MAD SWAN: A Semantic Web Service Composition System

George Markou and Ioannis Refanidis

Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece
{gmarkou, yrefanid}@uom.gr

Abstract. This paper describes our work towards the implementation of **MAD SWAN**, an open-source web-based application that comprises a web service registry, an XML editor, as well as manual and automatic web service composition modules. **MAD SWAN** supports various stages of web service composition and tackles the inherent non-determinism of the domain.

Keywords: Web Service; PPDDL; non-determinism; OWL-S; composition

1 Proposed Approach

MAD SWAN is an online application related to the composition of Web Services (WSC). The users of **MAD SWAN** (an anagram of "**M**anual **AND A**utomatic **S**emantic **W**sc") are currently able to advertise a new web service (WS) in a registry, retrieve and edit the ones already in it, and manually create workflows based on OWL-S control constructs that can be bound to already stored WS descriptions. The ultimate goal is to be able to generate a composite process model automatically, based on contingency planning and a prior translation of the original WSC domain, described in OWL-S, to the widely adopted AI planning language PPDDL [1]. The whole WSC process will be evaluated against pre-defined use case scenarios and quantitative criteria, as well as a comparison between the automatic and semi-automatic modules.

Our work is motivated by the fact that very few WSC systems tackle the problem of non-determinism, whereas the recent literature suggests a gap in their evaluation. To solve the WSC problem in a non-deterministic environment, we adopt a complete search algorithm that exhaustively generates solution plans for the various contingency combinations, until a time limit set by the users is reached. A, possibly suboptimal, contingency plan can then be constructed by linking these plans through searching for natural join points. To tackle the latter problem, we present an evaluation approach, which can be used and reproduced by other systems, comprising of detailed use case scenarios that are based on an existing, open test collection.

MAD SWAN abides by the main goal of WSs, that is, maximizing the reuse of loosely coupled components; for our implementation we make use of customized versions of already freely available components, such as iServe [2] and PetalsBPM [3]; the former is used as the basis for an online registry, and the latter is modified in order to be used for the manual composition of WSs (instead of BPMN diagrams). Furthermore, **MAD SWAN** follows the de facto WS and planning standards, i.e., OWL-

S and (P)PDDL. The system's lifecycle can be summarized as follows: OWL-S TC [4] v.4 is used throughout all the WSC stages, that is, the registry contains all of its WS descriptions, which the user can search for, edit and use for the creation of a composite WS. For that purpose, the WSs comprising the registry, or a subset of those, are translated to PPDDL. The solution to it is produced by the contingent planner, and it is translated to an OWL-S description.

We designed three use case scenarios, each with an increasing amount of non-determinism and complexity. All the WSs used are taken from OWL-S TC, with minor modifications in some of the descriptions. As a result, the evaluation process presented here can be used by a variety of WSC approaches, as a common test bed. The display format of the resulting workflow for one of the scenarios is shown in Figure 1b. A detailed description of the use case scenarios, and an analysis of the modules used in **MAD SWAN** can be found in [5]. Figure 1a presents part of the application, specifically the registry (on top), along with the XML editor.

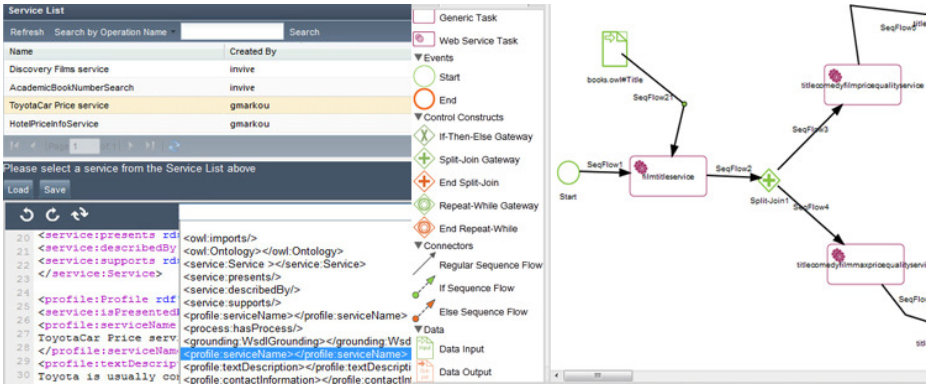

Fig. 1a. Application's interface and editor **Fig. 1b.** Manual WSC module and sample workflow

The full size images of Figure 1 are available in [6], along with the output workflows for all the use case scenarios, a graphical overview of the presented system, and an example of the planning algorithm.

References

1. Younes, H., Littman, M.: PPDDL1.0: An extension to PDDL for expressing planning domains with probabilistic effects. Technical Report, Carnegie Mellon University (2004)
2. Pedrinaci, C., Liu, D., Maleshkova, M. et al.: iServe: a Linked Services Publishing Platform. In: ORES '10. Hersonissos, Crete, Greece (2010)
3. Petals BPM architecture overview, http://goo.gl/647UF
4. OWL-S Service Retrieval Test Collection, http://goo.gl/ehmCG
5. Markou, G., Refanidis, I.: Towards an Automatic Non-Deterministic Web Service Composition Platform. In: NWeSP '12, Sao Carlos, Brazil (2012)
6. MAD SWAN - ESWC 2013 Images, http://goo.gl/ZzEOG