24

# WEB SERVICES / AGENT-BASED MODEL FOR INTER-ENTERPRISE COLLABORATION


Akbar Siami Namin and Weiming Shen
*National Research Council of Canada*
*{akbar.siami, weiming.shen}@nrc-cnrc.gc.ca*

Hamada Ghenniwa
*The University of Western Ontario*
*hghenniwa@eng.uwo.ca*
*CANADA*



*Web Services technology is a promising computing paradigm for applications integration over the Internet. Use of Web services and related technologies facilitates the implementation of virtual enterprises across heterogeneous hardware and software platforms. This paper proposes a Web services / agent-based model for inter-enterprise collaboration. It presents a multi-agent model in different levels of the enterprise's system architecture to accomplish a suitable selection of a registered service, to check the status of a process, to realize users' requests, and to react to them in a collaborative way with other agent-based Web services. Moreover, the paper proposes a multi-agent model to define a dynamic workflow capable of coordinating and monitoring the workflow processes.*


## 1. INTRODUCTION

Global competition has forced manufacturing enterprises, particularly SMEs (Small and Medium-sized Enterprises), to increase their productivity and profitability through optimal resource utilization. On the other hand, changing customer demands and manufacturing environments make resource utilization more and more unpredictable and unstable. Conventional congregations of enterprises operating together try to solve the above problem by production outsourcing to achieve maximal group benefits. The advantages of Virtual Enterprises enabled by information and communication technology provide new ways to facilitate inter-enterprise manufacturing resource sharing, and therefore improve the profitability of SMEs (Camarinha-Matos and Afsarmanesh, 1999). However, implementation of virtual enterprises is not an easy job, since it is usually related the integration of hardware and software environments as well as serious privacy and security concerns.

Web services technology is a promising computing paradigm for integrating legacy applications over the Internet. Using Web services and the related standards facilitates the implementation of virtual enterprises in heterogeneous hardware and software platforms. It can also address well the privacy and security issues. In particular, it has following important advantages:

- Platform Independency: Web services technology provides communication among participants at the application level. Enterprises will be able to maintain their heterogonous legacy systems while seamlessly sharing these resources in a virtual homogeneous environment;
- Loosely Coupled Components: Web services can be modified, replaced, and removed with minimum or without affecting on the collaboration;
- Service Registry: Enterprises are able to advertise their services effectively through "*Universal Description, Discovery, and Integration* (UDDI)" registry. It also enhances both the creation and evolution levels of a virtual enterprise's life cycle;
- System Modularity/Reusability: Methods and data in a Web service can be reused in several applications regardless of their platforms.

However, Web services do not provide a complete solution for virtual enterprises or enterprise collaborations. In particular, it does not support fully automatic and dynamic collaborations among enterprises.

Software agents have emerged as a promising technology for dealing with cooperation and decision-making in distributed applications. Agent based manufacturing has become a new paradigm for next generation manufacturing systems, together with other manufacturing paradigms such as Holonic Manufacturing Systems, Agile Manufacturing, Reconfigurable Manufacturing, etc. (Shen et al., 2001).

Software agents have been developed with sophisticated interaction patterns. They are efficient in enforcing automatic and dynamic collaborations. Agent-orientation is an appropriate design paradigm for e-Business systems with complex and distributed transactions, especially for Web services. In services realization, software agents are very instrumental to provide a focused and cohesive set of active service capabilities (Li et al., 2004)

Software agents can be considered as an appropriate paradigm to overcome some shortcomings of Web services for enterprise collaborations:

- A Web service is just a self-describing software component such that it does not have enough knowledge about its environment, users, software components, and outside world in general. In contrast, software agents are capable of reasoning, and interacting with other entities;
- Web services are discoverable by XML-based UDDI standard. Current standard of the UDDI is only able to recognize terms "syntactically". The main challenge in service discovery is how to find services, which are "semantically" the same as clients' desires. Software agents operate at the knowledge level, at which they are able to reason semantically on the service requesters.

This paper proposes a Web services / agent based model for collaborative virtual enterprises. We discuss the integration of software agents and Web services as a suitable solution for setting up a virtual enterprise. The rest of the paper is structured as follows. Section 2 reviews the related work. Sections 3, 4, and 5 present a Web services / agent-based architecture, enterprise model, and UDDI model for inter-enterprise collaboration, respectively. Section 6 discusses some implementation issues of developed prototype. Section 7 provides some conclusions and discusses the future work.

