Towards Web Services Oriented Agent-Based Manufacturing

Akbar Siami Namin[1] and Weiming Shen[2]

Integrated Manufacturing Technologies Institute
National Research Council of Canada
London, Ontario, Canada

ABSTRACT

The revolutionary approach of manufacturing systems design is changing from an isolated island of resources and data to a dynamic collaborative alliance of companies best known as a virtual enterprise. The virtual enterprise can be considered as a scenario that emerges in a world where individual companies come together as a coalition to fulfill the market requirements. Web Services technology is an emerging paradigm in establishing a standard environment for legacy applications integration over the Internet. Using Web Services and related standards has made it a best choice to implement a virtual enterprise in heterogeneous software platforms. In this paper we propose to integrate Virtual Enterprise, Web Services, and Software Agents technologies, to realize inter-enterprise collaborations while maintaining privacy and security. We believe that the combination of the paradigm of virtual enterprise, the standards of Web services, and the unique characteristics of intelligent software agents, is the best answer to future Web-based manufacturing. We define an architecture consisting of Web services and intelligent software agents and discuss how this combination has the ability to reason and change its static policies to a dynamic collaboration.

1. INTRODUCTION

The business world pushes enterprises to go beyond an isolated island of resources to a dynamic collaborative alliance of enterprises and carry out market demands toward satisfying customers. In order to survive and competition, enterprises need to join and collaborate together and form a virtual coalition and provide new kinds of services. Enterprise coalition, either stable or transient, known as a virtual enterprise [1], helps companies sharing their resources such as technologies, data, skills, and facilities and provides a new type of product or service, which is beyond the capability of any individual member of the alliance. However, prior to any operational stage, forming a virtual enterprise requires some vital prerequisites to be set.

The Internet and electronic business have affected the market deeply in such a way that economic growth is impressed based on accessibility of resources over the Internet. Web services technology is an emerging paradigm in establishing a standard environment for legacy applications integration over the Internet. Using Web services and the related standards facilitate integration in heterogeneous software platforms and providing a homogeneous environment for a virtual enterprise. Web services technology as new paradigm in distributed computing provides some solutions to address the manufacturing challenges and meet the requirements of establishing a virtual enterprise.

In this paper, we propose a Web services oriented agent-based approach for collaborative manufacturing. The approach considers a Web service component as an agent-based entity, which represents a service or resource. The rest of the paper is structured as follows: Section 2 clarifies the exiting challenges regarding collaborative environment; Section 3 reviews the literature and related work based on Web services and software agents for adopting in manufacturing and Web-based collaboration; Section 4 compares Web services and software agent technologies in order to find out their common properties; Our Web services oriented agent-base approach is proposed in Section 5; Section 6 presents the prototype architecture and implementation; A simple scenario for resource sharing via our model is described in Section 7; and Section 8 provides conclusion and future contributions.

[1] A graduate student in Computer Science department at Lakehead university and a guest worker at National Research Council Canada's Integrated Manufacturing Technologies Institute; E-mail:akbar.siami@nrc.gc.ca

[2] Corresponding Author: Tel.: (519) 430-7134; Fax: (519) 430-7064; E-mail: weiming.shen@nrc.gc.ca

2. MOTIVATIONS

Although Virtual Enterprise (VE) seems to be a suitable solution for today's businesses, setting up and establishing a typical VE is not straightforward. Collaboration among enterprises offers some challenges:

Diversity of Platforms. Resources, based on their properties, consist of a large variety of categories such as hardware, software, data, and methods, which are operating on their own legacy systems. In fact, heterogeneous system application suffers from lacking a flexible communication mechanism and exchanging information among enterprises involved in a diverse collaborative environment.

Web services technology provides communication at the application level, by which enterprises are able to keep their legacy systems developed by any language on any platform and share them throughout a heterogeneous environment. The platform independency feature of Web services technology is considered as a revolutionary value for enterprise collaboration.

Tightly Coupled Components. In existing collaborative environment, components are mostly dependent on each other. In other words, any kind of modification results in changing the other involved components. The tightly coupled behavior of components may cause some side effects such as: inconsistency, incompatibility, reliability, and failure. The involved enterprises in any coalition should be free to change their environment and legacy systems, regardless of concerning about raising the tightly coupled problems.

