Collaborative Design Systems as Semantic Web Services

Cássio V. S. Prazeres
University of São Paulo
prazeres@icmc.usp.br

Abstract. OWL-S enables the semantic description of Web Services, aiming at the automation of tasks such as discovery, selection, composition and monitoring of Web Services. Collaborative design systems has some needs to help designers in assembling design components by helping them in searching adequate components, detecting conflicts, searching related documents, finding people with the adequate skills, etc. In this paper, we propose that collaborative systems are represented as Semantic Web Services to provide automatic discovery, selection, composition and monitoring of these systems and of the components, the related documents and the people involved in the systems. We also propose a way to booking of resources and people involved in a collaborative activity.

Key words: Collaborative Systems, Semantic Web Services, OWL-S

1 Introduction

The semantic behind Semantic Web Services brings the opportunity of enhancing the automation of tasks such as Web Services discovery, selection, composition and monitoring. The Ontology Web Language for Services (OWL-S) has been designed to allow rich semantic specifications to be associated with Web Services [1]. More specifically, the OWL-S ontology formalizes three essential types of knowledge about a service: “what the service does”, “how the service works” and “how to access the service” [1].

One need in the development of collaborative systems which the design work can be done by several people distributed about time, space and organizations is locate and select adequate components, search for related documents, find people with the adequate skills, etc. Another need is to booking resources that may require exclusive access or may impose some requirements, as advanced booking, for instance or the booking of meetings between people involved in a collaborative activity.

In this paper, we propose that collaborative systems are represented as Semantic Web Services to provide automatic discovery, selection, composition and monitoring of these systems and of the components, the related documents, the people, etc. involved in the systems. We also propose a way to booking of resources and people involved in a collaborative activity.
This paper is organized as follows. Section 2 presents some related ontologies that we reuse in our proposal. A description of the proposal is presented in Section 3. Section 4 presents some related works. Section 5 summarizes our contributions.

2 Related Ontologies

This section presents the OWL-S ontology and some related ontologies that we intend to use in our proposal to turn computer-based collaborative design systems into Semantic Web Services.

2.1 OWL-S

Figure 1 shows the OWL-S upper ontology for services. The intent of this ontology is to provide three essential types of knowledge about a service: the ServiceProfile, the ServiceModel and the ServiceGrounding (shown in Figure 1) [1].

![Fig. 1. OWL-S Ontology for Services [1].](image)

The class ServiceProfile provides a superclass of every type of high-level description of the service. The ServiceProfile is used for publishing (providers) and discovering (requesters) services. The OWL-S Profile focuses on two aspects of the service functionality: a) inputs and outputs represent information transformation; b) preconditions and effects represent state changes produced by the execution of the service, which forms the IOPE (inputs, outputs, preconditions and effects). For a detailed perspective on how a service operates, a service is better viewed it as a process. Specifically, OWL-S defines a subclass of ServiceModel, the ProcessModel.

A process in OWL-S has two functions. One is to produce a data transformation from a set of inputs to a set of outputs. Second, to produce a transition in the world from one state to another. This transition is described by means of the preconditions and the effects of the process.

The grounding of a service specifies the details of how to access the service – details having mainly to do with protocol and message formats, serialization, transport, and addressing. A grounding can be thought of as a mapping from an
abstract to a concrete specification of those service description elements that are required for interacting with the service – in particular, the inputs and outputs of atomic processes.

2.2 Time Ontology

OWL-Time\footnote{http://www.isi.edu/~pan/damltime/time.owl} ontology is a complete specification of time that is required for Semantic Web applications \cite{2}. The purpose of this ontology is to provide quick access to the essential vocabulary in OWL for the basic temporal concepts and relations.

![Fig. 2. Subclass hierarchy of temporal concepts.](image)

Figure 2 presents the subclass hierarchy of temporal concepts proposed for the Time ontology. We use this ontology to define temporal concepts such as \textit{Instant}, \textit{Interval}, \textit{InstantEvent}, and \textit{IntervalEvent}. These concepts can be used to model temporal restrictions in resources (as documents) or persons in the collaborative design systems.

