Dynamic Web Service Composition: Use of Case Based Reasoning and AI Planning

Fouad HENNI

Mostaganem University – Algeria
fouad.henni@gmail.com

Baghdad ATMANI

Oran University – Algeria
atmani.baghdad@univ-oran.dz

Equipe de recherche Simulation, Intégration et Fouille de données (SIF)
Laboratoire d'Informatique d'Oran (LIO)

Abstract: Web services have emerged as a major technology for deploying automated interactions between distributed and heterogeneous applications. The main advantage of web services composition is the possibility of creating value-added services by combining existing ones to achieve customized tasks. How to combine these services efficiently into an arrangement that is both functionally sound and architecturally realizable is a very challenging topic that has founded a significant research area within computer science. A great deal of recent web-related research has concentrated on dynamic web service composition. Most of proposed models for dynamic composition use semantic descriptions of web services through the construction of domain ontology. In this paper, we present our approach to dynamically produce composite services. It is based on the use of two AI techniques: Case-Based Reasoning and AI planning. Our motivating scenario concerns a national system for the monitoring of childhood immunization.

Keywords: semantic Web services, dynamic composition, OWL-S, CBR, AI planning, immunization system

1 Introduction

A Web service is a software component identified by a URL, whose public interfaces and bindings are defined and described using XML. Web services provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks [1]. This has led to the emergence of Web services as a standard mechanism for accessing information and software components programmatically [2].

Service composition refers to the technique of composing arbitrarily complex services from relatively simpler services available over the Internet. Composition of Web services enables businesses to interact with each other and facilitates seamless business-to-business or enterprise application integration. Applications are to be assembled from a set of appropriate Web services and no longer written manually [3]. For example, a composite Web service for an online order from a retailer Web site could bring together a number of internal and external services such as credit checking, inventory status checking, inventory update, shipping, etc.

Web Service Composition is currently one the most hyped and addressed issue in the Service Oriented Computing. Several models, techniques and languages have been proposed to achieve service composition.

The construction of a composite Web service can be made up in three main steps (not necessarily in this order): (a) Creation of the *process model* specifying control and data flow among the activities. (b) *Discovery*, *selection* and *binding* of concrete Web services to every activity in the process model. (c) *Execution* of the composite service by a coordinating entity (e.g. a process execution engine) [4].

In *static* composition the process model is created manually and the bindings of concrete Web services to the process activities are done at design time. *Semi-dynamic* composition strategies actively support the user with the creation of the process model and/or in the services selection and bindings. Finally, in *Dynamic composition* the creation of the process model and the services selection and bindings are made at runtime. In this paper, the focus will be done on dynamic composition of services.

The remainder of this paper is organized as follows: Section 2 presents the main ideas in dynamic composition of Web services and particularly the use of Case-Based Reasoning (CBR) and AI planning. Our proposal of using both CBR and AI planning is described in section 3, while section 4 presents a scenario as a direct application of our proposal. The paper is concluded by a discussion of the solution, some limitations, and future works.

2 Dynamic Web service composition

In dynamic composition, automated tools are used to analyze a user query, and select and assemble Web service interfaces so that their composition will solve the user demand. From a user perspective, the composite service will continue to be considered as a simple service, even though it is composed of several Web services.

In order to support greater automation of service selection and invocation, recognition is growing of the need for richer semantic specifications of Web services, so as to enable fuller, more flexible automation of service provision and use, support the construction of more powerful tools and methodologies, and promote the use of semantically well-founded reasoning about services [5]. As a result, Web services have semantic descriptions in addition to their traditional standard syntactic description (WSDL). This is referred to as semantic Web services.

Semantic Web services solve Web service problems semantically and address Web services descriptions as a whole [6]. Semantic markup languages such as OWL-S [5, 7], WSDL-S [8] and SAWSDL [9] describe Web service capabilities and contents in a computer-interpretable language and improve service discovery, invocation, composition, monitoring, and recovery quality.

Several methods and tools have been proposed for dynamic Web service composition [2, 3, 10, 11, 12]. The majority of researches conducted in dynamic composition have their origins in the realm of artificial intelligence [10].

It is not in the scope of this paper to present an exhaustive list of all methods and techniques proposed for dynamic composition. In this work, we are particularly interested in the use of CBR and AI planning in order to achieve a dynamic composition.