## 2. RELATED WORK

Virtual enterprises based on software agents and Web services have a growing appeal. There have been significant research efforts to integrate software agents in a VE (Rabelo et al., 2001; Marik and Pechoucek, 2003).

The use of agents in Workflow Management Systems (WfMS), has been discussed in several papers (Yan et al., 2001). In an approach presented by Chang and Scott (1996), each workflow has been represented by multiple agents as personal, actor, and authorization agents. These agents perform actions on behalf of the workflow participants and facilitate interaction with other participants or organizations.

In ADEPT (Jennings et al., 1996), the multi-agent architecture consists of a number of autonomous agencies. A single agency consists of a set of subsidiary agencies, which are controlled by one responsible agent. Each agent is able to perform one or more services. None of these attempts adopts agent technology to compose workflow execution engine dynamically. The logic for workflow processing is hard-coded and thus it is hard to reuse the workflow execution engine for other business process.

The "Shadow Board Agent" architecture by orchestrating multiple Web services in an agent based transaction model was proposed in (Jin and Goschnick, 2003). Each participant is wrapped as a Web service and uses an agent-oriented approach to engineer each Web service as a software agent. Despite using agent-based architecture, the model lacks initial defining of workflows, monitoring mechanism, and appropriate rating mechanism.

Interleaving Web services composition and execution, using software agents and delegation have been discussed in (Maamar et al., 2003). Although the approach was based on software agent for Web services composition, however the model mostly focuses on the selection of services involved in the composition rather than composition itself.

A conceptual model for Web service reputation has been proposed in (Maximilein and Singh, 2001). In this model, There has been defined a "*Web Service Agent Proxy* (WSAP)" to access each service. A WSAP is an agent that acts as a proxy for clients of Web services.

## 3. A WEB SERVICES / AGENT-BASED MODEL

In this section, we propose a service oriented / multi-agent based model for virtual enterprises. We focus on integrating Web services and software agents inside the internal structure of a typical enterprise as well as adopting software agents inside the UDDI registry. On the enterprise side, we propose a goal-based model to define dynamic workflows. Also we introduce other agent-based components for coordination and monitoring purposes. On the UDDI side, we introduce some agent-based components to assist service requesters towards choosing the most suitable service provider. The proposed architecture is depicted in Figure 1.

On the enterprise side, A "*Proxy Agents Layer (PAL)*" is defined, which is sited on the top of other software units. PAL acts as an interface between the enterprise and outside world. A corresponding interface is defined for the UDDI server as well. The interface is named as "*Discovery Agent Layer (DAL)*", which is sited on the top of the UDDI server.

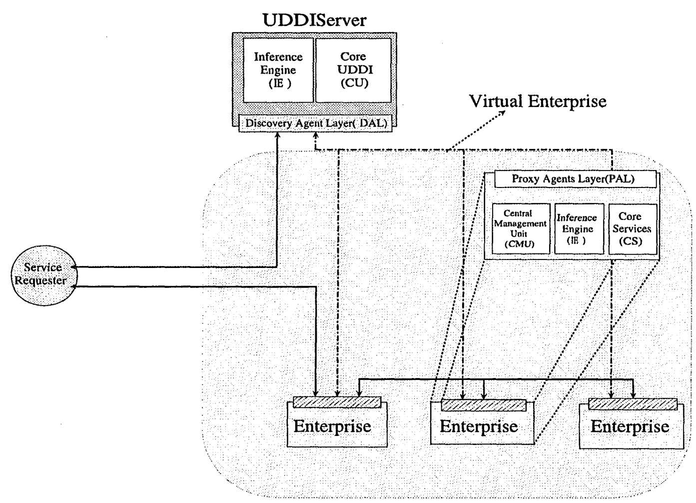


Figure 1 – A Web services/multi-agents based model for enterprise collaboration

As a simple scenario, a service requester (as a client), asks the UDDI server through the "Discovery Agent Layer (DAL)" to assist in locating a "suitable" and "reliable" service provider. Consequently DAL, in collaboration with some other components that have been defined in the UDDI server, presents the "recommended" service providers to the requester. Then, the requester, by choosing a suitable service provider, interacts with the provider (enterprise) through its "Proxy Agents Layer (PAL)".

