Semantic Web Enabled Virtual Travel Agency

Swati Ringe¹, Dhana Nandini², Gaurav Sharma³, Mayuri Rege⁴

Abstract—The existing Virtual Travel Agencies (VTA) are mostly web applications maintaining their own databases. Through data entered by all its partnering travel websites and businesses, the VTA’s database is thoroughly searched on the basis of combining all the requirements of the traveller. There is no possibility of service provider getting connected to the client via his own databases or web service. The reason behind this is the variation in data formats used in their databases which makes it a difficult job to integrate their services together to provide a common window for accomplishing the user’s requirements. Our project gives a solution to the above mentioned problems by implementing a VTA that performs semantic-mapping via ontology and multi-agents.

Index Terms—AMS (Agent Management System), DF (Directory Facilitator), JADE (Java Agent Development Environment), MAS (Multi-Agent System), SW (Semantic Web), SPA (service provider agents), VTA (Virtual Travel Agencies).

1 INTRODUCTION

Semantic Web (SW) is a vision of web in which the computer programs will be interpreted not only by humans but also by machines. Semantic Web is not a separate Web but an extension of the current one, in which information is given well defined meaning, better enabling computers and people to work in cooperation (Berners-Lee et al 2001). Web Services are server-based applications which may be accessed over the web via HTTP, but is meant primarily for interaction with other programs. Once a web Service is deployed other applications can discover and invoke the deployed service. To make services ubiquitously available we need a semantic-based approach such that applications can reason about a service’s capability to a level of detail that permits their discovery, integration and composition. For discovery, composition, etc., one could take the syntactic approach in which the services being sought in response to a query simply have their inputs syntactically matched to those of the query. But alternatively, one could take a semantic approach in which the semantics of input output parameters are considered. In this paper we present an approach for implementing semantics through Semantic web enabled services using Multi-Agent-Systems and Ontology and also provide a practical solutions to the discovery, integration, scalability and reliability problems of web-service.

2 MULTI-AGENTS

2.1 AGENTS

Agents are programs acting on behalf of another person, an entity or a process. Intelligent agents are widely known and useful for application automation and Internet commerce (Hendler 2001; Blake 2001).

For example, Hendler (2001) indicated that ontology-based intelligent agents could obviously enhance application integration and thus improve Internet commerce. In short Agents are independent entities.

Some of the properties of Agents that makes it important from implementation point of view are that agents
• are not strictly invoked for a task, but activate themselves.
• do not require interaction of user.
• may invoke other tasks including communication.

3 MULTI-AGENT SYSTEM

A Multi-Agent System (MAS) consists of a team or organization of software agents, collectively performing a task which could not be performed by any individual agent. [3]

ONTOLOGY: An ontology is a specification of a conceptualization that is designed for reuse across multiple applications and implementations, a specification of a conceptualization is a written, formal description of a set of concepts and relationships in a domain of interest.

A distinguishing characteristic of ontologies compared to conventional hierarchical structures is their degree of connectedness, their ability to model coherent, linked relationships. [4]

Fig. 1. Hotel

Prof. Swati Ringe is currently working as an assistant Professor in Computer Engineering Department, Fr.C.R.C.E. Bandra(w),Mumbai-50, India E-mail: swati@fragnel.edu.in
Dhana Nandini, Gaurav Sharma, Mayuri Rege are the final year students in Computer Engineering Department, Fr.C.R.C.E. Bandra(w),Mumbai-50, India E-mail: nandini.rajaruban@gmail.com², gauravsh34@gmail.com³, mayuri_rege6@yahoo.com⁴
4 VIRTUAL TRAVEL AGENCY

A Web Service of a Virtual Travel Agency, short: VTA, offers end-user services for searching and booking hotel and flight tickets. This Web Service is composed out of other Web Services, namely accommodation and transportation services that are provided and published by different companies and are registered with the VTA.

Fig. 2. Block diagram of Virtual Travel Agency (VTA)

5 OVERVIEW

In this model, if required, the input request from User Agent is directly decomposed by the system into atomic task/activities Task1, Task2, Task3 ... Task n, based on its ontological description. After that, for each of the atomic task, the candidate software agents, who are acting as semantic web service components, are discovered and finally the correct one is selected and the task gets assigned. SPA1, SPA2, SPA3 ... SPA n are the selected service provider agents (SPA) for tasks Task1, Task2, Task3 ... Task n respectively. The user agent, now negotiate with each of the SPA and assign it the respective task. The arrangement for the negotiation can take place using FIPA Contract Net Protocol and SPA can accept task by means of agent’s communication interface built upon FIPA-ACL. [1]

6 ARCHITECTURE

Process

Step 1.
The user places his request from the JSP page.
Step 2.
The servlet grabs the request parameters and give them to the Gateway Agent.
Step 3.
The Gateway Agent now refers to the Ontological Description to get the synonyms of the primary parameters and using DF(Directory Facilitator).Now using DF, it will find the appropriate service provider agents.
Step 4.
Now Gateway Agent will encode the data in the form of ontology
And pass it to the appropriate selected service provider agents.
Step 5.
The service provider agent on receiving the ontological data will decode it to its native data format.
Step 6.
Service Provider Agent will now forward the data to the web service
Step 7.
The web service will retrieve the requested data from the database and return it to the service provider agent.

