Enhancing Semantic Web Services Composition with User Interaction*

Peter Bartalos and Mária Bieliková
Institute of Informatics and Software Engineering, Faculty of Informatics
and Information Technologies, Slovak University of Technology in Bratislava
Ilkovičova 3, 842 16 Bratislava, Slovakia
{bartalos,bielik}@fiit.stuba.sk

Abstract

The semantic web services composition process arranges several web services into one composite to realize complex workflows. To do this, semantic metadata of web services' description are used. The current approaches based mainly on AI planning are immature to be used in practice. In this paper we propose an approach involving users in the semantic web services composition to help overcome problems occurring in the composition process. The basic idea is to find the users helpful in situations when preconditions are not satisfied or some input data are not available which are in demand to create a composition.

1 Introduction

Semantic web services is a topical research area aimed at exploiting semantic annotation of web service descriptions [2]. One of the most studied topic of this area is the automation of the semantic web service composition trying to arrange more services into one complex to be able to realize more complicated workflows [7]. Composition is studied in different contexts to achieve practical applicability [3, 9].

Several methods used in semantic web services composition based mostly on AI planning techniques were recently proposed. These include state space search, graph based planning, HTN (Hierarchical Task Network) planning, approaches based on logical programming and others [10]. They are usually combined together to bring the best results. The basic approach is to transform the composition problem into state space search task. The state space represents all the possible states of our world where the web services are modeled as actions altering these states. The task is to find a sequence of actions altering the initial state into the goal state, i.e. the search for a path between these two states is performed. To perform the search different space search algorithms are employed such as forward/backward chaining, hill-climbing. Because the state space can be very large, from computational reasons it is usually impossible to find a solution by searching the whole space (brute force approach). Composition tools use different heuristics to speed up the search, for example such one estimating the distance between the current and the goal state.

However, a lot of work has been done in the field of semantic web services composition, there are still several problems needed to be solved, before the automatic composition can be applied in practice. The basic problem is that tools for automatic composition are immature to handle situations when some web services are unavailable or their descriptions are insufficient to find them (although the available web services allow creating a composition able to achieve a goal). Other problem is related to the input data for web services which are not always available. The determination of the values of input parameters that are needed to get the desired results in automatic manner is hard or even impossible in many cases. Usually users – domain experts are required to do this. Even if the mentioned problems are not occurring, the current composition tools are not able to use all the control constructs available to compose complex workflows. This reduces the set of use cases for these tools.

In this paper we deal with an approach involving the users in the semantic web services composition. We propose an approach to find users able to solve problems occurring within automatic semantic web services composition. It exploits semantic metadata of web services descriptions and user models to realize this.

2 Related Work

One way how to solve problems occurring within semantic web services composition is to incorporate users in this process. In this approach the human intelligence is exploited to overcome the difficulties that cannot be solved in

*This work was partially supported by the Slovak Research and Development Agency under the contract No. APVV-0391-06, "Semantic Composition of Web and Grid Services" and by the Scientific Grant Agency of Slovak Republic, grant No. VG1/3102/06.

automatic manner. The idea is not new. In [5] an approach describing a framework for interactive service composition is presented. The tool called *CAT* (Composition Analysis Tool) assists here the users in constructing a computational pathway taking an advantage of semantic description of services. The computational pathway is a set of operations and links connecting operations based on the input and output parameter constraints. The framework helps the user by generating suggestions how to solve the problems occurred within the composition.

The authors argue that: (i) complex applications require user interaction, (ii) partial workflows with high-level descriptions of component services are needed, (iii) constraints shared by abstract types of components need to be checked at every step. The presented web services composition approach complements WSDL descriptions with task and domain ontologies. The domain ontology specifies the data types in WSDL descriptions. The task ontology is used to describe abstract types of operations and services. This way we move from the syntactic to the semantics level. Here the inputs, outputs and the function of the service are described using terms defined in the domain ontology.

Building of the computational pathway starts from a high level description of the problem using ontological terms. The building process is a combination of backward chaining and user interaction. In this iterative process the CAT tool is used to generate suggestions for the user to help solving problems occurred within the composition. Its aim is to achieve a situation, when all expected results are reached, links in pathways are consistent, all the input data are provided, all the operations are grounded (there are actual operations which can be executed). The suggestions include for example finding a web services having outputs compatible with inputs of those services from the actual composition having not defined the source of the input data.

