Semantic Web Service Composition using Planning and Ontology Concept Relevance

Ourania Hatzi[1] , Georgios Meditskos[2], Dimitris Vrakas[2], Nick Bassiliades[2],
Dimosthenis Anagnostopoulos[1] and Ioannis Vlahavas[2]

[1]Department of Informatics and Telematics,
Harokopio University of Athens
Kallithea, Athens, Greece
{raniah, dimosthe}@hua.gr

[2]Department of Informatics,
Aristotle University of Thessaloniki,
Thessaloniki, Greece
{gmeditsk, dvrakas, nbassili, vlahavas}@csd.auth.gr

Abstract: This paper presents PORSCE II, a system that combines planning and ontology concept relevance for automatically composing semantic web services. The presented approach includes transformation of the web service composition problem into a planning problem, enhancement with semantic awareness and relaxation and solution through external planners. The produced plans are visualized and their accuracy is assessed.

Keywords: Automated Web Service Composition, AI Planning, Ontology Concept Relevance, Semantic Awareness, Semantic Relaxation.

I. INTRODUCTION

Web services offer a well-defined interface which can deal with the major problem of interoperability on the web. However, atomic service functionality is often not adequate and web service composition, which results in enhanced functionality, is essential. Automation of the composition process is preferred, as the number of available web services is expected to continuously increase over time.

In this paper, the automated web service composition paradigm is approached as a planning problem, enhanced with semantic awareness, over the OWL-S [6] profile descriptions of semantic web services. More specifically, PORSCE II is presented; a combination of a domain-independent planning system, a concept relevance module for discovering similarities among OWL ontology concepts, and a visual component. PORSCE II transforms the web service composition problem into a planning problem, performs semantic relaxation if required, obtains solutions utilizing external planning systems, and estimates the complex service accuracy.

The rest of the paper is organized as follows: Section 2 discusses some related work, Section 3 outlines the system architecture and Sections 4 and 5 elaborate on the core elements of the system. Section 6 presents a case study and evaluation. Finally, Section 7 concludes the paper and poses future directions.

II. RELATED WORK

This section mentions some prominent approaches for automatic web service composition via planning.

SHOP-2 [2] uses DAML-S service descriptions and performs HTN planning to obtain a solution. The disadvantage of this approach lies in the fact that the planning process, due to its hierarchical nature, requires given decomposition rules, which have to be encoded in advance using domain-specific knowledge.

OWLS-Xplan [3] uses OWL-S descriptions of web services to derive planning domains and problems, and invokes the Xplan planner to generate complex services. The system is PDDL compliant; however, semantic information provided from domain ontologies is not fully utilized; therefore, the planning module requires exact matching for service inputs and outputs.

The architecture presented in [11] employs planning based on Event Calculus to automatically generate workflows for web service composition.

The main advantages of PORSCE II, compared to the aforementioned approaches, lie in the fact that the OWL-S descriptions of the web services along with the corresponding ontologies are adequate information for the system to determine how to form valid complex services that satisfy given goals, without any prior or additional knowledge. Furthermore, by performing semantic relaxation, the system can find approximate complex services when no accurate plan exists.

III. SYSTEM OVERVIEW AND ARCHITECTURE

PORSCE II is the evolution of the prototype system PORSCE [7].

The main features of the system include:

- Transformation of OWL-S atomic web service descriptions into planning operators.
- Interaction with the user in order to acquire their preferences regarding the complex service and desired metrics for concept similarity.
- Enhancing the planning domain and problem with equivalent and semantically similar concepts.
- Providing solutions to the problem by invoking external planning systems.
- Assessing the accuracy of the complex services.
- Visualizing the solution and calculated metrics.

PORSCE II comprises of four subcomponents, namely the OWL-S Parser, the Transformation Component, the OWL Ontology Manager and the Visualizer. The system architecture along with the interactions among the components are depicted in Fig. 1.

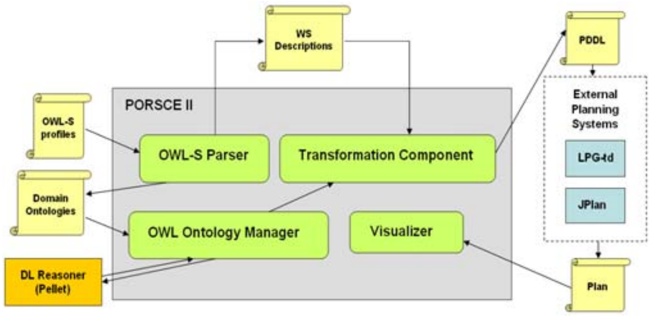

Fig. 1 The architecture of PORSCE II.