3 Proposal

Section 2 presented the OWL-S ontology for services and the Time ontology. In this section, we present our proposal to use Semantic Web Services with OWL-S and Time ontology in the support to collaborative design activity. We propose to develop all collaborative design systems as Semantic Web Services to provide automatic discovery, selection, composition, and monitoring of these systems and their involved resources. We also propose to use Time ontology to provide temporal restriction in resources that may require exclusive access or may impose some requirements, as advanced booking, for instance.
3.1 Discovery, Selection and Composition

The task of service composition in OWL-S involves the automatic discovery, selection, composition, and interoperation of Web services to perform some task, given a high-level description of an objective. With the OWL-S markup of Web services, the information necessary to select and compose services will be encoded at service Web sites. Software can be written to manipulate these representations, together with a specification of the objectives of the task, to achieve them automatically [1].

The development of collaborative systems, which the design work can be done by several people distributed about time, space and organizations, requires to locate and select adequate components, search for related documents, find people with the adequate skills, etc., and, in a final step, compose collaborative systems with these distributed components.

Collaborative design systems may use of the Semantic Web Services properties as automatic discovery, selection and composition to integrate several distributed collaborative systems in an automatic way. Components, related documents and people with the adequate skills in a collaborative systems can be represented as Semantic Web Services to provide automatic discovery, selection and composition of collaborative design activities.

3.2 Booking

Particularly in the context of OWL-S, discovery algorithms focus on the functionality aspects of the services, which are usually given by the transformation from a set of inputs to a set of outputs. Discovery strategies usually do not take into account that some services (or resources) may not be available around the clock and, more restrictively, may require exclusive access. Such service access restrictions may impose some requirements, as advanced booking, for instance. The information regarding the category of a service, which may be seen as a property of the service in the OWL-S specifications, may be used to identify the services that need to be previously booked.

We use the Time ontology to add temporal concepts to the service and resource descriptions so as to make it possible to provide booking and to impose restrictions in the use of the resources involved in the service.

We propose an approach [3] to allow the discovery of Web Services based not only on service categories or service functionality, but also on temporal restrictions. We use the Time [4] ontology to formalize temporal concepts and to present a matching algorithm which leverages the work reported in the literature, so as to consider availability matching.

3.3 Monitoring

Individual services and, even more, compositions of services, may often require some time to execute completely. A user may want to know, during this period, how is the status of his or her request, in a collaborative design system. Also,
plans may change, requiring alterations in the actions the software agent must take [1].

4 Related Works

In a previous work [5], we have proposed supporting remote experiments as Semantic Web Services. They use ontologies to formalize experiments as Semantic Web Service. The aim is to promote the standardization of tasks such as experiment discovery, invocation, composition and monitoring as well to automate these tasks. In another previous work [6], we have proposed an approach, called Semantic and Context-Aware Available Services (SeCoAS), that combines Semantic Context Modeling and Reasoning with Semantic Web Services.

We have discussed elsewhere parallel efforts toward making WebLabs as Semantic Web Services [5], as well toward investigating problems associated with the construction of applications presenting temporal restrictions, we have proposed an approach [3] to allow the discovery of Web Services based on service parameters, service categories and temporal restrictions.

The service discovery in OWL-S is made by matching IOPE’s from the requester with the ones from the provider of the service. Xie et al. [7] propose the use of natural language for access web services described with OWL-S. This solution can be used together with our proposal to map user query requests (in natural language) into their corresponding web services described with OWL-S. In Section 3.3 we proposed the use of a monitoring service to monitor the collaborative design activities. The references [8–10] present proposals of monitoring Semantic Web Services and may be evaluated to use with our proposal.

5 Final Remarks

We have presented a proposal to represent collaborative design systems as Semantic Web Services in order to provide automatic discovery, selection, composition, monitoring and booking of these systems and of the components, the related documents, the people, etc. involved in the systems.

Our proposal combines the Time ontology, OWL-S ontology and Resource ontology, at the same time that adds temporal concepts to the service and resource description that makes it possible to do booking and to impose restrictions in the use of the resources involved in the collaborative design systems.

By using OWL-S to describe an collaborative design systems as a Semantic Web Service, we have imported all aspects of the OWL-S to the collaborative design systems domain. Aspects like location, selection, composition, and monitoring of Web Services automatically is applied to the collaborative design systems domain.
Acknowledgments

We would like to thank the following organizations: CAPES, for financial support to Cäsio Prazeres and to the Graduate Programs in Computer Science at the University of São Paulo and at the Federal University of São Carlos.

References