IA Planning is certainly the area that offers the most operational solutions in dynamic composition of services [13-16]. Several tools are available for research use and many studies still try to improve the performances, in particular by proposing AI planners dedicated to dynamic generation of composite Web services plans.

On the other hand, recent research used CBR efficiently in dynamic (or semi-dynamic) Web service composition. We aim to apply CBR over an AI Planner. The idea is that the used plans are generated by an AI planner and whenever a new query is given the system first attempts to get a solution from the stored cases. If no similar case is found, or in case of unsatisfactory solution: a planner is used to generate a new solution from scratch.

The following subsections present the main ideas in using AI planning and CBR for dynamic Web service composition.

2.1 AI Planning for Web service composition

Let's recall that a planning problem can be defined as a five tuple $<S, S_0, G, A, \Gamma>$ where, S is the set of all possible states of the world, $S_0 \subset S$ denotes the initial state of the world, $G \subset S$ denotes the goal state of the world that the planning system attempts to reach, A is the set of actions the planner can perform in attempting to change one state to another state in the world and the transition relation $\Gamma \subset SxAxS$ defines the precondition and effects for the execution of each action [10].

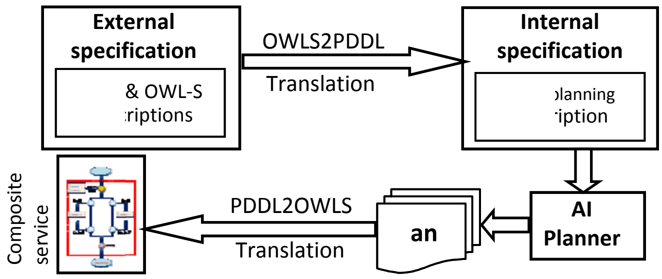

Figure 1: Applying AI planning to Web service composition

A simple analogy can be made between a Web service composition problem and a planning problem as follows: consider the user query as the initial state (S_0) of the world; the set of available Web services represents the set (A) of actions; Web service inputs (resp. ouputs) represent the precondition (resp. effects) of the corresponding action. This correspondence makes it possible to transform a Web service composition problem into a planning problem. Then, an AI planner can be used to derive a plan to offer an acceptable solution to the user query.

This transformation can be done by translating the original description of the problem into a description which corresponds to a planning problem. A Web service composition problem is often described using the OWL-S language [7]. This description is referred to as the *external specification*. On the other hand, the PDDL language [17] is most often used for the description of a planning problem. This description is referred to as the *internal specification*. Figure 1 depicts the overall principle of resolving a Web service composition problem by using AI planning.

Many research works [13, 15, 16] used the principle of figure 1 to generate a composition plan automatically. However, there are some limits in translating OWL-S descriptions into PDDL. These restrictions concern some complex plan structures allowed by OWL-S (such as unordered and iterations) but not permitted in PDDL.

2.2 Case based reasoning for Web service composition

Case-based reasoning is a problem solving paradigm that in many respects is fundamentally different from other major AI approaches [18]. In CBR, the primary knowledge source is a memory of stored cases (case base) recording specific prior episodes. The processes involved in CBR can be described by: A new problem is matched against cases in the case base and one or more similar cases are *retrieved*. A solution suggested by the matching cases is then *reused* and tested for success. Unless the retrieved case is a close match the solution will probably have to be *revised* producing a new case that can be *retained* [19] (figure 2).

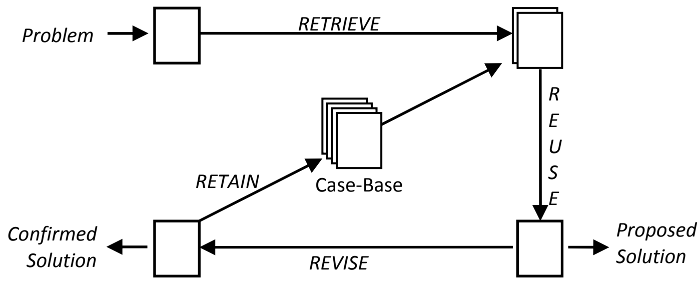

Figure 2: The CBR Cycle [19]

During the last few years, many research works used CBR in Web service composition. We present in the following the main ideas published in this area.