## 4. ENTERPRISE MODEL

The detailed model of a virtual enterprise is depicted in Figure 2. All interactions between an enterprise and outside world are carried out by the "Proxy Agent Layer (PAL)". The "*Core Services (CS)*" unit consists of two components: Web services and corresponding database. The definition and functionalities of the Core Services (CS) are the same as current paradigm of Web services including the enterprise database. The other two units, namely, "*Central Management Unit (CMU)*" and "*Inference Engine (IE)*" are responsible for defining, managing workflows and knowledge representation. The agentified Web services will run on the service provider's side and do not affect the traffic of the network. The defined local agents communicate with the "Proxy Agents Layer (PAL)" to find out the existing Web services. Hence, among defined local agents, only PAL and Ontology Agent (OA) have knowledge about the Web services. The knowledge is acquired through communicating with the Web services.

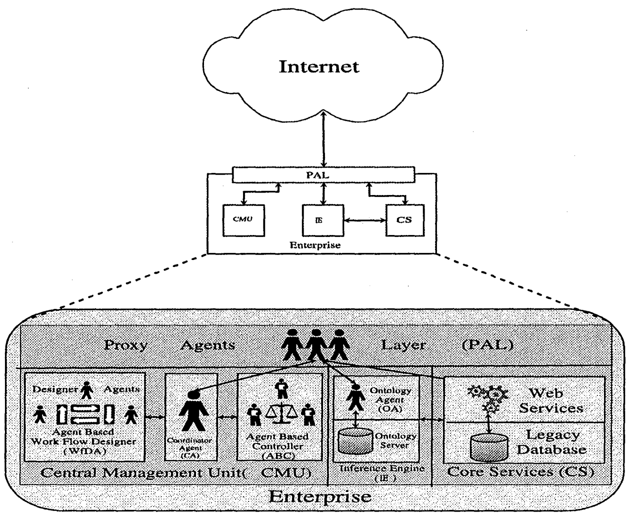


Figure 2 – A Web services / agent-oriented architecture for an enterprise

## 4.1 Central Management Unit (CMU)

Central Management Unit (CMU) is an "organizer" component with direct contact with the PAL. The CMU is responsible for:

- Accepting requests from PAL to design workflows;
- Informing PAL to locate suitable service providers and delegate the tasks of workflows to them;
- Assigning the tasks to the qualified enterprises;
- Coordinating the workflow amongst enterprises;
- Monitoring the progress and status of the delegated tasks.

CMU is responsible for most of the activities required at the creation and evaluation stage of a typical virtual enterprise's life cycle. These activities include designing workflows, partner searching, agreement, and monitoring the status of the delegated tasks.

### *4.1.1 Workflow Designer Agents (WfDA)*

A workflow can be considered as a goal or set of goals. In other words, a service consumer is concerned about achieving the service; rather than "how" to gain it. To this extent, we let the defined agent-based units design the appropriate sub goals in runtime.

The most important activity of WfDA is to design a "cross-enterprise workflow specification" at runtime, by which it is possible to create a dynamic, not only in specification but also in assigning tasks to enterprises. The start point in WfDA is to achieve a request, or goal that has been asked by a service consumer interested in sharing resources as services and creating a virtual enterprise.

*4.1.2 Coordination Agent (CA)*

The Coordination Agent (CA) carries out all interactions between CMU and PAL. CA is responsible for coordinating and managing the processes of a workflow, which have been designed at WfDA. The main duties of CA are as follows:

- Communicating with PAL;
- Interacting with WfDA;
- Coordinating the involved enterprises;
- Communicating with Agent-Based Controller (ABC) by (1) informing ABC to create a corresponding controller agent in order to compare the plan and progress; and (2) receiving information from ABC, regarding the progress of the delegated tasks.

*4.1.3 Agent-Based Controller (ABC)*

In our model, after assigning each task to an individual enterprise, CA informs ABC to create a corresponding "controller agent" and monitor the progress and status of the underlined process, which is carried out by the enterprise. By getting a feedback from the enterprise, the controller agent compares the results of the enterprise's activities with the plan and informs CA to make a decision.

