PlanICS 2.0 - A Tool for Composing Services*

Artur Niewiadomski[1] and Wojciech Penczek[1,2]

[1] ICS, UPH, Siedlce, Poland, artur.niewiadomski@uph.edu.pl
[2] ICS PAS, Warsaw, Poland, wpenczek@gmail.com

Abstract. This poster reports on the current state of the PlanICS toolset, which aims at solving the Web service composition problem by dividing it into several stages. These include an abstract planning, an offer collecting, and a concrete planning.

Keywords: Web Service Composition, multi-phase Planning, SMT, GA

1 Introduction

A Web Service Composition is a hot topic of many theoretical and practical approaches. It is so deeply investigated since typically a simple Web service does not need to satisfy a user objective. Moreover, due to a support of automatic tools the user is exempted from a manual preparation of execution plans, matching services to each other, and choosing optimal providers for all components. In this poster, we report on the current state of the Web service composition system PlanICS [1]. We describe the general idea behind the system and its modules as well as the work in progress together with some future work directions.

2 PlanICS

PlanICS makes use of a uniform semantic description of services and service types as a part of the *ontology*, which contains also the objects processed by the services. The user query is expressed in a fully declarative language defined over terms from the ontology. The user describes two object sets, called the *initial* and the *expected world*. The task of PlanICS consists in finding a way of transforming the initial world into a superset of the expected one using service types available in the ontology and matching them later with real-world services.

The general system architecture is shown in Figure 1. PlanICS divides the composition process into several stages. The first phase, called the *abstract planning*, deals only with the service types of the ontology. So far, we have implemented two abstract planners: the SMT-based one [3] and the other based on Genetic Algorithms (GA) [8]. Currently, we investigate hybrid algorithms combining SMT with GA, and we work on a translation of the abstract planning to a task for tools dealing with Petri nets, like LoLA [7]. Moreover, we work

* This work has been supported by the National Science Centre under the grant No. 2011/01/B/ST6/01477.

on extending abstract planning to its temporal [4] and parametric version. The abstract planners find multisets of service types that potentially satisfy a *user query*. Still, such a multiset can be viewed as the union of finer equivalence classes defined by partial orders that are identified by the Multiset Explorer module [2].

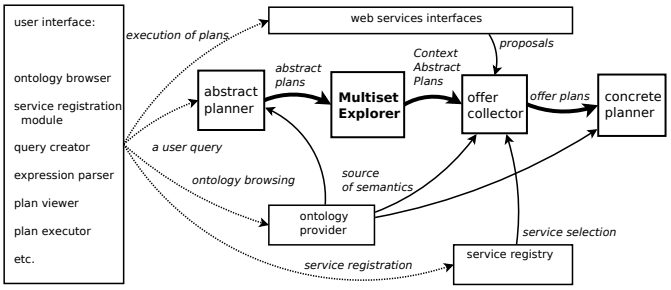

Fig. 1. PlanICS architecture overview.

The second planning stage is performed by the Offer Collector (OC) module, which, in cooperation with service registry, communicates with Web services collecting data to replace the abstract attribute values computed in the first planning phase. Moreover, OC is also to build a set of constraints over offers corresponding to the dependencies from the abstract plan, and resulting from the user query. Then, concrete planners (CPs) get into action. Their task is to prepare a concrete plan by choosing one offer from each set in such a way that all the constraints are satisfied, and the quality function (a part of the user query) is maximized. We provide implementations of CPs based on SMT and GA [5], as well as the hybrid one [6] combining the power of both the methods.

References

1. D. Doliwa et al. PlanICS - a Web Service Compositon Toolset. *Fundam. Inform.*, 112(1):47–71, 2011.
2. L. Mikulski et al. Generating CA-Plans from Multisets of Services. In *PNSE, this volume*, 2014.
3. A. Niewiadomski and W. Penczek. Towards SMT-based Abstract Planning in PlanICS Ontology. In *KEOD*, pages 123–131, 2013.
4. A. Niewiadomski and W. Penczek. SMT-based Abstract Temporal Planning. In *PNSE, this volume*, 2014.
5. A. Niewiadomski, W. Penczek, and J. Skaruz. SMT vs Genetic Algorithms: Concrete Planning in PlanICS Framework. In *CS&P*, pages 309–321, 2013.
6. A. Niewiadomski, W. Penczek, and J. Skaruz. Genetic Algorithm to the Power of SMT: a Hybrid Approach to Web Service Composition Problem. In *Service Computation*, pages 44–48, 2014.
7. A. Niewiadomski and K. Wolf. LoLA as Abstract Planning Engine of PlanICS. In *PNSE, this volume*, 2014.
8. J. Skaruz, A. Niewiadomski, and W. Penczek. Evolutionary Algorithms for Abstract Planning. In *PPAM (1)*, volume 8384 of *LNCS*, pages 392–401. Springer, 2013.