Even though the idea of incorporating users into the service composition process is beneficial is right, the approach presented in [5] does not exploit all the possibilities. The overall approach is poorly usable in complicated situations when large amount of services is composed. One disadvantage is the usage of backward chaining for automatic composition, which is not suitable in complex problems from computational reasons. The authors state that the CAT tool needs semantic descriptions of the services. However, it is not clear how the approach takes the advantage of the available semantics.

3 Approach Overview

An overview of our proposal for web services composition supported by user interaction is depicted in Figure 1. The first step is goal gathering. The result of this step is a formalized goal description. To achieve this goal an existing automatic semantic web services composition approach is applied. During the planning it can happen that no input data is available or some preconditions are not satisfied which are required to execute the web service. These cause that some parts of the workflow are undefined (clouds in the figure) or the potential web services cannot be applied.

Let denote the function determining the truth-value of the preconditions as $value()$, the function determining if data is unavailable as $NA()$ – true if unavailable, the set of preconditions as Pre, the set of input data required for successful composition as $Input$. The mentioned situations then include for example the following: not all preconditions are satisfied ($\exists p \in Pre : value(p) = false$), not all input data are available ($\exists i \in Input : NA(i) = true$).

To recognize a situation when users should be or can be incorporated, sophisticated methods must be developed. The problem is to recognize the situation when users can help to solve the problematic part of the workflow. In the problematic cases human users are involved in the composition process. They may collaborate during it. The two main cases when collaboration is required are: (i) the users work on dependent problems – collaboration is needed to design a complete workflow, (ii) multiple solutions for a given problem exist – the users should collaborate to choose the best one. Based on the collaboration, the missing parts of the workflow are defined. This way, a composite web service representing the complete plan, which is not possible to be composed in fully automatic manner is created. In this, no unsatisfied preconditions exist ($\nexists p \in Pre : value(p) = false$), no input data are unavailable($\nexists i \in Input : NA(i) = true$). If this holds the execution can take place.

4 Extending Automatic Semantic Web Services Composition

Our proposal for developing a web services composition method exploiting user interaction is to extend some existing efficient automatic approach. One of the best candidate seems to be *OWLS-XPlan* [8]. It uses hierarchical task networks and various state space search algorithms to create the plan. To speed up the search, various heuristics are used. These are used to choose the next state which is estimated to be the closest to the goal state. This state is chosen from a set of states reachable from the current state by the application of web services which have satisfied preconditions and have available input data.

Our aim is to extend the set of reachable states by exploiting the users. This extension includes states not reachable originally, but the user can proceed to make them. This includes a provision of input data or changing the world's state in such a way that other services are applicable and thus new states are reachable. If the extension set includes a