The OWL-S Parser parses a set of OWL-S web service profiles and produces the web service descriptions, which are consequently fed to the Transformation Component. Furthermore, it determines the domain ontologies used for organizing the concepts appearing in the OWL-S profiles, which are then forwarded to the OWL Ontology Manager (OOM). The OOM applies the selected algorithm for calculating concept similarity and determines semantically relevant concepts. The Transformation Component is responsible for expressing the problem of web service composition as a planning problem, interacting with the user, enhancing the planning problem with semantic information retrieved from the OOM, cooperating with external planning systems in order to acquire solutions to the problem, and evaluating the accuracy of these solutions. The purpose of the Visualizer is to provide the user with a visual representation of the plan, which is in fact a description of the complex service.

IV. OWL Ontology Manager

OOM enables approximations when an exact input to output matching is not possible, by providing semantically equivalent and relevant concepts. In our approach, two ontology concepts are considered relevant if and only if (a) they have a specific hierarchical relationship, and (b) their semantic distance does not exceed a user-defined threshold.

A. Hierarhical Relationships

Four possible hierarchical relationships exist between two ontology concepts A and B:

- exact(A, B): The two concepts should have the same URI or they should be equivalent, in terms of OWL class equivalence, i.e. $A = B \vee A \equiv B$.
- plugin(A, B): The concept A should be subsumed by the concept *B*, i.e. $A \sqsubseteq B$.
- subsume(A, B): The concept A should subsume the concept B, i.e. $B \sqsubseteq A$. In both the plugin and subsume cases the subsumption relationships of equivalent concepts are not considered.
- sibling(A, B): The two concepts have a common superclass T, such as $A \sqsubseteq T \wedge B \sqsubseteq T$.

B. Semantic Distance

Two methods for determining the semantic distance between two ontology concepts are provided:

- The Edge-Counting Distance (ec) is based on the distance of two concepts in terms of the number of edges found on the shortest path between them in the ontology [7]. An edge exists between two concepts A and B if A is the direct subclass of B.
- The Upwards Cotopic Distance [10] is defined in terms of the upwards cotopic measure, denoted as uc(A) that represents the set of the superclasses of the concept A, including A itself. In PORSCE II, this definition has been modified in order to incorporate the semantics of an ontology hierarchy and is calculated as:

$$d_{uc}(A,B) = 1 - \frac{|uc(A) \cap uc(B)| - 1}{|uc(A) \cup uc(B)| - 1}$$

In both cases, the implementation of the distance between two concepts returns a value between 0 and 1, with 1 denoting absolute mismatch.

V. Transformation Component

The Transformation Component is responsible for representing the web service composition problem in planning terms, performing semantic relaxation, interacting with the user to formulate the planning problem, invoking the external planners and evaluating the solution.

A. Problem Representation

A planning problem is a tuple <I,A,G> where I is the initial state, G is a set of goals and set A contains all the actions that can be used to modify states. Each action has three lists of facts containing the preconditions, the facts that are added to the state and the facts that are deleted from the state after its application. The solution to a planning problem (plan) is a sequence of actions, which, if applied to I, lead to a state S' such that $S' \supseteq G$.

In PORSCE II, the inputs that the user wishes to provide to the complex service formulate the initial state of the problem, while the desired outputs of the complex service formulate the goals of the problem. The available OWL-S web service profiles are used in order to obtain the actions available in the planning domain. The preconditions list of each action is formed by the inputs of the service, the add effects are the outputs of the service and the delete list is left empty, since only services with no negative effects are considered. Finally, the description of the desired complex service is the produced plan.

B. Semantic Relaxation

The planning system needs to be aware of possible semantic similarities among syntactically different concepts, in order to be able to match preconditions and effects correctly during the planning process. In order to achieve that, PORSCE II enhances the planning problem with semantic information, maintaining independence from the planner. In a pre-processing phase, the OOM provides all the semantically similar concepts for both the facts of the initial state and the outputs of the available actions. The enhancement of the problem by PORSCE II is based on the following rules:

- The original concepts of the initial state together with the equivalent and semantically similar concepts form a new set of facts noted as the Expanded Initial State (EIS).
- The goals of the problem remain the same.
- The Enhanced Operator Set (EOS) is produced, by including in the effects list of each operator all the equivalent and semantically similar concepts for the concepts in its initial effects list.

