki}@imperial.ac.uk

Abstract. In this paper we investigate the transformation of OWL-S process models to ISPL - the system description language for MCMAS, a symbolic model checker for multi agent systems. We take the view that services can be considered as agents and service compositions as multi agent systems. We illustrate how atomic and composite processes in OWL-S can be encoded into ISPL using the proposed transformation rules for a restricted set of data types. As an illustrative example, we use an extended version of the BravoAir process model. We formalise certain interesting properties of the example in temporal-epistemic logic and present results from their verification using MCMAS.

1 Introduction

The verification of web service behaviour and interaction protocols is now an integral aspect of several frameworks providing service oriented solutions to the IT industry. The Increasing complexities that arise during service composition, make offline verification as model checking [4] crucial in successfully implementing and using services. Model checkers typically use specialised formats for the specification of behaviour, different from those commonly used for describing services. Examples of such system description languages include Promela, used with the checker SPIN [9] and NuSMV, used with the checker NuSMV [3]. However in the web service domain, WSBPEL [16], WSCDL [15] and OWL-S [17], are some popular and widely used standards for describing service behaviour, their composition and interaction protocols.

The languages above work at different levels of abstraction. In order to verify services, an important first step is to investigate how the input language to the model checker can be adapted to encode a suitable abstraction of the service behaviour, which has been described using one of the above standards. Figure 1 illustrates the general architecture of a verification framework for services. As highlighted, a crucial component is the "compiler" that takes as input the service specification, and generates a suitably abstracted model/program, encoded in the system description language for the checker. The model and the desired properties to be verified are fed to the checker. By performing a systematic exploration of the complete set of states that can be generated during an interaction between a service and its clients, the model checker is able to verify desirable properties of the composition. Generating a well abstracted model is thus crucial to the verification of services. However, developing a tool that generates such

L. Aroyo et al. (Eds.): ESWC 2009, LNCS 5554, pp. 578–592, 2009.