**Step 8.**
The service provider agent will then encode the data in the form of ontology and return it to the calling gateway agent.

**Step 9.**
Now the gateway agent will decode the data and give it to the servlet.

**Step 10.**
The servlet then displays the result in a new JSP page to the user.

### 6.1 JADE TECHNOLOGY TO IMPLEMENT MULTI-AGENT-SYSTEMS

JADE (Java Agent Development Environment) is a middleware that facilitates the development of multi-agent systems. It includes:

1. A runtime environment where JADE agents can “live” and that must be active on a given host before one or more agents can be executed on that host.
2. A library of classes that programmers have to/can use (directly or by specializing them) to develop their agents.
3. A suite of graphical tools that allows administrating and monitoring the activity of running agents.

Each running instance of the JADE runtime environment is called a Container as it can contain several agents. The set of active containers is called a Platform.

A single special *Main container* must always be active in a platform and all other containers register with it as soon as they start. Besides the ability of accepting registrations from other containers, a main container differs from normal containers as it holds two special agents.

The **AMS** (Agent Management System) that provides the naming service (i.e. ensures that each agent in the platform has a unique name) and represents the authority in the platform (for instance it is possible to create/kill agents on remote containers by requesting that to the AMS).

The **DF** (Directory Facilitator) that provides a Yellow Pages service by means of which an agent can find other agents providing the services he requires in order to achieve his goals. Using the Yellow Pages service provided by the DF agent When an agent A communicates with another agent B, a certain amount of information I is transferred from A to B by means of an ACL message. Inside the ACL message, I is represented as a content expression consistent with a proper content language and encoded in a proper format. Both A and B have their own way of internally representing I. Hence it's clear that each time agent A sends a piece of information I to agent B,

1) A needs to convert his internal representation of I into the corresponding ACL content expression representation and B needs to perform the opposite conversion.

2) Moreover B should also perform a number of semantic checks to verify that I is a meaningful piece of information. The support for content languages and ontologies provided by JADE is designed to automatically perform all the above conversion and check operations.[6]

Exploiting the JADE content language and ontology support included in the jade.content package to make agents talk and reason about “things and facts”.

The Ontology here is organized using three Schemas-  
1) Predicate Schema  
2) Concept Schema  
3) AgentAction Schema

We define an ontology including the schemas for the types of predicate, agent action and concept that are pertinent to the addressed domain. Developing proper Java classes for all types of predicate, agent action and concept in the ontology will complete the part syntax definition that will be referred by all the agents working in a multi-agent-system.[6]

### 7 ADVANTAGES VS EXISTING SYSTEMS

#### The Discovery Problem:

Given a repository of Web services, and a query requesting a service, automatically finding a service from the repository that matches the query requirements is the Web service Discovery problem.

This paper provides solution to the discovery problem in the following way—For instance, consider that the user is giving the input parameters to search for a hotel in a location, say, Mumbai. Now the request is forwarded to the Gateway Agent which is with the VTA. The Gateway Agent now refers to the Ontological Description to get the synonyms of the primary parameter which is with the VTA. The Gateway Agent now refers to the Ontological Description to get the synonyms of the primary parameter which in this case is Hotel. Then the Gateway Agent will search in its DF Register to find the appropriate Service Provider Agents and selects them which in turn invokes the appropriate web services. Now, matching of secondary parameters with the returned results take place and the relevant outputs are provided to the user. [2]

#### Scalability:

The architecture is scalable as new service provider agents and gateway agents can easily be added without affecting the performance.

#### Reliability:

Agents are independent entities. Hence, even if a particular service provider agent fails, the system will still work as a
whole because the other service provider agents will still be active.

Integration:
The architecture combines different service provider agents to achieve a complex task.

8 CONCLUSION
The Agents are assigned the task of finding an appropriate Server Agent to invoke the service depending upon the parameters that has been provided by the Client-end, thus avoiding the static binding and invocation problem found in web-services and providing partially, a solution to the discovery problem which is mentioned above. Agents being independent entities make it an easier task to impart the system scalability and to some extent proving it to be a bit more reliable as well. Thus one of the many solutions that has been proposed to implement a Semantic Web has been implemented in our project.

REFERENCES
[1] Sandeep Kumar, R.B.Mishra. Multi-Agent Based Semantic Web Service Composition Models