An approach to integrate web services and argumentation into a BDI system

(Extended Abstract)

Federico Schlesinger
LIDIC - Universidad Nacional de San Luis
5700 San Luis, Argentina
fedest@gmail.com

Marcelo Errecalde
LIDIC - Universidad Nacional de San Luis
5700 San Luis, Argentina
merreca@unsl.edu.ar

Guillermo Aguirre
LIDIC - Universidad Nacional de San Luis
5700 San Luis, Argentina
gaguirre@unsl.edu.ar

ABSTRACT

Intelligent agents have to be provided with different skills and technological resources in order to deal with highly changing environments, uncertain, incomplete and potentially inconsistent information and bounded computational resources. BDI architectures, argumentation-based techniques and recent technologies like web services have been incorporated in the design of intelligent agents to address some of these difficult aspects. However, they have usually been focused on some partial aspects and it is not easy to see how these models and technologies could be effectively integrated in a single framework. In this work, we propose a general framework that integrates web services and argumentation-based inference into the design of a BDI system. Our proposal extends the well known advantages of BDI systems by using web services to access relevant information available on the Web and defeasible argumentation inference to manage incomplete and potentially inconsistent information.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence - Intelligent Agents, Languages and structures

General Terms

Design, Theory

Keywords

BDI Model, Web Services, Argumentation

1. INTRODUCTION

The main purpose of our work is to extend the well known advantages of BDI systems by using web services to access relevant information available on the Web and defeasible argumentation inference to manage incomplete and potentially inconsistent information. We propose for the integration of these approaches a framework (WADEX) which allows BDI agents (implemented in Jadex [2]) to use different Web Services (by means of the WSIG component [1]) and to use a general-purpose defeasible argumentation formalism (DeLP [4]) as a general knowledge representation and inference mechanism. The DeLP facilities were provided by a DeLP-Server [3], an implementation of a DeLP interpreter that follows a client-server architecture.

Cite as: An approach to integrate web services and argumentation into a BDI system (Extended Abstract), Federico Schlesinger, Marcelo Errecalde and Guillermo Aguirre, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 1371-1372

2. THE WADEX FRAMEWORK

We named *WADEX - W*eb Services, *A*rgumentation and (BDI) Ja*dex* - our proposal to integrate the three aforementioned concepts. This proposal includes an *integration architecture* (Section 2.1) and a *general control structure* (Section 2.2). The former, describes how the software components implementing these models interact. The later provides some general guidelines about how the main steps involved in a particular application can be programmed.

2.1 THE INTEGRATION ARCHITECTURE

The integration architecture is depicted in Figure 1. Here, the main implementation resources and technologies used in implementing a concrete agent are shown: JADE and Jadex, the DeLP-Server, WSIG and Web Services in general. Most of the application logic is encapsulated in a Jadex agent, following its BDI model. This agent runs on top of a deployed JADE platform. There are other agents also running on the same JADE platform; they can be of diverse nature and can communicate through messages with the main Jadex agent. One JADE agent with special characteristics is the WSIG's gateway agent: it provides our Jadex agent with an interface to invoke Web services. Besides, our Jade agent can make queries to the DeLP Server. In this case the queries are realized directly as primitive actions of the agent instead of using agent messaging through the JADE platform.

The Jadex reasoning engine was developed with a multiagent context in mind, and therefore message communication with other agents in the platform is very well integrated in its BDI model: Jadex provides facilities for sending and receiving messages from plans, whether in a synchronous or asynchronous way; received messages can cause the adoption of new goals and/or the execution of new plans; etc. The use of WSIG allows to extend these advantages to the use of Web Services as well, as the process of invoking a Web Service becomes almost transparent for the agent.

A web service invocation process starts with a lookup for the service in the *Directory Facilitator* (*DF*). The DF is an special agent of JADE implementing the *yellow pages* ser-

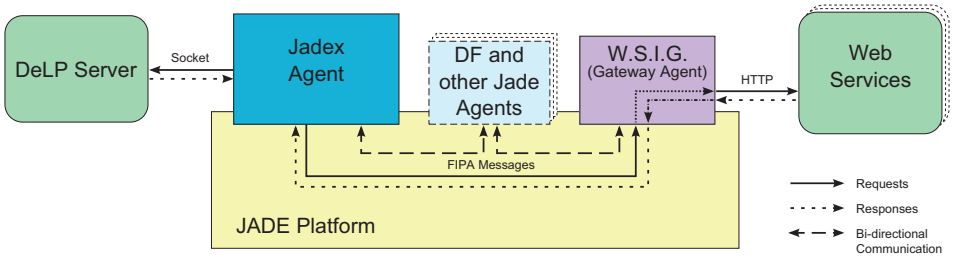

Figure 1: The integration architecture.

vice, *i.e.*, it allows agents in the platform to publish/retrieve descriptions of services that other agents are willing to consume/provide . Once the service has been located, the agent formats a request message in the body of a plan and sends it. Information obtained from the web service invocation can be stored in the agent's beliefbase. It is worth mentioning that the common context of Web Service invocation is from object oriented (OO) programs. Then, having an OO beliefbase (such is the case of Jadex) results in a simpler management of the information.

