

Towards Automatic Non-Deterministic Web Service Composition

George Markou

Dept of Applied Informatics
University of Macedonia
Thessaloniki, Greece
e-mail: gmarkou@uom.gr

Ioannis Refanidis

Dept of Applied Informatics
University of Macedonia
Thessaloniki, Greece
e-mail: yrefanid@uom.gr

Abstract— This paper describes our ongoing work towards the implementation of an online Web Service Composition system, based on the most prevalent Web Service standards and utilizing other open source projects as sub-elements. The system will treat non-determinism as an inherent characteristic of the problem and will tackle it by exploiting AI planning technology, specifically contingency planning. The paper also presents three detailed use case scenarios to evaluate the system's capabilities. The final system will be the first online application of its kind able to support various stages of Web Service Composition.

Keywords— *Web Service; NuPDDL; non-determinism; OWL-S; composition*

I. INTRODUCTION

As Web Services (WSs) exist and operate in an ever-changing and expanding environment, it is difficult to expect from a human user, or even an expert, to manually or semi-automatically complete the goal of a Web Service Composition (WSC) process. The number of WSs is growing continuously and, as such, the WSs discovery phase becomes more difficult. Web Services can change interfaces or even part of their usage multiple times throughout their lifespan; even if they remain static, there is always the possibility that their execution is not successful. A WSC process should automatically detect and respond to such changes in a way that a human will probably not be able to.

This paper presents our ongoing work towards a WSC system that will exhibit the following functionalities:

- Advertisement of a new WS in a registry, as well as online editing and retrieval of the WSs already stored.
- Translation between the language used to describe the semantic WSs taking part in the composition and an Artificial Intelligence (AI) planning one, OWL-S and NuPDDL [1], respectively.
- Generation of a composition process model, based on contingency planning and OWL-S' control constructs.
- Evaluation of the WSC process, based on quantitative criteria (e.g., the number of WSs considered for the composition, the transformation time of the WSC domain to a planning one, or the total planning time) and pre-defined use case scenarios.

To our knowledge, no other open source web-based WSC system exists. Our system's implementation is based on existing freely available components so as to insure

maximum conformance to the current WS standards and facilitate its quantitative evaluation and comparison to other WSC systems. So far, the advertisement functionality has already been integrated in our online application; currently, we are working on the implementation of a manual WSC module, as well as the complete mapping between the elements of OWL-S and NuPDDL.

The remainder of the paper is organized as follows: Section II reviews related work, while Section III presents the approach that is being followed to implement the various WSC functionalities of the proposed system. Section IV focuses on its evaluation; more specifically, it presents three scenarios that can be used as test cases for our system. Finally, Section V concludes the paper and poses directions for future work.

II. RELATED WORK

Several approaches that convert the original WSC problem to a planning one have been proposed; perhaps the most notable is [2], in which the available WSs' OWL-S process models are translated to a SHOP2 domain, and the WSC problem to a compatible HTN planning problem by describing it as a composite process that can be decomposed to simpler ones (with simple processes being atomic WSs). Then, the planner generates a plan that can be converted back to an OWL-S process and executed by OWL-S API.

SHOP2 plans for tasks in the same order in which they will be executed, which allows it to be aware of the current state of the world at each step. In that way it can gain significant reasoning power in regard to its precondition-evaluation mechanism and reduce the complexity of the planning process. On the other hand, the approach cannot cope with non-determinism in the WSC problem and is planner-dependent, limiting it in comparison to others that translate the WSC problem to a PDDL-compliant one.

An approach very similar to [2] is presented in [3], which couples SHOP2 with an OWL-DL reasoner so as to tackle common problems of HTN methods, such as the inability to associate preferences with possible decompositions. Although the results presented are promising and the planner's performance is adequate, the approach is based on the initial provision of a template, which cannot always be known a priori. Moreover, the authors extend OWL-S with a new process type, thus hindering the use of existing test sets or tools. Finally, the use of a specific planner and internal representation excludes the use of alternative PDDL-

compliant planners that could possibly tackle the problem more efficiently if a standardized presentation was adopted.

An AI planning methodology is also followed in [4], which treats the application of a WS as a belief update operation. Moreover, it identifies two special cases of WSC that are more tractable and allow for a compilation into planning under uncertainty and the subsequent use of an already existing conformant planner (Conformant-FF). Again, though, despite the use of planning techniques and a planner that takes as input PDDL-like problem descriptions, no standardized WS description or planning language is used.

