

# Automatic Web Services Composition Using SHOP2

Dan Wu<sup>+</sup> Evren Sirin<sup>+</sup> James Hendler<sup>+</sup> Dana Nau<sup>+</sup> Bijan Parsia<sup>\*</sup>

Department of Computer Science  
University of Maryland  
College Park, MD 20742

<sup>+</sup>{dandan, evren, hendler, nau}@cs.umd.edu

<sup>\*</sup>bparsia@isr.umd.edu

## ABSTRACT

Semantic markup of Web services will enable the automation of various kinds of tasks, including discovery, composition, and execution of Web services. We describe how an AI planning system (SHOP2) can be used with DAML-S Web service descriptions to automatically compose Web services.

## Categories and Subject Descriptors

I.2.4 [Artificial Intelligence]: Knowledge Representation Formalisms and Methods—*Representation languages*

## General Terms

Algorithms, Theory, Languages

## Keywords

Semantic Web, DAML-S, Web Services, Web Service Composition, Automated Reasoning

## 1. INTRODUCTION

As Web services – that is, programs and devices accessible via standard Web protocols – proliferate, it becomes more difficult to find the specific service that can perform the task at hand. It becomes even more difficult when there is no single service capable of performing that task, but there are combinations of existing services that could. Sufficiently rich, machine readable descriptions of Web services would allow the creation of novel, compound Web services with little or no direct human intervention. Semantic Web languages, such as the Web Ontology Language (OWL) or DAML+OIL, provide the foundations for such sufficiently rich descriptions.

In May 2001, the DARPA Agent Markup Language (DAML) Program released the first version of DAML-S [1], a set of ontologies for describing the properties and capabilities of Web services. The purpose of DAML-S markup of Web services is to support effective automation of various kinds of tasks including Web service discovery, composition, execution, and monitoring.

For our work, we are motivated by issues related to automated Web service composition. One part of DAML-S, namely its process ontology, provides a standard language for describing the composition of Web services. Below, we describe how SHOP2 [2] — an AI planning system — can be used with DAML-S Web service descriptions to automatically compose Web services.

Copyright is held by the author/owner(s).  
WWW2003, May 20–24, 2003, Budapest, Hungary.  
ACM xxx.

## 2. MOTIVATING EXAMPLE

The example we describe here is a variation of the example described in Scientific American article [3]. Suppose Bill and Joan's mother goes to her physician complaining of pain and tingling in her legs and the physician proposes the following sequence of activities: 1) A prescription for Relafen, an anti-inflammatory drug; 2) An MRI scan and an electromyography, both of these are diagnostic tests to try to determine possible causes for the symptoms; 3) A follow-up appointment with the physician to discuss the results of the diagnostic tests.

Bill and Joan need to do the following things for their mother:

- fill the prescription at a pharmacy;
- make appointments for the two treatments;
- make an appointment for doctor's follow-up meeting.

For the three appointment times, there are the following preferences and constraints:

- For the two treatments:
  - Bill and Joan would prefer two appointment times that are close together scheduled at one or two nearby places, so that only one person needs to drive once.
  - Otherwise, they would prefer two appointment times on different days, so that each person needs to drive once.
- The appointment time for doctor's follow up check must be later than the appointment times for the two treatments.
- An appointment time must fit the schedule of the person that will drive to the appointment.

Consider a possible scenario in the near future, where Bill and Joan can use Web services to schedule their mother's appointments. It would be difficult for Bill and Joan to finish their task by consulting the Web services manually, because:

- They may have to try every available pair of close appointment times at any two nearby treatment centers in order to find one that fits their schedules.
- Furthermore, if they first choose an appointment time for one treatment and then find they have to use this same time for the other treatment, then they have to reschedule the first appointment.

Instead, suppose we use the DAML-S process ontology to encode a description of how to compose Web services for tasks such as the one faced by Bill and Joan. If we have an agent technology which can implement this encoding, then we can perform Bill and Joan's Web services composition task automatically.

## 3. BACKGROUND

### 3.1 DAML-S

In the DAML-S process ontology, each Web service is modelled as a process. A process can be either atomic or composite. An atomic process represents a directly executable Web service. A composite process can be decomposed into other atomic or composite processes. How to decompose a composite processes is specified through its control constructs. A composite process represents a compound Web service. The goal of automatic service composition is as follows: for a composite process instance, find a collection of atomic process instances that forms a legal composition of this instance based on the composite process's DAML-S definition.