The new problem, namely <EIS,EOS,G> is then encoded into PDDL [5] and forwarded to the planner.

C. *Acquiring Solutions*

The planning systems currently incorporated in the system are JPlan [1] and LPG-*td* [8]. Both planners proved to be remarkably fast and can handle a respectable number of operators (currently over 2000), which is very important as the number of available web services is expected to increase significantly over time. After the planning process is completed, the plans, which might be sequential or structured in levels, are visualized and their accuracy is estimated.

D. *Complex Service Evaluation*

Calculated statistics and quality metrics include the number of actions and levels in a plan, and the plan accuracy metric, when semantic relaxation takes place.

For the calculation of the plan semantic distance, each concept appearing in the inputs or outputs of the actions of the plan is annotated by the OOM with a semantic distance di with respect to the original concept it was derived from and a weight wi, associated with the kind of hierarchical relationship to the original concept. Consequently, for the upwards cotopic distance metric, the plan semantic distance is calculated as

$$PSD_{uc} = \prod_{i=0}^{n} w_i d_i , d_i \neq 0$$

while the edge-counting case is similar.

The plan accuracy metric is calculated as 1-*PSD*; therefore, if there is exact input to output matching, or if only equivalent concepts are used, then the plan accuracy metric value is 1, while it decreases as the plan becomes less accurate.

VI. Case Study and System Evaluation

The initial test sets used for experiments were obtained from the OWLS-TC version 2.2 revision 1 [4], and included web service descriptions for various domains. For further experimentation, several service descriptions were modified or added to the domains, accommodating the demonstration of the system capabilities. The scenario presented in this paper concerns the electronic purchase of a book. The user provides a book title and author, credit card info and their address, and requires a charge to credit card for the purchase, the estimated shipping dates and the customs cost for the specific item. The web services that were modified / added to the domain are depicted in Table 1.

TABLE I. The modified / added web services.

Service	Inputs	Outputs
BookToPublisher	Book, Author	Publisher
CreditCardCharge	OrderData, CreditCard	Payment
ElectronicOrder	Electronic	OrderData
PublisherElectronicOrder	PublisherInfo	OrderData
ElectronicOrderInfo	Electronic	OrderInfo
Shipping	Address, OrderData	ShippingDate
WaysOfOrder	Publisher	Electronic
CustomsCost	Publisher, OrderData	CustomsCost

Planning operators are formulated by these available web services. The initial and goal states are produced by the inputs and desired outputs of the complex service, respectively, while the GUI of PORSCE II enables the user to visually designate these inputs and outputs. The exact matching plans produced by JPlan and LPG-td for the specific case study using the operator set described above are presented in Fig. 2.

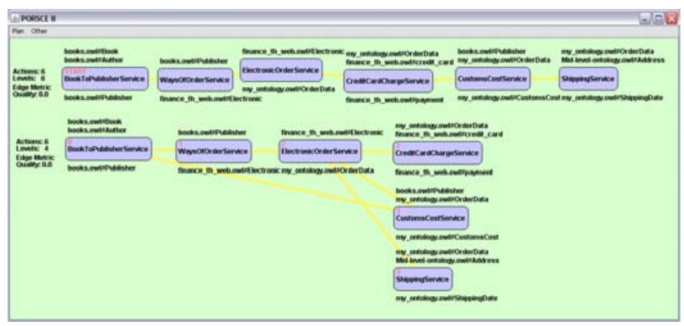

Fig. 2. The plans from JPlan (top) and LPG-td (bottom) for the specific case study.

In cases when no accurate solution can be obtained, PORSCE II is able to present the user with complex services that approximate the required functionality, which are obtained by performing semantically relaxed concept matching (Fig. 3). Note the decreased accuracy depicted by the calculated metric in Fig. 3.

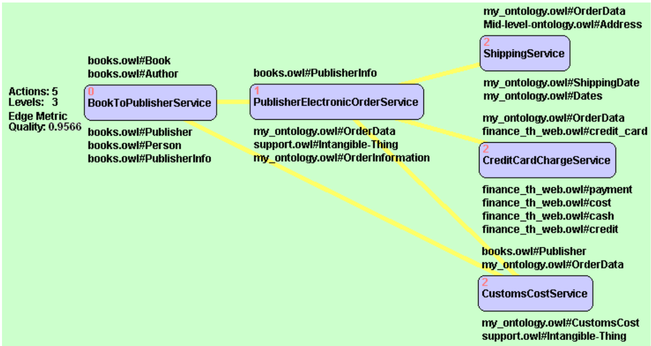

Fig. 3. Approximate complex service.