PDDL and OWL-S are, respectively, the de facto planning language and the most widely used semantic description language. Moreover, the latter has been heavily influenced from planning languages, such as PDDL, and for that reason, a (perhaps partial) mapping from OWL-S to PDDL is relatively natural and intuitive. As such, there are several attempts that utilize them together in WSC problems.

OWLS-Xplan [5] incorporates a conversion tool that translates OWL-S descriptions to corresponding PDDL 2.1 ones; this translation, though, does not output a standard PDDL file, but a modified version of it in XML, which, according to the authors, simplifies parsing, reading, and communicating PDDL descriptions using SOAP. Then, Xplan, a hybrid planner that combines guided local search with graph planning and a simple form of HTN decomposition, is called to solve the planning problem. A similar approach is adopted in [6], which, however, can use two alternative external PDDL planners to obtain a solution to the WSC problem.

A three step process for the solution of WSC problems is presented in [7], which involves the translation of OWL-S descriptions and OWL ontologies to a PDDL domain and problem description, the solution of the planning problem through a planner, and the translation of the PDDL plan back to a composite OWL-S WS. Nevertheless, as the authors note, the focus of the paper is only on the first step, and the work presented is basically exploratory.

A novelty of our system is the fact that it will be open source and based on a publicly available online application. To our knowledge, there are currently no web-based systems supporting multiples phases of the WSC process available. YaWSA [8] was the only WSC system in the literature that allowed users to compose services from a web-based interface. However, it was simplistic and only implemented a WSC process, without offering a registry, or the ability to view and edit the WSs' descriptions online. Moreover, at the time of writing it was no longer available for public use.

A prototype web-based WSC system is described in [9], supporting WS browsing, the creation of composite services, service flow execution, and the generation of OWL-S descriptions used for describing their common process pattern instances. These instances are meant to bridge the gap between the users' requirements and the technical service descriptions, as the authors consider OWL-S to be insufficient and not abstract enough to achieve such a result on its own. However, a public link to a running demo of their implementation is not provided.

Finally, it should be noted that the recent bibliography [10] suggests a gap in the evaluation process of the current WSC systems. Not only is there no standard WS test set [4], but most approaches, especially the ones related to planning based techniques, simply evaluate their methodology on a single case study, without referring to quantitative criteria [11, 12, 13]. Only recently, however, a few approaches, such as [5, 6, 14], provided notable exceptions to this rule.

The most extensive evaluation results are provided in [4], which analyzes two artificial benchmarks with different encoding methods and planners, and measures the total runtime of the planner, as well as the number of search states and actions in the output plans. In [6] a single case study is presented, with a different number of WSs participating in the WSC experiments, and measuring the preprocessing, transformation (from OWL-S to PDDL) and planning time required. However, the use of only one planner is referenced, despite the possible use of two different ones, and the atomic WSs that comprise the composite one seem to be (mostly) hand-tailored by the authors, although entire domains of the OWL-S Service Retrieval Test Collection (OWL-S TC) [15] are used for the composition in general.

Finally, Kona et al. [14] present three detailed versions of a single use case scenario, each suited for a mode of their WSC algorithm, along with the IOPEs of the services that take part in the solution of the problem. The test collection used is also mentioned; a modified version of the 2006 WS-Challenge made to fit the authors' framework of choice, as well as various quantitative results regarding the experiments; the number of WSs participating in the WSC, the number of I/O parameters each WS had, and the preprocessing and query execution time needed to obtain a solution. In contrast to [4] and [6], however, the authors of [14] do not provide details regarding the machine that was used to run the experiments.

III. PROPOSED APPROACH

WSs' technologies are based on the idea of maximizing the reuse of loosely coupled components. As such, our view is that the systems implementing WS' functionalities should also be created with the same approach in mind and incorporate already freely available components as their sub-elements. Apart from the additional effort required to create a new component from scratch, such approaches have led to an abundance of applications and standards that only slightly differ from each other, while making the quantitative comparison of different systems difficult; this fact was illustrated in the previous section, and also demonstrated by various surveys relying only on qualitative criteria to review the available methodologies [10].