Due to the simpler nature of the queries made to the DeLP server, we adopted in our proposal the more straightforward approach of sending queries and receiving responses directly using sockets and without any agent messages involved.

2.2 A GENERAL CONTROL STRUCTURE

In the previous section, it was mentioned that all the general application logic is controlled by a Jadex (BDI) agent which acts as an initiator of all the interactions requiring distinct web services and inference processes. Therefore, it is important to analyze the main steps involved in this general control structure.

An alternative that seems to capture the requirements of a significant number of potential applications we have in mind is shown in Figure 2. Here, the control structure of a WADEX agent consists of a main cycle with three generic processes: the *acceptability filter* ($acc\text{-}filt$), the *selection process* ($select$), and the *execution stage* ($exec$).

Let $\mathcal{A}$ be the set of available *alternatives*.
Let $\mathcal{O}$ be the set of acceptable *options*
Let o be the selected *option*
While true do
1. $\mathcal{O} \leftarrow acc\text{-}filt(\mathcal{A})$
2. $o \leftarrow select(\mathcal{O})$
3. $execute(o)$
endWhile

Figure 2: Control structure of a WADEX agent.

The *acceptability filter* is the process in charge of considering the whole set of available alternatives and discarding those that do not satisfy essential requirements to be considered acceptable options. In our implementation, argumentation-based inference is used to make this kind of decisions, after obtaining relevant information from web services.

The *selection process* takes as input the options that survive the acceptability filter and compares them in order to decide which alternative will be selected. In some cases, this process can simply delegate this responsibility to the user, but other more elaborated methods can be used. These methods include voting approaches, argumentation-based defeasible decision making or some hybrid approach which can mix manual and automatic decisions.

Finally, the *execution stage* involves deciding *how* to accomplish the selected option and executing this plan to achieve the desired goals. At the present time, this stage involves selecting and executing plans from a library of pre-compiled plans, following the standard approach adopted in most PRS-based BDI systems. Argumentation-based inference can also be taken advantage of in this stage, to help select among available plans.

A central aspect of BDI architectures is the possibility of reconsidering intentions. In our system, we assume that the execution stage can take certain amount of time during which the world state might change leading to one of two different situations: i) the selected option ceases to be acceptable, or ii) a new option arrives, whose adoption can be considered better than the finalization of the one currently in execution. When a situation like this is detected, the execution stage is interrupted. The the control loop starts over from the acceptability filter or from the execution stage for the execution of the new option, respectively.

The previous paragraphs only give a very simplified and generic view about how the main software components of a WADEX agent interact. A more detailed description of the implementation aspects of WADEX is given in [5] and a good example of how the different components of our framework interact in a real problem (a WADEX-based personal assistant) is presented in [6].

3. REFERENCES

[1] F. L. Bellifemine, G. Caire, and D. Greenwood. *Developing Multi-Agent Systems with JADE*. John Wiley & Sons, NJ, Apr. 2007.

[2] L. Braubach, A. Pokahr, and W. Lamersdorf. Jadex: A bdi agent system combining middleware and reasoning. In *Software Agent-Based Applications, Platforms and Development Kits*, pages 143–168. Birkhaeuser, 2005.

[3] A. J. García, N. D. Rotstein, M. Tucat, and G. R. Simari. An argumentative reasoning service for deliberative agents. In *2nd KSEM*, volume 4798 of *LNCS*, pages 128–139. Springer, 2007.

[4] A. J. García and G. R. Simari. Defeasible logic programming: An argumentative approach. tplp, 2004.

[5] F. Schlesinger, M. Errecalde, and G. Aguirre. The WADEX Framework. Technical report, Universidad Nacional de San Luis, 2009.

[6] F. Schlesinger, E. Ferretti, M. Errecalde, and G. Aguirre. An argumentation-based bdi personal assistant. In *Proceedings of the 23 IEA-AIE*, 2010. Accepted, to be published.