Lajmi et al. [22] propose an approach called WeSCo CBR that aims at enhancing the process of Web service composition by using a CBR technique. Web services are annotated using OWL-S and grouped into communities to facilitate the search process. In order to improve the search of the most relevant case (for a new case), a classification of the existing cases is proposed. The proposed solution is intended to respond to a request for a medical diagnosis of the early detection of cardiac ischemia and arrhythmia.

Osman et al. [20] present an approach that uses CBR for modeling dynamic Web service discovery and matchmaking. The framework considers Web services execution experiences in the decision making process and it is sensitive to rules issued by the service requester. The framework also uses OWL semantic descriptions extensively to implement the components of the CBR engine, as well as the services selection profiles. In addition, the proposal uses a classification of user groups into profiles that have standard set of constraint rankings.

Recently, Lee et al. [6] build a prototype that combines planning and CBR for dynamic service composition. The work accepts a service request from a user through intent analysis producing a goal model by extending the service request with keywords representing the user intent. CBR is used to provide composite services quickly. The tool JSHOP2 [14] is used to generate composition plans. The work used simulated Web services for transport, including airline tickets and other services. It also proposed merging internal and external services to meet user needs.

3. Our Proposal

Our approach to dynamically produce composite services is based on the use of case-based reasoning and AI planning. We apply CBR to store planning and related information in a case base to create planning much faster when users have similar needs. The overall architecture of the system is depicted in figure 3.

A case is a triplet consisting of the goal model extracted from the query, the corresponding OWL-S solution and an outcome. The goal model is used as features for case searching and matchmaking.

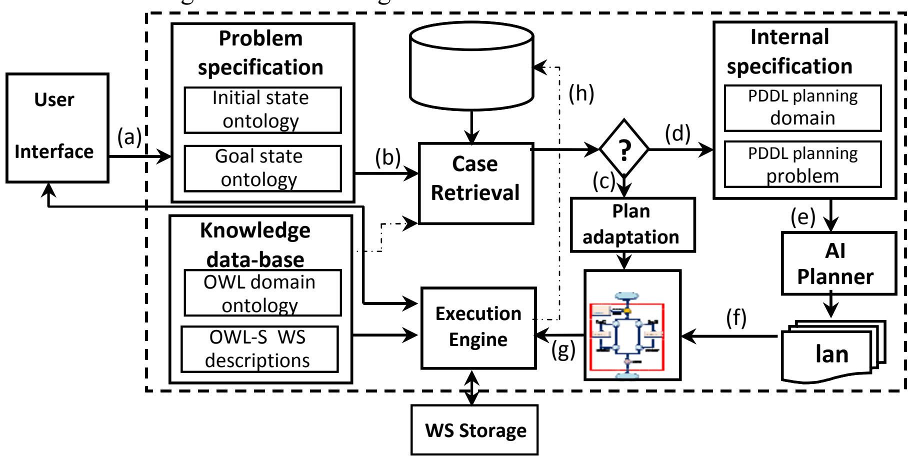

Figure 3: Architecture of the proposed solution

a) A new query is introduced via the user interface. This query is considered as a new case and is semantically annotated using OWL-S.
b) The case retrieval module tries to find a match for the new case in the case base.
c) Unless the retrieved case is an exact match, an adaptation of the corresponding solution is necessary.
d) When no matches exist, or in case or unsatisfactory solution, the new problem is translated into a planning problem.
e) An AI planner is used to derive a new plan for the translated problem. Our system uses OWLS-Xplan2.0 [23] to generate a new AI composition plan.
f) In order to be executed, the generated plan is translated into OWL-S.

g) The execution engine binds the composite service activities to concrete Web services (by querying service registries) and returns the resulting composite service to the user. An evaluation of the proposed solution is then made.
h) Depending on the evaluation the new case can be stored in the case base.

