A Proposal for a Equipments Repair System using Agents and Web Services

Alberto Caballero
Havana University (CUJAE)
Faculty of Industrial Engineering
Department of Mathematics and Computer Science
caballero@uci.cu

Jorge Marx Gómez
Technical University of Clausthal
Department of Computer Science
Department of Business Informatics
gomez@informatik.tu-clausthal.de

Abstract

The paper shows the first ideas for a proposal of system using agent and Web Services to suggest to the user the best equipment repair real service. The user obtains a listing with the most suitable real services according to his preferences and requirements. It treats the most important elements considered in the proposal development. This work is not finished yet. I is concerned as a research-in-progress paper.

Keywords: Agents, Enterprise Java Beans, Equipments Repair Service, Framework, Web Service

1 Introduction

The development of systems for the Internet has gone evolving toward the use of Web Services. Web Services are a flexible type of technology, that allows companies to create new products and services faster than with existing methods [1] by dynamically assembling loosely coupled components. This is very different from the traditional approach for developing applications. Fixed application tends to resist change, whereas Web Services assume that change is ever present.

Web Services require research in: explicit representations of e-Services and their capabilities, their re-use in different contexts to form new and dynamic services, the creation of a heterogeneous and competitive environment, reputation networks, negotiation, contracts, etc.

Also, the current users of the Internet demand from the Internet and the system running over it with more autonomy, intelligence, flexibility and pro-activity. Also, it is necessary to face the high degree of distribution of the parts of the system and to achieve the harmonic integration of them by means of cooperation mechanisms, coordination and planning. Agent-based systems and multi-agent systems are systems that try to satisfy these necessities.

In this way, incorporating agents into Web Service-based systems transforms it into an active and flexible environment. It is possible to guarantee the appropriate representation of users' interests, in competitive negotiation environment to achieve the needed e-Service contract [1, 2]

This article describes the first efforts to develop a distributed, intelligent and flexible system for users that need the Equipments Repair Services. In this system the user will be represented by an agent, that is capable to negotiate in order to obtain the best service. In another side, in the system there is a collection of Web Services, which represent the real equipment repair services, and a community of agents that represent each Web Service in the negotiation process with the users' agent. The principal aim of the user's agent is to acquire a listing composed by the most suitable equipment repair services for the user.

In chapter 2, the paper shows the basic system requirements, the used system architecture, and describes the agent roles and their interaction in the negotiation process. Continuing, chapter 3 is about the main characteristics of Web Services in this system. Chapter 4 describes the considered ontology, RepairServiceOntology, used within the agent CNP-interaction to contract the Web Service. Finally, the paper points the conclusions and the perspectives of the unfinished research work.

2 The Proposal

The problem that we are trying to solve here is to find the best services for equipments repair for the user, according to his preferences and under the assumption, that the services will be implemented and accessible as Web Services. These Web Services will analyze the request and the real-world services' characteristics, and offer a proposal according to the user's preferences. The Web Services characteristics will be shown in chapter 3.

A traditional solution would be a solution where the user needs to know all characteristics of each offered services. Easily, the user can decide the most suitable service according to his preferences, through a basic (automatic or manual) analysis.

Our proposal intends to remove this requirement: a user does not need to have previous knowledge about the services and its characteristics. We attempt to show an intelligent and flexible system that considers the user's preferences in each decision made.

According to previous experience [2], we add a layer of agents to guarantee the desired features by means of the negotiation agents' capabilities. In the negotiation process, the agents take into account the user's preferences, the Web Services availability, the characteristics of the real-world services, and the coincidence between them. The prototype system consists of:

- user agents that represent the users' requirements and preferences
- Web Services, each of them implement the mechanism to process the requirements and propose a solution. In this case, the solution represents the expected characteristics of the real services taking into account the given requirement
- service agents that are wrappers for the Web Services.