Our system supports various functionalities relating to different stages of WSC; the first one is the ability to store the service descriptions that will be used later in the discovery of suitable WSs in a registry. The core of the application is based on iServe [16], an open platform for publishing and discovering services. Specifically, we make use of its web-based application that allows users to browse, query and upload services, which, in our case, are semantically described in OWL-S. We have added an XML

editor to the application, made several improvements to its interface and functionality, and populated the registry with version 4.0 of the OWL-S TC.

As aforementioned, the planning module will use NuPDDL for its purposes, as it is compatible with PDDL2.1, retaining most of it, including the handling of functions, conditional effects, and quantifiers; it is also capable of modeling non-deterministic action effects through the introduction of new keywords, such as *oneof* and *unknown*. Since the WSs in the registry are described semantically through OWL-S, a translation between the two languages must take place; we adopted an approach similar to [5, 6, 17] that imply that this conversion is straightforward, at least partially, and present their own mapping.

In [5], the OWL-S' ServiceProfile input parameters are converted to identically named ones of a PDDL action, and the *hasPrecondition* and *hasEffect* parameters to the precondition and effect of the action respectively; in [6], a similar approach is followed, with SWRL used to model the WSs' preconditions and positive effects, and RuleML to model their delete effects. However, [17] and [5] note the problematic conversion of non-physical knowledge from OWL-S inputs and outputs to PDDL. Both tackle the problem by introducing a new predicate in the PDDL domain; the first creates a predicate *agentKnows* with one argument that can either be bound to an input or an output parameter, while the second adds every output variable *X* to the world state through the introduction of an add-effect predicate *agentHasKnowledgeAbout(X)* (the same process is followed in an analogous manner for every input parameter).

After the translation, AI planning techniques can be used to generate the output plan/composite WS. We opt for the incorporation of a contingent planner, so as to generate plans that can cope with the most influential and likely contingencies, as composite WSs may fail to execute correctly for various reasons, such as the unavailability of an atomic WS involved in the plan, or simply because the output of their successful execution is not the expected one.

Our goal is not to develop a plan for every possible contingency, as the WSC domain has too many sources of uncertainty for such an approach to succeed. Instead, similarly to [18], we will produce a seed plan, examine it to determine significant or likely points of failure, and add a conditional branch to recover the plan's execution; this process will be repeated until we either reach a plateau or run

out of time. As we cannot cope with every possible point of failure, a re-planning module will also be incorporated.

Finally, we will convert the NuPDDL plan back to an OWL-S (composite) WS, that is, create an OWL-S profile and its process description, without, however, providing a corresponding WSDL definition, in a fashion similar to that described in [14, 19]. In short, the profile description of the new composite WS will treat it as an atomic service with IOPEs, while the process model will be based on OWL-S control constructs that describe the way the WSs that compose the composite one interact with each other. The OWL-S API [20] that will be used to implement the conversion supports composite processes that use OWL-S control constructs, such as *Split-Join*, and conditional constructs like *IfThenElse*, which will be necessary to produce correct solutions to the use cases presented in the next section. Figure 1 illustrates our approach.

Our work is still in progress and the development of the web-based application is still in alpha version; as such it is not yet available publicly to users. However, a link to its current source code is available in [21].

IV. EVALUATION OF RESULTS

As aforementioned, there is currently no standard WS test bed, concerning both the scenarios used to test the WSC process, and the WSs that take part in it. However, the recent trend of widespread use of OWL-S TC, as a test bed in the recent S3 contests [22], or in the recent literature [2, 5, 7], suggests its suitability for use in our evaluation experiments.

We believe that it is beneficiary to define specific use case scenarios in detail, as well as provide the actual WSs' descriptions that will be used. As such, we have designed three use case scenarios, each based on the WSs contained in a domain of OWL-S TC, and with an increasing amount of non-determinism and complexity than the previous one. In order to design useful test cases for our system, we made several minor modifications to the available WSs' descriptions and their relative ontologies, and also added a few descriptions to the collection, albeit similar to the ones already included in it. A full description of the use cases and the WSs they are based on can be found in [23].