### 4.2 Inference Engine (IE)

In the current technology of Web services, requesters, by acquiring some meta-data from WSDL, realize how to exchange business data with the underlined Web services. These kinds of syntactically "invoking" services cannot cover "requesting" based on semantics. In fact, an enterprise needs an intelligent component such as an "*Ontology Agent (OA)*" to realize and discover the exchanged messages and map them to existing services. The structure of IE is similar to defined inference engine in UDDI server and we describe it in detail in the following sections.

### 4.3 Proxy Agents Layer (PAL)

The Proxy Agents Layer (PAL) can be considered as a complementary component for Web services technology in order to change a passive enterprise to a proactive entity capable of involving in transactions proactively. All communications between an enterprise and its outside world will go through the PAL. Moreover, PAL is responsible for exchanging data among internal components (CMU, IE, and CS) of an enterprise. From another point of view, PAL can be considered as a wrapper, by which the functionalities and complexities of the internal structure of an enterprise are encapsulated from outside visions.

PAL must be able to realize the format of current set of Web services standards such as: SOAP, WSDL, and UDDI. The main responsibilities of PAL include:

- Routing all incoming or outgoing messages to suitable software components either inside or outside the enterprise. The interaction can be:
  1) Receiving a request for an available existing service;
  2) Communicating with the Inference Engine (IE) in order to realize any ambiguity about the meaning of the used terminology;

3) Communicating with the Central Management Unit (CMU) to decompose the requested service.

- Contacting with the UDDI server in order to either look for an enterprise or report the quality of used services.

# 5. UDDI SERVER MODEL

The proposed architecture for a UDDI server is depicted in Figure 3. All interactions between a UDDI server and its outside world are carried out by the Discovery Agent Layer (DAL). DAL acts as an interface that accepts requests from outside world and analyzes them by collaborating with other defined internal components. In the following subsections, we describe each internal component in detail.

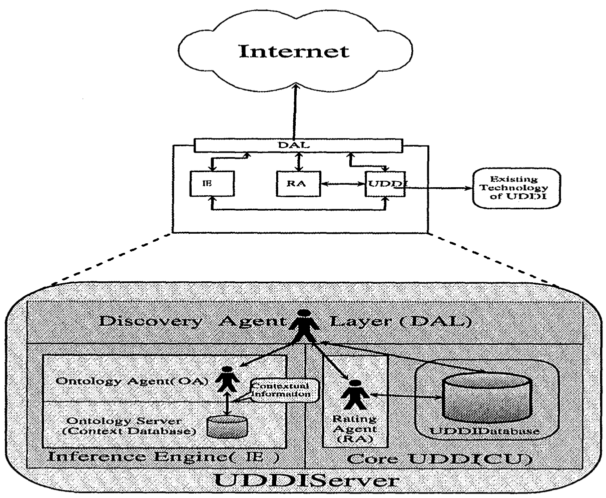


Figure 3 – An agent-oriented UDDI architecture

## 5.1 Discovery Agent Layer (DAL)

The Discovery Agent Layer (DAL) acts as an interface between the internal components of the UDDI server and the outside world. DAL can behave like a router to transfer incoming data to a suitable component located inside the UDDI server. DAL should be able to realize Web services standards such as UDDI, WSDL, and SOAP. The main responsibilities of DAL include:

- Routing incoming and outgoing messages to a suitable internal component or outside world;
- Exchanging information between the internal components of the UDDI server;
- Assisting the service requesters in finding suitable service providers by locating and suggesting the most qualified and matched services;
- Managing all received information regarding performances of an enterprise and updating the qualification of service providers' profiles.

### 5.2 Inference Engine (IE)

In our model, the ontology related components include an agent-based entity called "Ontology Agent (OA)" and a knowledge base component called "Ontology server", which is responsible for knowledge representation.

The Ontology Agent (OA), accepts the terminologies that have been used in the request, "interprets" them into a specific format, known as "context information", and submits the interpreted data to the ontology server to "process". The process on the context information is carried out by some "classifications" of some "context types".

Eventually, OA sends back to DAL the meaning of the terms with an understandable format to search at the Core UDDI.

