Chapter V
A Multiagent Approach for Configuring and Explaining Workflow of Semantic Web Services

Vasco Furtado
University of Fortaleza, Brazil

Leonardo Ayres
University of Fortaleza, Brazil

Gustavo Fernandes
University of Fortaleza, Brazil

ABSTRACT

In this paper, we describe a multiagent approach that configures semantic Web services following a design problem solving method. For that, a propose-and-revise strategy was modeled and implemented in OWL-S. The other contribution of the approach is the extension of OWL-S for recording the service execution flow followed by the agent. The trace of the agent’s reasoning is stored in node sets represented in a proof markup language which make up a framework for explanations on the Web. The proposed approach is shown to be useful in the context of e-business where the workflow definition of services is automatically performed by the agents in order to configure a computer from a customer order.
INTRODUCTION

In the Semantic Web context, the popular queries realized by the user in Web pages tend to evolve into an interactive process between a person and intelligent software agents (Berners-Lee, Hendler, & Lassila, 2001). From users’ requirements and preferences, intelligent agents must be able to find services and, if necessary, compose them to produce the results which are expected by the users.

This automation process has led to the conception of the so-called “virtual organizations”, where organizations are dynamically linked in a way as to meet a specific objective (or many). Several works have been proposed in the sense of automating or semi-automating the processes of discovery, composition, and execution of Web services (McIlraith, Son, & Zeng, 2001; Paolucci, Kawamura, Payne, & Sycara, 2002). The proposals presented are not concerned with a fundamental aspect in this context: how to explain to the users why a particular result was produced in the automatic process of Web service selection and execution.

In this article, we describe our multiagent system that is able, from users’ goals and preferences, to automatically create a workflow of semantic Web services that, in turn, is able to design an artifact. During the distributed process of problem solving, the system generates justifications that will be used to provide explanations to the user. The generation of the final workflow of the Semantic Web services is driven by the agent’s reasoning module which follows a design problem solving method, more specifically a configuration one. The basis for such modeling was the work in the area of Knowledge Engineering that defines Problem-Solving Methods (PSM) (Fensel & Benjamins, 1998) to assist in the representation and acquisition of knowledge from experts. PSM are patterns describing how to use reasoning steps to solve knowledge-intensive tasks such as planning, scheduling, diagnosis, design, and assessment.

User’s preferences are modeled in OWL (McGuinness & van Harmelen, 2004) and are used by the agents for creating the Web service workflow. The agent’s reasoning module implements the propose-and-revise PSM in OWL-S (OWL-S Java API, 2004), a set of ontologies written in OWL for describing Web services. The rationale behind the use of OWL-S is twofold. First, we aim at having an explicit conceptual representation of the tasks the agent must realize as well as the way to link this conceptual definition with Web services that implement the tasks. The second point is that we embedded in the OWL-S API, a service for generating an infrastructure of justifications which allow the production of explanations to the user. Thus, our solution presents a distributed way to solve a design problem via agents and Semantic Web services as well as providing an explanation of how the problem was solved by means of the different components involved.

OWL-S is one of most mature and well-developed solutions for modeling and implementation of Semantic Web services technology, allowing the representation of composite processes. Also, it has an API (OWL-S Java API, 2004) that interprets the ontologies allowing the execution of services that follow the flow described in the ontology. Its extension was necessary in the sense of propitiating the capacity for explanations of the service execution. As mentioned by Moore (1995), extensiveness is a prominent feature to be observed in explanation systems, which indicates that the explanation facilities must be easily extendable in order to be easily reused for the development of other applications, thereby reducing the effort of implementation. The extension of OWL-S that we are proposing constitutes an important contribution in that sense, and was done through modifications in the OWL-S API to record node sets in PML (Proof Markup Language), a language created with the objective of representing proofs and providing explanations on the Web (Pinheiro, McGuinness, & Fikes, 2004). In this way, our methodology is shown to be extensible.
Related Content

**Ripple Effect in Web Applications**
[www.igi-global.com/article/ripple-effect-web-applications/44919?camid=4v1a](www.igi-global.com/article/ripple-effect-web-applications/44919?camid=4v1a)

**Understanding the Deployment of Competitive Intelligence Through Moments of Translation**
[www.igi-global.com/article/understanding-the-deployment-of-competitive-intelligence-through-moments-of-translation/89328?camid=4v1a](www.igi-global.com/article/understanding-the-deployment-of-competitive-intelligence-through-moments-of-translation/89328?camid=4v1a)

**Incremental Learning for Interactive E-Mail Filtering**
[www.igi-global.com/chapter/incremental-learning-interactive-mail-filtering/5031?camid=4v1a](www.igi-global.com/chapter/incremental-learning-interactive-mail-filtering/5031?camid=4v1a)

**Analysis and Customization of Web-Based Electronic Catalogs**
[www.igi-global.com/chapter/analysis-customization-web-based-electronic/31119?camid=4v1a](www.igi-global.com/chapter/analysis-customization-web-based-electronic/31119?camid=4v1a)