The first use case is fully deterministic, allowing for the output of a fully serialized composite WS; it refers to a user who knows part of a movie title and wants to retrieve all the comedy films that exist with a similar title, along with their

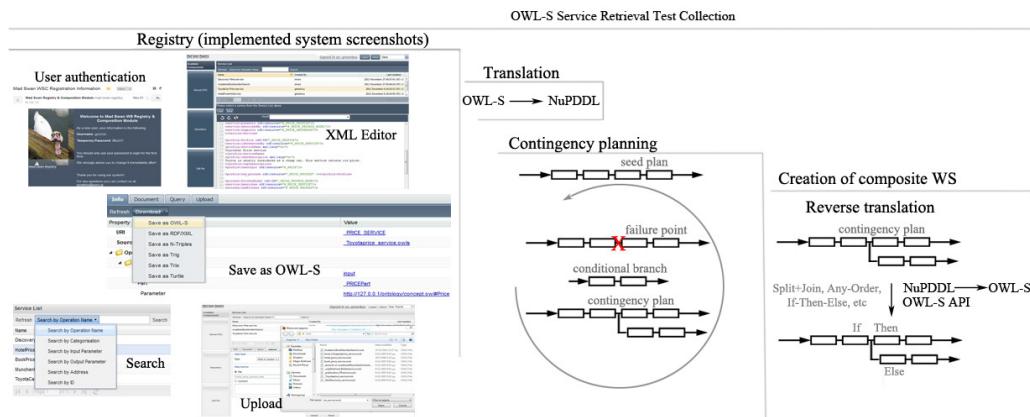


Figure 1. System overview (using a modified figure from [18]).

pricing information. The other two scenarios feature non-deterministic elements, such as preferences between types of products, or cases where a WS may have different outcomes. Particularly, the second one refers to an online bookstore user who wants to purchase a book with a preferred method of payment (a cheque, debit or credit card), with the WS having different outcomes depending on whether the book is in stock at the store or not. If it is available, the composite WS should add the book to the user's shopping cart, purchase it with the specified method of payment, and output information regarding it, such as its author. If, however, it is not in stock, no payment should be made, and no further information concerning it should be displayed to the client.

The third use case concerns the purchase of a camera; the user has a preference towards an analog SLR model, but is willing to settle for other ones if that one is not in stock. Apart from the addition of preferences, this scenario differs from the second one in that more than one sellers are assumed to exist, and the composite WS should check with all of them to determine if the item is in stock. As such, if a store is found that sells the analog SLR model and has it in stock, it should be added to the user's shopping cart. If it is not in stock, the search should continue for another store that sells it, and if one cannot be found, the process should be repeated, this time searching for the camera's compact version, or, if all else fails, for any camera available in stock.

Although the first two scenarios can be considered as special cases of the last one, it is important to showcase that the system can indeed cope with the generation of both sequential and conditional plans, with and without preferences. Moreover, the importance of the scenarios lies in that they exhibit that this particular test set can be used to produce meaningful use cases that can evaluate the capabilities of WSC approaches efficiently and in a manner that is reproducible and extensible.

V. CONCLUSION AND FUTURE WORK

In this paper, we presented our ongoing work towards the implementation of an online WSC system that makes use of non-deterministic AI planning techniques and of already freely available WS-related components. Furthermore, we described in detail three use case scenarios that will be used to evaluate such systems, based on an existing WSs' test collection.

The fact that the final system will support various stages of WSC, as well as that it will be online and open source, is important, as at the moment, there is no other system with similar capabilities. Moreover, the scenarios enable us to test whether the proposed system can cope with the demands of WSC efficiently. In addition to this fact, it is our hope that they can be used by other WSC projects as a common test bed, since they provide detailed descriptions of the WSs involved, as well as their intended goals. Furthermore, they can be used by systems supporting either deterministic or non-deterministic planning.

We expect that in the near future we will be able to demonstrate the first results of this effort through a publicly available online prototype.

ACKNOWLEDGMENT

This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program “Education and Lifelong Learning” of the National Strategic Reference Framework (NSRF) - Research Funding Program: Heracleitus II. Investing in knowledge society through the European Social Fund.