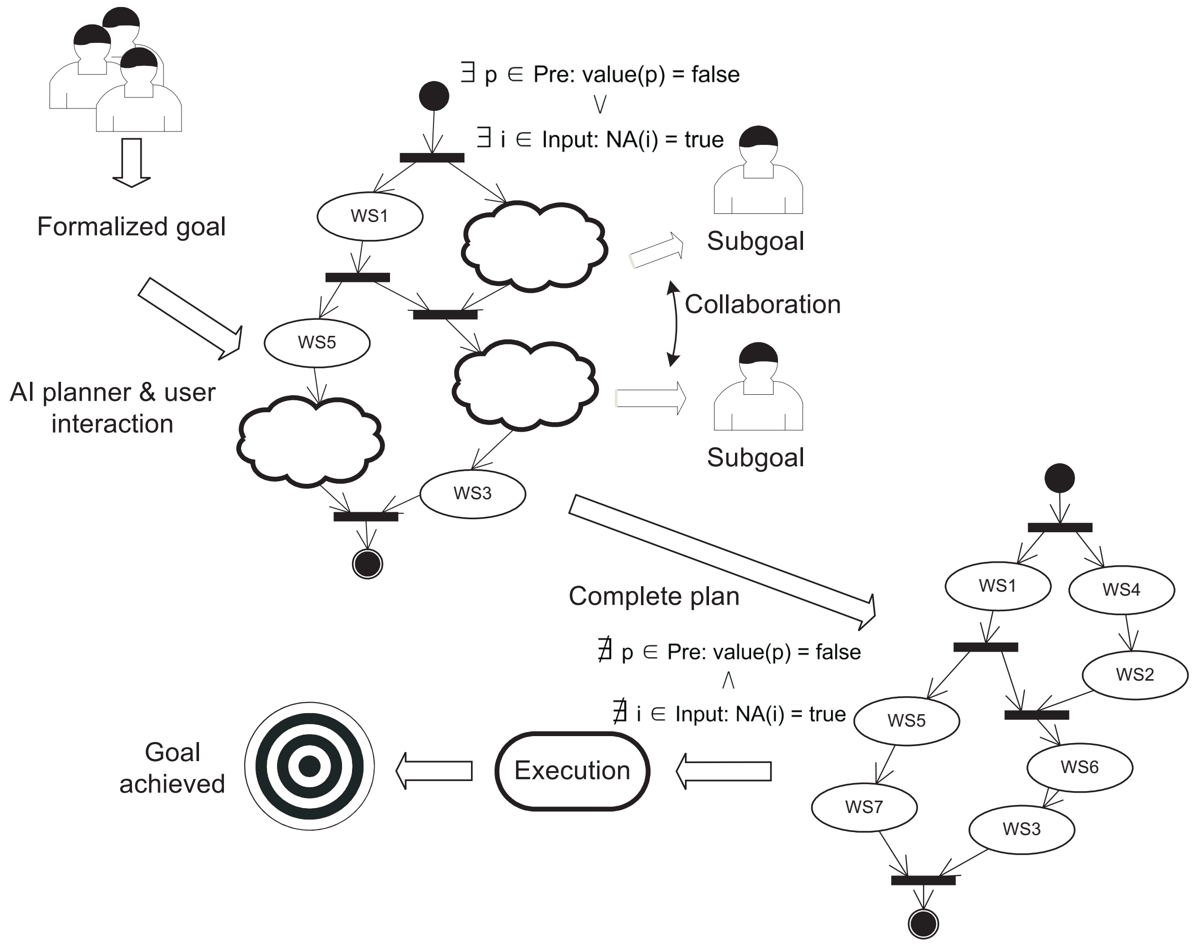

Figure 1. Web service composition supported by users' interaction.

state, which is closer to the goal state than any other state in the original set, we can use it to reach the goal state faster. This way we speed up the web service composition.

The extension set is built based on estimation whether any user can provide useful data or achieve conditions, which are in demand to execute the service. Hence we use semantic web services, the data and conditions are described as concepts in domain ontologies. The estimations are based on the user's knowledge of concepts retrieved from a user model [1]. The user model is used to hold useful information about the user, usually used to personalize his work with an application. In our case it is built in such a way that it contains information about the user's knowledge of concepts describing input data or conditions required to execute a service. If we are not able to estimate the users' knowledge of some concept directly from the user model (the complete user model is rarely available), we use heuristics to estimate it. The heuristics are based on the user model and semantic concept similarity.

Concept similarity in our approach is based on an evaluation of the similarity of two concepts considering the semantic level [4, 6]. The similarity value is denoted by sem. It is normalized to get values from range $< 0, 1 >$ (0 for non similar and 1 for equivalent concepts). Let denote N as a set of neighbor concepts of c. This set contains concepts closely related with concept c where the knowledge level of these concepts by the user is known. Now, we introduce heuristic $know(c, u)$ estimating the knowledge of concept c by the user u. If not already available in the model, it is computed as follows:

$$know(c, u) = \frac{\sum_{\forall n \in N} \{sem(c, n)\ .\ know(n, u)\}}{|N|}$$

Hence, we take into consideration all available users denoted by U, we compute $full\_know$ representing the full knowledge of concept c considering all available users.

$$full\_know(c) = \max_{\forall u \in U} \{know(c, u)\}$$

Finally, $reach$ denotes the heuristic estimating the reachability of the state S from the current state. In other words it estimates the applicability of a web service resulting in state S. Req denotes the set of concepts representing the input parameters and preconditions of this web service. It is equal to 1 if all the concepts in Req are known. It decreases to 0 as the estimation of users' knowledge decreases.

$$reach(S) = \frac{1}{1 + \sum_{\forall r \in Req} [1 - full\_know(r)]}$$

The resulting $reach$ heuristic is used to speed up the search. It can be used in such a way that we determine a *threshold* deciding if we believe that the given state is reachable or not. In other case, we combine it with the heuristic estimating the distance between the actual and goal state.