4. Motivating scenario

Our prototype for dynamic Web service composition is currently applied in a national research project (PNR 12/u310/65) [24] that concerns the Monitoring of Childhood Immunization (MCI) in Algeria. The system presently underway aims to have total immunization coverage and an access to the immunization status of every child from any department all over the country. In order to insure that every child is immunized according to a fixed calendar a vaccination notebook (VN) is established and maintained by the immunization monitoring service (IMS). This notebook is generated by the IMS of the municipality where the child was born (city of birth CB). Every municipality is attached to an IMS which in turn monitors several immunization services (IS). Children are dispatched into different ISs according to their parents' address (PA at birth date).
The information manipulated by the MCI system comes from many sources:

a) The birth registry located at the municipality: Information about the child's name, date of birth, parents' names, hospital of birth, name of the doctor, etc.
b) The address registry located at the municipality: Information about the IS a child is assigned to according to his PA and to the urban cutting.
c) The vaccination notebooks registry located at an IMS: The history of previous vaccinations for a given child and the schedule of incoming immunizations.

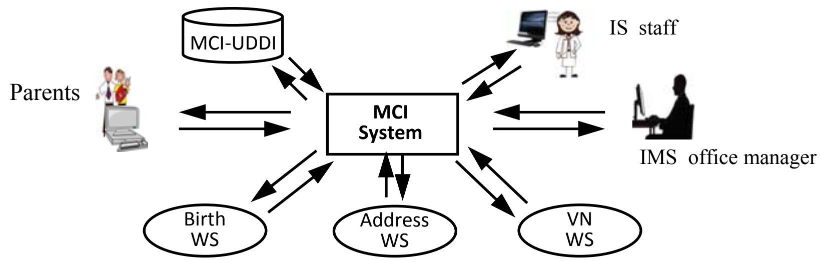

Figure 4: A Service Oriented Architecture for the MCI-System

Web services are used to access each registry. Every municipality and every IMS has its own registries. And even though the structure of information stored in different municipalities or IMSs is roughly the same (e.g. the birth registry), different Web services should be implemented because of particular considerations (e.g. use of different DBMS). It means that the activities are exactly the same for all municipalities and IMSs, but each of which may rely on a different technological platform. All Web services are advertised in a private UDDI called MCI-UDDI. Figure 4 depicts the overall functional structure of the MCI. Domain ontology is developed which allows giving OWL-S annotations for published Web services.

Queries to the MCI system come from different types of users and each query triggers a composition of services depending on the information given by the user (CB, PA, ..), the type of user, and the desired result.

5. Conclusion and discussions

We presented a solution that combines CBR and AI planning for dynamic composition of services. Instead of testing the solution on simulated Web services we have chosen to apply our proposal on a real example. The use of CBR gives a way to memorize past experiences in order to reuse previous successful solutions. As a result, a solution is provided quickly. On the other hand, the use of AI planning allows proposing a solution when no previous similar cases exist or when the proposed solution does not satisfy the user. AI planning also allows populating the case base when applying our solution in a new domain. The advantage of using PDDL is to pave the way toward the use of a wide range of planners. Moreover, in addition of using an existing planner, we are implementing a new AI planner that utilizes the principle of the cellular machine [25]. The objective is to produce faster and more efficient plans.

A few issues in the use of CBR are still under examination. In particular we are experiencing the use of decision trees to improve the similarity calculus as in [26]. The other issue is the adaptation of a solution. We are still working on a satisfactory approach to adapt an existing solution.

6. Referencies

1. Web Services Architecture. http://www.w3.org/TR/ws-arch/wsa.pdf.
2. Agarwal, V., Chafle, G., Mittal, S., Strivastava, B.: Understanding Approaches for Web Service Composition and Execution. Proc. Of the 1st Bangalore Annual Compute Conference (COMPUTE'08), 2008.
3. Srivastava, B., Koehler, J.: Web Service Composition Current Solutions and Open Problems. In Workshop on Planning for Web Services (ICAPS), 2003.
4. Prasath Sivasubramanian, S., Ilavarasan, E., Vadivelou, G.: Dynamic Web Service Composition: Challenges and Techniques. Int. Conf. on Intelligent Agent & Multi-Agent Systems (IAMA'09), 2009 (1-8).
5. Martin, D., Paolucci, M., McIlraith, S., Burstein, M., McDermott, D., McGuinness, D., Parsia, B.,Payne, T., Sabou, M.,Solanki, M., Srinivasan, N., and Sycara, K., Bringing Semantics to Web Services: The OWL-S Approach. Proc. 1st Int. Workshop on Semantic Web Services and Web Process Composition (SWSWPC'04), California, USA, 2004.
6. Lee, C.L., Liu, A., and Huang, H.: Using Planning and Case-Based Reasoning for Service Composition. In J. Advanced Computational Intelligence and Intelligent Informatics, Vol. 14, Issue 5, 2010 (540-548).
7. OWL-S: Semantic Markup for Web Services. http://www.w3.org/Submission/OWL-S/.