REFERENCES

- [1] NuPDDL: Non-determinism and more in PDDL, <http://mbp.fbk.eu/NuPDDL.html> 03/08/2012
- [2] E. Sirin, B. Parsia, D. Wu, J. Hendler, and D. Nau, “HTN planning for web service composition using SHOP2”, *J. Web Semant.*, vol. 1, no. 4, Oct. 2004, pp. 377-396.
- [3] E. Sirin, B. Parsia, and J. Hendler, “Template-based composition of semantic web services”, Proc. *1st International AAAI Fall Symposium on Agents and the Semantic Web*, Nov. 2005.
- [4] J. Hoffmann, P. Bertoli, M. Helmert, and M. Pistore, “Message-based web service composition, integrity constraints, and planning under uncertainty: a new connection”, *J. Artif. Intell. Res.*, vol. 35, May 2009, pp.49-117.
- [5] M. Klusch, A. Gerber, and M. Schmidt, “Semantic web service composition planning with OWLS-Xplan”, Proc. *1st International AAAI Fall Symposium on Agents and the Semantic Web*, Nov. 2005.
- [6] O. Hatzi, D. Vrakas, M. Nikolaidou, et al., “An integrated approach to automated semantic web service composition through planning”, *IEEE Trans. Service Computing*, April 2011, pp. 301-308.
- [7] Y. Bo and Q. Zheng, “A method of semantic web service composition based PDDL”, Proc. *IEEE International Conference on Service-Oriented Computing and Applications (SOCA '09)*, Dec. 2009, pp. 1-4.
- [8] A. Macdonald, “Service composition with hyper-programming”. Technical Report, University of St Andrews, 2007. http://www.cs.st-andrews.ac.uk/~angus/docs/yawsa/final_report.pdf 03/09/2012
- [9] X. Du, W. Song, and M. Munro, “Using common process patterns for semantic web services composition”, Proc. *15th International Conference on Information Systems Development (ISD'06)*, Sept. 2006.
- [10] M. Chan, J. Bishop, and L. Baresi, “Survey and comparison of planning techniques for web services composition”. Technical Report. University of Pretoria. 2007. <http://polelo.cs.up.ac.za/papers/Chan-Bishop-Baresi.pdf> 03/11/2012
- [11] K. Chen, J. Xu, and S. Reiff-Marganiec, “Markov-HTN planning approach to enhance flexibility of automatic web service composition”, Proc. *IEEE International Conference on Web Services (ICWS'09)*, July 2009, pp. 9-16.
- [12] D.V. McDermott, “Estimated-regression planning for interactions with web services”, Proc. *Sixth International Conference on Artificial Intelligence Planning Systems (AIPS '02)*, April 2002, pp. 204-211.
- [13] S. McIlraith and T. Son, “Adapting Golog for composition of semantic web services”, Proc. *8th International Conference on Principles of Knowledge Representation and Reasoning (KR2002)*, April 2002, pp. 482-496.
- [14] S. Kona, A. Bansal, M.B. Blake, and G. Gupta, “Generalized Semantics-Based Service Composition”, Proc. *IEEE International Conference on Web Services (ICWS'08)*, Sept. 2008.

- [15] SemWebCentral OWL-S Service Retrieval Test Collection, http://semwebcentral.org/frs/?group_id=89 02/15/2012
- [16] C. Pedrinaci, D. Liu, M. Maleshkova, et al., “iServe: a linked services publishing platform”, Proc. *Ontology Repositories and Editors for the Semantic Web Workshop at the 7th Extended Semantic Web Conference (ORES '10)*, May 2010.
- [17] P. Birchmeier, “Semi-automated semantic web service composition planner supporting alternative plans synthesis and imprecise planning”. Diploma Thesis, University of Zurich, 2007. http://www.ifi.uzh.ch/pax/uploads/pdf/publication/1691/Birchmeier_Peter.pdf 03/12/2012
- [18] R. Dearden, N. Meuleau, S. Ramakrishnan, D. Smith, and R. Washington, “Contingency planning for planetary rovers”, Proc. *3rd International NASA Workshop on Planning & Scheduling for Space (IW PSS '02)*, Oct. 2002.
- [19] E. Ziaka, D. Vrakas, and N. Bassiliades, “Translating web services composition plans to OWL-S descriptions”, Proc. *3rd International Conference on Agents and Artificial Intelligence (ICAART '11)*, Jan. 2011, pp. 167-176.
- [20] OWL-S API Introduction, <http://on.cs.unibas.ch/owls-api/index.html> 03/02/2012
- [21] Application’s source code, <http://ai.uom.gr/gmarkou/Files/MadSwanSourceCode/> 03/11/2012
- [22] S3 Contest: Retrieval Performance Evaluation of Matchmakers for Semantic Web Services, <http://www-ags.dfki.uni-sb.de/~klusck/s3/html/2011.html> 01/13/2012
- [23] Markou, G.: “*Heracleitus II - WSC Use Case Scenarios*”. Technical report. University of Macedonia, 2012. Available at http://ai.uom.gr/gmarkou/Files/Mad_Swan_Use_Case_Scenarios.pdf 04/26/2012