### 5.3 Core UDDI (CU)

We consider two subunits inside the Core UD (CU): the UDDI-Database and the Rating Agent (RA). The UDDI database is the same as the current technology of the UDDI registry.

The RA provides some information in terms of qualification of services for assisting service requesters in choosing the most reliable enterprises. RA, by expanding the UDDI's knowledge of existing services, evaluates the quality of underlined services. RA updates the rating data by considering some measures such as availability, performance, reliability, and response time. The rating information is accessible for any service requester as public information.

## 6. PROTOTYPE IMPLEMENTATION

In order to prove the feasibility of the proposed model, a simple prototype has been implemented. The prototype is a simplified distributed system, which represents integration and resource sharing through a cooperative distributed system. It consists of following entities (Figure 4):

- Three enterprises as service providers capable of providing some services or resources;
- An agent based Web portal behaving as a gateway through which end users send their requests to the registered Web services.

The prototype is developed using popular Web programming tools and languages under the Windows NT/2000 environment. Java API for XML-based Remote Procedural Call known as JAX-RPC has been used to create and deploy all Web services.

As a simple scenario, there are three companies that possess some expensive machines offered as services to customers. A service requester, interested in using these machines through communicating with the Web portal, asks for the bids. Consequently, the Web portal, which has knowledge about the services and their providers, sends a "*Simple Object Access Protocol (SOAP)*" message ("*Call for Bids*") to enterprises. Thus, any service provider based on their facilities and availabilities proposes its bid to the Web portal. The Web portal, by analyzing the received bids, suggests the most suitable service to the customer.

Each service provider has its own databases, services, and configuration files, which protects the privacy of their internal resources. The underlined Web services are deployed on the Apache Tomcat Web container. The Web portal is created by Java Servlets, deployed on Apache Tomcat Web container as well. Figure 5 shows two snapshots of the implemented prototype.

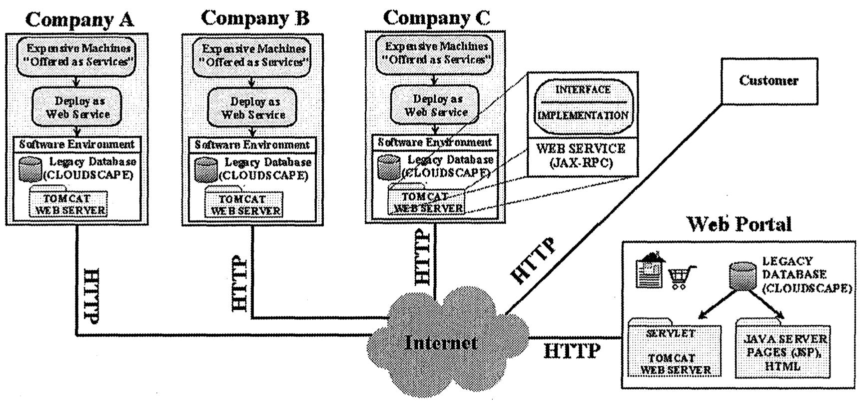


Figure 4 – The manufacturing resources sharing scenario

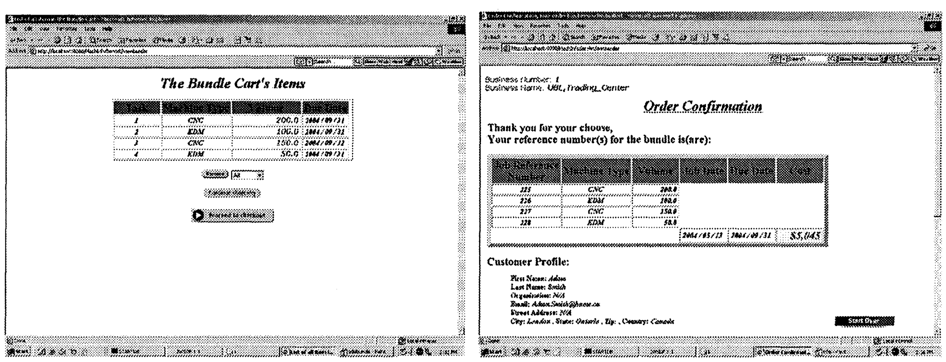


Figure 5 – Snapshots of the implemented prototype

## 7. CONCLUSION AND FUTURE CONTRIBUTION

In order to carry out customers' requests, enterprises need to collaborate with each other and share their resources. Enterprise collaboration, permanent or temporary, requests for a higher level of technology, which allows enterprises to integrate their applications regardless of platforms, data structures, or models. In this paper, we discuss the feasibility of using software agents and Web services technology for adopting into a collaborative environment. The paper proposes a Web services / agent-based approach towards setting up a distributed environment for inter-enterprise collaborations.