The $reach$ heuristic can be exploited in several ways depending on the concrete conditions. If we assume static composition, i.e. the world is not changing during the planning, we can apply preprocessing before planning to compute the $reach$ heuristic for each state taking into consideration available users. If the composition is dynamic, we need to perform computation in each step of the search.

At the end we need to be sure that the users are really capable to input data or change conditions estimated by the $reach$ heuristic. If we are not sure, i.e. the values of $reach$ are not always 1, we need to ask the user. This can interleave with the composition. Based on the $reach$ heuristic we can ask the user questions which we consider to be answered positively (we reduce the number of questions). Before this we can interrupt the planning or parallely continue the most feasible way. The $reach$ heuristic is capable also of implicit determination of the user(s) who should be involved in the composition.

5 Conclusions and Future Work

This paper presents a proposal for improving web service composition, which is currently practically hard to be used in practice. We anticipate that the user(s) should be incorporated in most problematic situations in this process. Our aim is to examine approaches exploiting users most effectively and do bother them minimally (this is not restricted only to the composition but also to the process of a goal gathering and the execution). This should be achieved in such a way that we maximize the automation of the composition process, make suggestion for the users, offer them tools supporting their work in collaborative manner to create the desired workflow. Our proposed approach speeds up the planning, determines the required input data and the users needed to be involved. The exploitation of the approach is domain dependent and needs to be adopted to it.

Our approach is developed in a context of project *Semco-WS* dealing with semantic web services composition. We experiment with proposed approach in a domain of crisis management. The aim is to help the users of the crisis management system to realize complex workflows to succeed in crisis situations.

In future work we plan to complete our proposal and examine it in more details. In particular, we deal with the definition of the neighbor set N. We will also examine whether it is important to distinguish between the concepts describing the input data and preconditions in the metric $know(c, u)$ or we can consider the knowledge of these concepts uniquely, i.e. if there is an important difference between the capability of the users to input some data and achieve some conditions in this context. Work should be done also to examine the impact of calibration of the used heuristics and the dependencies on the domain.

References

[1] P. Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87–129, 1996.

[2] J. Cardoso and A. P. Sheth. *Semantic Web Services, Processes and Applications*. Springer, 2006.

[3] L. Hluchy, O. Habala, M. Babik, M. Laclavik, Z. Balogh, and E. Gatial. Knowledge-based platform for environmental risk management. In *ISPDC '07: Proceedings of the Sixth International Symposium on Parallel and Distributed Computing*, pages 3–9, Washington, DC, USA, 2007.

[4] C. Kiefer, A. Bernstein, and M. Stocker. The fundamentals of isparql: A virtual triple approach for similarity-based semantic web tasks. In *6th International Semantic Web Conference (ISWC)*, pages 295–309. 2007.

[5] J. Kim and Y. Gil. Towards interactive composition of semantic web services. In *Semantic Web Services - 2004 AAAI Spring Symposium*, 2004.

[6] J. W. Kim and K. S. Candan. Cp/cv: concept similarity mining without frequency information from domain describing taxonomies. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 483–492, New York, NY, USA, 2006. ACM.

[7] M. Klusch. Semantic web service coordination. In M. Schumacher, H. Helin, and H. Schuldt, editors, *CASCOM - Intelligent Service Coordination in the Semantic Web*. Birkhaeuser Verlag, Springer, 2008.

[8] M. Klusch, A. Gerber, and M. Schmidt. Semantic web service composition planning with owls-xplan. In *AAAI Fall Symposium on Semantic Web and Agents*, Arlington VA, USA, 2005. AAAI Press.

[9] J. Paralic and M. Paralic. Some approaches to text mining and their potential for semantic web applications. *Information and Organizational Sciences*, 31(1):157–170, 2007.

[10] J. Peer. *Web Service Composition as AI Planning – a Survey*. University of St.Gallen, 2005.