In order to study the behavior of the system as the number of available web services increases, web services were added to the domain progressively in batches. The time performance results presented in Table 2 were obtained from a number of runs on a machine with Dual-Core AMD Opteron Processor at 2.20GHz with 1GB of RAM memory and concern times for preprocessing, transformation of the OWL-S web service profiles to PDDL operators and planning. Experiments were performed without semantic relaxation (X rows), and with semantic relaxation using either the edge-counting distance metric (E rows) or the upwards cotopic metric (C rows).

TABLE II. TIME MEASUREMENTS IN MILLISECONDS

Number of web services		10	100	500	1000
Preprocessing time		5857	6104	5875	5703
Transformation time per web service	X	459	700	702	792
	E	453	671	757	797
	C	459	746	1457	3901
Planning time (JPlan)	X	31	63	93	172
	E	14	37	125	176
	C	23	22	156	375
Planning time (LPGtd)	X	1	13	16	17
	E	4	6	15	16
	C	3	5	16	16

The preprocessing time did not show significant fluctuation, as it depends only on the number and structure of the processed ontologies and not on the number of available web services. The average transformation time per web service profile converged to approximately 0.8 seconds for the exact matching and the edge-counting distance metric cases; however, in the upwards cotopic metric distance, it increases as available web services increase, due to the higher complexity of the algorithm used for concept relevance. Both planners show an increase in planning time as the number of actions increases, however, LPG-td is proved remarkably faster than JPlan.

VII. Conclusions and Future Work

The work presented in this paper concerns the approach of the automated semantic web service composition problem through planning, enhanced with semantic awareness and relaxation. The approach is supported by the development of the PORSCE II system, which not only transforms the problem but also solves it, visualizes and evaluates the solutions and allows interventions to the complex services.

Future goals include the extension of the system in order to translate the plan describing the complex service into OWL-S for consequent execution and automatic feedback. Furthermore, integration with the VLEPPO system [9] is a promising future direction, in order to accommodate visual design and solving of web service composition problems as planning problems. Finally, it lies in our immediate plans to study ways to enhance the services representation and explore the ability to produce various complex services according to non-functional user preferences, such as Quality of Service (QoS), resource and time constraints.

Acknowledgment

This work was partially supported by a PENED program (EPAN M.8.3.1, No. 03EΔ73), jointly funded by the European Union and the Greek government (General Secretariat of Research and Technology).

References

[1] JPlan, http://sourceforge.net/projects/jplan

[2] E. Sirin, B. Parsia, D. Wu, J. Hendler and D. Nau, 2004. HTN planning for web service composition using SHOP2. Journal of Web Semantics, 1(4) 377–396.

[3] M. Klusch, A. Gerber, M. Schmidt: Semantic Web Service Composition Planning with OWLS-XPlan. AAAI Fall Symposium on Semantic Web and Agents, USA, 2005.

[4] OWLS-TC, www.semwebcentral.org/projects/owls-tc/

[5] M. Ghallab, A. Howe, C. Knoblock, D. McDermott, A. Ram, M. Veloso, D. Weld, D. Wilkins, "PDDL -- the Planning Domain Definition Language". Technical report, Yale University, New Haven, CT (1998).

[6] OWL-S 1.1, http://www.daml.org/services/owl-s/1.1/

[7] O. Hatzi, G. Meditskos, D. Vrakas, N. Bassiliades, D. Anagnostopoulos, I. Vlahavas, A Synergy of Planning and Ontology Concept Ranking for Semantic Web Service Composition, IBERAMIA'08, LNCS 5290 Springer, 42-51

[8] A. Gerevini, A. Saetti, I. Serina, LPG-TD: a Fully Automated Planner for PDDL2.2 Domains",ICAPS-04.

[9] O. Hatzi, D. Vrakas, N. Bassiliades, D. Anagnostopoulos, I. Vlahavas, VLEPpO: A Visual Language for Problem Representation, In Proc. PlanSIG 2007, 60-66.

[10] A. Maedche and V. Zacharias, Clustering Ontology-Based Metadata in the Semantic Web, European Conf. Principles of Data Mining and Knowledge Discovery, 2002.

[11] L. Chen, X. Yang, Applying AI Planning to Semantic Web Services for workflow Generation. In Proc. SKG 2005.