8. Miller, J., Verma, K., Shelth, A., Aggarwal, R., Sivashanmugan, K.: WSDL-S: Adding Semantics to WSDL white paper. Technical Report, Large Scale Distributed Information Systems (2004).
9. Semantic Annotations for WSDL and XML Schema – Usage Guide. Available at: http://www.w3.org/TR/sawsdl-guide/.
10. Rao, J., Su, X.: A Survey of Automated Web Service Composition Methods. In Proc. 1st Int. Workshop on Semantic Web Services and Web Process Composition, July 2004.
11. PonHarshavardhan, Akilandeswari, J., Manjari, M.: Dynamic Web Service Composition Problems and Solution -A Survey. In Proc. 2nd Int. Conference on Information Systems and Technology (ICIST), India, May 2011 (1-5).
12. Milanovic, N., Malek, M.; Current Solutions for Web Service Composition. IEEE Internet Computing, Vol. 8, No.6, 2004 (51-59).
13. Ďurčík, Z.: Automated Web Service Composition with Knowledge Approach. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 2, No. 2, 2010 (35-42).
14. Wu, D., Sirin, E., Hendler, J., Nau, D., Parsia., B.: Automatic Web Services Composition using SHOP2. In Proc. 2nd Int. Semantic Web Conference (ISWC), 2003.
15. Klusch , M., Gerber, A.: Semantic Web Service Composition Planning with OWLS-XPlan, 1st Int. AAAI Fall Symposium on Agents and the Semantic Web, USA, 2005.
16. Peer, J.: A PDDL Based Tool for Automatic Web Service Composition. Lecture Notes in Computer Science, Vol. 3208, 2004 (149-163).
17. Ghallab, M., Howe, A., Knoblock, C., McDermott, D., Ram, A., Veloso, M., Weld, D., Wilkins, D.: PDDL - The Planning Domain Definition Language, Version 1.2. Yale Center for Computational Vision and Control, Tech. Report CVC TR-98-003/DCS TR-1165, October, 1998.
18. Aamodt , A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications, Vol.7 No.1, 1994 (39-59).
19. Watson, I., Marir, F : Case-Based Reasoning: A Review. The Knowledge Engineering Review, Vol. 9, No. 4, 1994 (355-381).
20. Osman, T., Dhavalkumar Thakker, D., Al-Dabass, D.; Semantic-Driven Matchmaking of Web Services Using Case-Based Reasoning. IEEE Int. Conference on Web Services (ICWS'06), 2006 (29-36).
21. Leake, B.: CBR in Context: The Present and Future. In Leake, D., ed., 1996, Case-Based Reasoning: Experiences, Lessons, and Future Directions. Menlo Park: AAAI Press/MIT Press, 1996.
22. Lajmi, S., Ghedira, C., Ghedira, K., Benslimane, D.: CBR Method for Web Service Composition. In Advanced Internet Based Systems and Applications, Lecture Notes in Computer Science, Vol. 4879, 2009 (314-326).
23. Sem WebCentral. Available in: http://www.semwebcentral.org/projects/owls-xplan/.
24. Accepted PNR projects. Available in: http://www.nasr-dz.org/dprep/pnr2/projets-pnr/PNR_a.htm.
25. Atmani, B., Beldjilali., B.: Knowledge Discovery in Database: Induction Graph and Cellular Automaton. J. Computing and Informatics, Vol. 26, No. 2, 2007, (1001-1027).
26. Benbelkacem, S., Atmani,B., Mansoul, A.: Planification Guidée par Raisonnement à Base de Cas et Datamining : Remémoration des Cas par Arbre de Décision. Accepted paper in EGC 2012 : http://eric.univ-lyon2.fr/~aide/?page=accepted_papers.html, Bordeaux, France, Jan 2012.
27. Kolodner, J. L.: Case-Based Reasoning. Morgan Kaufmann, San Mateo, 1993.