On the UDDI side, the paper defines some components such as Discovery Agent Layer (DAL) as an interface between the UDDI server and its outside world, Inference Engine (IE) responsible for realizing user's requests semantically, and

Core UDDI (CU), which consists of some components including the current technology of the UDDI.

On the enterprise side, some key components have been defined such as Central Management Unit (CMU) to define a dynamic workflow, Proxy Agents Layer (PAL) as an interface between the service and outside world, Inference Engine (IE) responsible for analyzing the incoming requests semantically, and Core Services (CS) including the current technology of Web services. From the implementation point of view regarding the complexity of workflows, the proposed model is believed to be feasible. The workflows are distributed among participants and consequently each participant takes its own responsibility of designing its delegated workflow and each workflow is viewed as a service that is offered by a provider.

As the future work the following challenges are to be addressed:

- Creating a Standard Ontology Server: In order to realize and offer services to end users, a Web service or the UDDI server should have a unified, standard, and complete ontology server, by which ontology agent interprets and reasons about the user requirements.
- Creating a Genetic Algorithm for Rating: The rating algorithm should be applicable for different domains and gives a standard and unified solution for different UDDI servers. Moreover, the algorithm should be dynamic enough to accept user-defined attributes for rating and taking them into account.
- Dealing with Security Issues: Without secure transactions, enterprises may not involve in any kind of business. The Proxy Agents Layer should be intelligent enough to prevent malicious requests to get inside the enterprise.

## REFERENCES


1. Camarinha-Matos, L.M. and Afsarmanesh, H. Infrastructure for Virtual Enterprise, Kluwer Academic Publishers, Norwell, Massachusetts, 1999.
2. Chang, J. and Scott, C. "Agent-based workflow: TRP support Environment (TSE)", Computer Networks and ISDN Systems, pp 1501-1511, 1996.
3. Jennings, N. R., Faratin, P., Johnson, M.J., Norman, T.J., O'Brien, P. and Wiegand, M.E. "Agent-based business process management", International Journal of Cooperative Information Systems, pp. 105-130, 1996.
4. Jin, T. and Goschnick, S. "Utilizing Web Services in an Agent Based Transaction Model (ABT)", Workshop on Web Services and Agent-based Engineering, Melbourne, Australia, 2003.
5. Li, Y., Ghenniwa, H. and Shen, W. "Agent-Based Web Services Framework and Development Environment", to appear in Computational Intelligence, November 2004.
6. Maamar, A., Sheng, Q.Z. and Benatallah, B. "Interleaving Web Services Composition and Execution Using Software Agents and Delegation", AAMAS'03 Workshop on Web Services and Agent-based Engineering, 2003.
7. Marik, V. and Pechoucek, M., Agent Technology for Virtual Organization, in L. M. Camarinha-Matos, H. Afsarmanesh, (ed.), Processes and Foundations for Virtual Organizations, Kluwer Academic Publishers, pp. 243-252, 2003.
8. Maximilein, M. and Singh, M.P. "Reputation and Endorsement for Web Services", ACM SIGecom Exchange vol. 3, no. 1, pp. 24-31, 2001.
9. Rabelo, R.J., Afsarmanesh, H. and Camarinha-Matos, L.M. "Federated multi-agent scheduling in virtual enterprises", in Proceedings of PRO-VE'01, pp.145-156, 2001.
10. Shen, W., Norrie, D.H. and Barthes, J.P. Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing, Taylor and Francis, London, UK, 2001.
11. Yan, Y., Maamar, Z. and Shen, W. "Integration of Workflow and Agent Technology for Business Process Management", Proc. of CSCWD 2001, London, Ontario, Canada, pp. 420-426, July 2001.