Ontology-Based Application Server to the Execution of Imperative Natural Language Requests

Flávia Linhalis and Dilvan de Abreu Moreira

University of São Paulo, Institute of Mathematics and Science Computing - ICMC
P.O. Box 688, São Carlos, Brazil
{flavia, dilvan}@icmc.usp.br

Abstract. This paper is about using ontologies to help the execution of imperative requests expressed in natural language. In order to achieve this goal, we developed the prototype of an Ontology-Based Application Server to the execution of Natural Language requests (NL-OBAS). The NL-OBAS provides services to allow users to describe requests in several natural languages and uses software components to execute them. One of the advantages of our approach is that natural language is first converted to an interlingua, UNL (Universal Networking Language). The interlingua allows the use of different human languages to express the requests (other systems are restricted to English). The semantics of the interlingua, enhanced by ontologies, is used to retrieve the appropriated software components to compose a dynamic service to execute the requests expressed in natural language.

1 Introduction

It is a fact that the Web has become the main interface for the exchange of information between government, institutions, small businesses and big companies computer systems and common people. An evidence of this is the growing use of application servers, like Jboss, WebSephere and Zope. Application servers provide support for Web application development by comprising functionalities of conventional middleware such as transaction monitors or object brokers, but they also incorporate technologies for Web access [1]. The goal of an application server is to transparently handle, in a Web environment, aspects like concurrency, naming, security, database connectivity (and so on) for distributed component based software. In a simplified and generalized view, we can say that an application server can facilitate the development of applications by providing transparency to the programmers. They also serve as a bridge between the users Internet browsers and the services offered by government and industry.

Application servers try to manage various issues (transactions, session management, user rights etc.) in an application independent way. This is achieved by software configured with the help of administration tools and corresponding configuration files. Though this constitutes a very flexible way of developing and administrating a distributed application, configuration files do not provide a

H. Legind Larsen et al. (Eds.): FQAS 2006, LNAI 4027, pp. 589–600, 2006.

high-level of abstraction. The reason is that they lack a coherent formal model. Thus, it is very hard to find information that depends on the integration of several descriptions. For example, managing component dependencies, versions, and licenses is a typical problem in an ever-growing repository of programming libraries and components [1].

Oberle et al. [1] try to remedy these problems by applying Semantic Web technology, i.e. ontologies and inference engines, in the application server itself. In an Ontology-based Application Server (OBAS), an ontology can join together so far separated aspects of the server like security, component dependencies, version or deployment information. This should help the programmers in their complex tasks during the development of a program.

The architecture proposed and partially implemented by Oberle et al. [1] is suitable to enterprise applications because it includes, for example, transaction manager and security services. However, it is important to emphasize that an OBAS is characterized by the presence of an ontology to integrate the services of an application server, but not necessarily enterprise capabilities should be provided.

We can say that OBAS technology is the future for today's application servers. And, in the same way that current application servers now occupy a key position in the development of Web applications, in the future, OBASs, with their semantic centric approach, will have the same role in the development of Semantic Web applications.

The Web was a revolution making digital information available to anyone with a browser and an Internet connection. But, what is the next step for the Web? What if it could help to break down or at least to drastically lower the language barrier for the Internet? The Universal Networking Language (UNL) project [2] has exactly this goal. The Internet community is still segregated by language boundaries. Theoretically, this seems to be the only major obstacle standing in the way of international and interpersonal communication in the information society [3].

The UNU/IAS (Institute of Advanced Studies of the United Nations University) initiated the UNL project aiming to solve this problem. Started in 1996, the project embraces several universities and research institutions from Brazil, China, Egypt, France, Germany, India, Indonesia, Italy, Japan, Jordan, Latvia, Mongolia, Russia, Spain and Thailand. In the following years more groups are expected to join [3].

The project proposes an interlingua, entitled Universal Networking Language (UNL)[4]. UNL has sufficient expressive power to represent relevant information conveyed by natural languages. For each natural language, two systems should be developed: a DeConverter capable of translating texts from UNL to the target natural language, and an EnConverter which has to convert natural language texts into UNL. A DeConverter and EnConverter for each language form a Language Server residing somewhere in the Internet. All language servers will be connected to allow any Internet user to deconvert an UNL document found on the web into his/her native language, as well as to produce UNL representations of the texts he/she wishes to make available to the whole Internet community [3].