To specify and develop the agents, we use the FIPA standard. FIPA allow the interoperability between different agents' development frameworks [3]. Among other specifications, there is the standardization of some interaction protocols available: Contract Net Protocol (CNP) [4, 5], Dutch Auction (DA) [6] and English Auction (EA) [7]. In this case, we used the Contract Net Protocol to guide negotiation between user agents and service agents. The sequence diagram can be seen in the CNP standardization document [4].

This community of agents negotiates with the purpose of establishing contract relationships. To achieve an effective recruiting of the services, they are distinguished two agent roles (see figure 1).

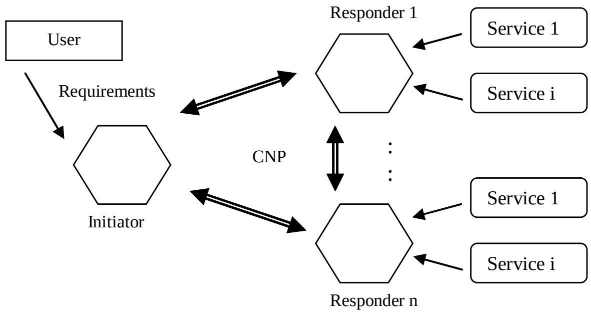

Fig 1: Agent roles interaction

- Initiator Role: Is exhibited by agents that represent the user's preferences and requirements or for those agents that need to subcontract others to achieve its aim. This role takes charge of starting and developing interaction with the rest of the agents.
- Responder Role: Is used by agents that receive the Initiator Role agents' requests. This role receives interaction requests, analyzes them and, if it considers them as interestingly, does its proposal. The proposal will be done according to the characteristics of specialized Web Services.

Following the CNP standard, an agent, that shows the Initiator role, broadcasts a Call-For-Proposal (CFP) message. A Responder agent receives this message and, if it considers it useful, responds with a proposal. The Initiator agent receives proposals from all Responder agents, selects the most suitable and contracts corresponding agents. The content of each CNP message is specified in the ontology RepairServiceOntology, described in chapter 4. In this case, generally, Initiator agents do not know anything about the localization, performance, responsible, provider of Web Services. It only has requirements and preferences and send a CFP messages to the agent community. Each Responder agents analyzes, if it can offer services to satisfy this requirements, replies and negotiates its proposals.

In the moment the Initiator agent broadcasts a CFP message, it should decide to whom it sends the proposal request message. For this purpose, the Initiator agent should possess models for measuring the performance of the other agents. These models must show the performance of each agents according to the user's preferences, requirements, and be organized according to the contracted services up to now.

When Responder agents receive a CFP, it should be able to review all its services and select the suitable one. If neither service was selected, the agent must be able to decide if is necessary to contract another agent. In this case, the agent also has the Initiator role sending a CFP messages.

When the proposal is finally accepted, the Responder agent needs to invoke the corresponding Web Service using the ontology ServiceOntology developed in a previous work [2]. ServiceOntology describes the concepts related to the Web Services location and invocation, and the way to match with the Web Service server technology (in this case, JBoss) using WSDL [8].

3 Web Services

As we have seen, the corresponding agents work as wrappers for the Web Services, belonging to an infrastructure based on J2EE [9]. In this infrastructure we have a distributed application, to which we will superimpose the capacities of flexible invocation of services based on agents. Each one of the Web services is implemented as an Enterprise Java Beans (EJB).

In the following the main attributes and methods' type of the Web Services will be pointed out. These descriptions are in accordance to the real enterprise service.

The Java class that represents a Web Service has:

Public attributes such as:

- object to repair,
- address of the real repair place, telephone, contacts email, etc.,
- time repair average,
- price repair average,
- reliability,
- workers' experience degree,
- post-sale services.

Private attributes such as:

- number of waiting requests,
- number of solved requests,
- availability of materials.

Methods such as:

- those that shows the public attributes,
- those that offers proposals of repair time, price and post-sale services according to the received services,
- auxiliary methods.

4 RepairServiceOntology

The ontology named RepairServiceOntology describes the used concepts and concepts' relationships in the negotiation processes to contract a service. Basically, these concepts are messages' contents exchanged by the Initiator agent and Responder agent in a CNP negotiation. This ontology is implemented in Protégé 2000. Using the package BeanGenerator and the corresponding Java classes were generated. The classes diagram for the ontology RepairServiceOntology is shown in figure 2.