More specifically, we can build an agent that can plan a collection of Web service requests, whose execution will achieve the user's goal. HTN (Hierarchical Task Network) planning seems very promising for this, because the concept of task decomposition in HTN planning is very similar to the concept of composite process decomposition in DAML-S.

### 3.2 SHOP2

SHOP2 is a domain-independent HTN planning system. HTN planning is an AI planning methodology that creates plans by task decomposition. This is a process in which the planning system decomposes tasks into smaller and smaller subtasks, until primitive tasks are found that can be performed directly. SHOP2's knowledge base contains operators and methods. Each operator is a description of what needs to be done to accomplish some primitive task, and each method tells how to decompose some compound task into partially ordered subtasks.

A planning problem in SHOP2 is a triple  $(S, T, D)$ , where  $S$  is initial state,  $T$  is a goal task list, and  $D$  is a domain description. SHOP2 will return a plan  $P = (p_1 p_2 \dots p_n)$ , a sequence of instantiated operators that will achieve  $T$  from  $S$  in  $D$ .

## 4. FROM DAML-S TO SHOP2

To use SHOP2 for automatic DAML-S service composition, we encode a collection of DAML-S process definitions  $M$  for a group of Web services into a SHOP2 domain  $D$  as follows:<sup>1</sup>

- We encode each atomic process with effects in  $M$  as a SHOP2 operator that simulates the effects of a world-altering Web service by changing its local state via the operator. Therefore, such a Web service is not executed during the planning process and SHOP2 can backtrack.
- We encode each atomic process with output in  $M$  as a SHOP2 operator whose precondition include a call to an information collecting Web service. In this way, the information collecting Web service is executed during the planning process.
- We encode each composite process definition in  $M$  as one or more SHOP2 methods that tell how to decompose an HTN task that represents the composite process.

The instance of a composite process  $T$  in  $M$  will become a task for SHOP2. SHOP2's recursive decomposition of this task into operator instances will correspond directly to the recursive decomposition of the composite process instance into atomic processes instances.

<sup>1</sup>In our translation, we assume that in DAML-S, an atomic process can either have effects or outputs. An atomic process with only output models an information collecting Web service. An atomic process with only effect models an world altering Web service

## 5. IMPLEMENTATION

Our implementation includes:

- A DAML-S to SHOP2 translator which translates a collection of DAML-S process definitions into a SHOP domain.
- An interface which lets users specify the request for a service.
- A monitor which handles SHOP2's calls to external information collecting Web services during planning. This monitor system will cache the responses of the information collecting Web services to avoid invoking a Web service with same parameters more than once during planning. We assume that the cached information will not be changed by other agents during planning; we will generalize this in our future work
- A SHOP2 to DAML-S plan converter, which will convert a plan to DAML-S format which can be directly executed by a DAML-S executor.

To test the effectiveness of our approach, we have run SHOP2 on several instances of the problem described in Section 2. These problem instances varied from cases where it was easy to schedule satisfactory appointments to a case in which no nearby treatment centers had treatment times slot were close together, so that Bill and Joan would both have to drive Mom for treatments on separate days. In all of these cases, SHOP2 was easily able to find the best possible solution.

## 6. RELATED WORK

Our technology here is based on work with SHOP family of planning systems, with SHOP2 being the newest one in the family. One other technology that addresses aspect of automatic Web service composition and that deserves mention is the work of McIlraith [4] and others in Stanford University on adapting Golog for programming the semantic Web.

## 7. CONCLUSION

We have described a way to address the problem of automatic Web service composition by exploiting AI planning techniques, in particular, the SHOP2 planning system. Our approach involves translating DAML-S definitions of Web services into SHOP2 methods and operators, so that SHOP2 can be used with DAML-S Web service descriptions to automatically compose Web services.

## 8. ACKNOWLEDGMENTS

This work was supported in part by Air Force Research Laboratory grant F30602-00-2-0505.

## 9. REFERENCES

- [1] The DAML Services Coalition. DAML-S: Web Service Description for the Semantic Web. *The First International Semantic Web Conference (ISWC)*, Sardinia (Italy), June, 2002.
- [2] D. Nau, H. Muñoz-Avila, Y. Cao, A. Lotem, and S. Mitchell. Total-Order Planning with Partially Ordered Subtasks. In *IJCAI-2001*, Seattle, August, 2001.
- [3] T. Berners-Lee, J. Hendler, O. Lassila. The Semantic Web. *Scientific American*, May, 2001.
- [4] S. McIlraith and T. Son. Adapting Golog for Composition of Semantic Web Services. *Proceedings of the Eighth International Conference on Knowledge Representation and Reasoning (KR2002)*, Toulouse, France, April, 2002.