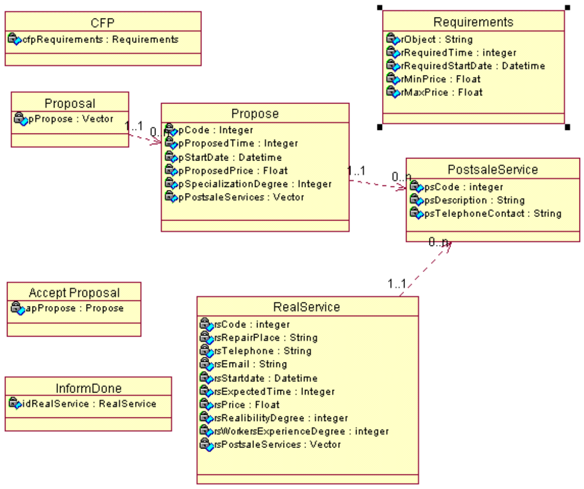

Fig 2: RepairServiceOntology Classes Diagram

The main attribute of the CFP messages is the content given by the user's requirements (Requirements). The users can specify their necessities according to the name of object to repair, the required start date and time of repair, and a price range.

When a Responder agent does a proposal, it fills a list with all proposals, where each of them is one instance of the Propose class. A proposal is given by an identifier, proposed time, start date and price, the workers' experience degree and a set of post-sale services. Each post-sale service has a code, a description and telephone number for possible contacts.

An initiator agent analyzes all proposals, decides which is the most suitable one and sends one instance of class Propose as a messages AcceptProposal.

The Responder agent receives the AcceptProposal message and runs the corresponding Web Services. When the Web Services execution is finished, the agent takes the result and initiates an puts an instance of the class RealService into anInformDone message. In RealService the agent informs about the characteristics of the most suitable real service according to the user requirements including:

- code,
- repair place,
- contact telephone and email,
- start date,
- expected time,
- price,
- service reliability degree,
- workers' experience degree and a set of post-sale services.

5 Conclusions and Future Work

This paper shows another example of integration of agents and Web Services into flexible and intelligent applications for the Internet. In this way, it validates the initial proposal we developed and published in other previous papers [2, 10]. In a further step the approach has to be implemented, to be analyzed and to be tested emphasizing in the learning models.

References

[1] Burg, Bernand. Agents in the world of active Web-services. Springer's Lecture Note in Computer Science, March 2002.

[2] Alberto Caballero, Juan A. Botía, and Antonio Gómez-Skarmeta. El papel de los agentes FIPA en aplicaciones basadas en servicios Web. Appear in Proceedings of the Congreso de la Asociación Española de Inteligencia Artificial, November 2003.

[3] Foundation for Intelligent Physical Agents. FIPA Abstract Architecture Specification. SC00001, 2002.

[4] Foundation for Intelligent Physical Agents. FIPA Contract Net Interaction Protocol Specification. SC00029, 2002.

[5] Reid R. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. In A. H. Bond and L. Gasser, editors, Readings in Distributed Artificial Intelligence, pages 357–366. Morgan Kaufmann Publishers, Los Altos, CA, 1988.

[6] Foundation for Intelligent Physical Agents. FIPA Dutch Auction Interaction Protocol Specification. XC00032, 2001.

[7] Foundation for Intelligent Physical Agents. FIPA English Auction Interaction Protocol Specification. XC00031, 2001.

[8] E.Christensen, F. Cubera, G. Meredith, and S. Weerawarana. Web Services description language (wsdl) 1.1. Technical report, W3 Consortium, 2001.

[9] Sun Microsystems, Inc. JavaTM 2 Platform, Enterprise Edition (J2EETM), 2002.

[10] A. Caballero, J. Marx Gómez, SAMA: A Prototype System for Management of Devices, Services and Applications in Enterprise's Networks based on a JMX Agent Model.