Enterprises, by using Web services, are able to create some independent components, which are free to be modified and replaced without affecting any other component in the collaboration. In fact, Web services technology provides a loosely coupled environment, at which enterprises freely take full control of their legacy systems regardless of the others' components.

Service Locating. Enterprise collaboration demands enterprises discover and locate the desired resources as services. Traditional methods in locating desirable services and their providers, such as advertising, cannot be considered as an efficient method. Apparently, in a competitive environment, efficient and easy locating of a suitable enterprise is one of the main issues in a successful collaboration.

Web services technology, by introducing the *Universal Description, Discovery and Integration* (UDDI) standard, provides some facilities for enterprises to locate their suitable and matched services. Enterprises by registering through the UDDI registry, advertise their offered services more effectively. The role of the UDDI registry can be considered as the role of yellow pages in any telephone directory. Consequently, a service requester, by searching the UDDI database, discovers the desired enterprises that are offering a desired service. The UDDI facility enhances both the creation and evolution levels of a virtual enterprise's life cycle.

Module Redundancy. Existing technologies of software industry for establishing a virtual enterprise suffers from redundancy in developing software objects as services. Software components are not reusable through a heterogeneous environment. Some technologies, such as *Common Object Request Broker Architecture* (CORBA), are addressing re-usability in a limited approach that makes them difficult to be used in heterogeneous platforms.

The designed methods, as offered services, in a Web service can be used in any application and programming languages regardless of their platforms. Thus, to develop an application there is no need for implementation from the beginning. Application developers that need to implement a method, or service, invoke the offered methods as available services and create their application products. The module reusability improves the product reliability as well. Interested readers are invited to refer to [2] for more discussions.

3. RELATD WORK

Agent technology has emerged as a promising technology for dealing with cooperation and decision-making in distributed applications. A number of researchers have attempted to apply agent technology to manufacturing enterprise integration, supply chain management, manufacturing planning, scheduling and control, materials handling, and holonic manufacturing systems [3][4]. This section briefly reviews some related projects.

Our approach is very similar to the MetaMorph [5] project, which uses agent-based mediator-centric federation architecture. In MetaMorph, each agent links with mediator agents to find out agents in the environment. This type of architecture is suited for complex distributed integrated manufacturing with a large number of agents. However, our approach is based on adopting Web services technology to facilitate and overcome some difficulties of using the agent technology. The AARIA [6] project has been designed to communicate with customers and suppliers and allocate resources to new jobs as they enter into the system, and to optimize schedules across resources. Some agents have been taken into account for Unit Processes. The AARIA incorporates new features for scheduling optimization and fault recovery.

There are some other agent-based projects related to virtual enterprises and supply chain management. The ISCM [7] addresses the coordination problems at both the tactical and operational levels. The supply chain is organized as a network of cooperation intelligent agents, each performing one or more supply chain functions. The CIIMPLEX [8] is composed of service agents such as Agent Name Server, Broker Agent, etc. and special agents each of which possesses specialized expertise for a particular application. The MetaMorph II [9] project uses a hybrid agent-based architecture, which is organized at the highest level through subsystem mediators. Each subsystem is integrated within the system through a mediator and each subsystem can be an agent-based system. The work in [10] has a very interesting discussion regarding agent-based manufacturing.

4. Web Services vs. Software Agents: A Comparison

In this section, we compare Web services and software agents to discover the similarities of these two technologies. In other words, we are concerned about the possibility of treating a Web service as an agent. In order to have a standard and reasonable comparison, we focus on the standard specification of "*Foundation for Intelligent Physical Agents*," FIPA [11]. Technically, any object with some properties can be viewed as a weak agent. A weak agent is an object with some agenda. Thus, *an independent functioning unit*, such as an object, can be viewed as a weak agent. On the other hand, a Web service, as an independent functioning unit (autonomous), with some agenda (offering services) can be treated as a weak agent.

FIPA defines an agent as [11]: *"An agent is a computational process that implements the autonomous, communicating functionality of an application."* or in another definition [11]: "*An agent is an encapsulated software entity with its own state, behavior, thread of controls and ability to interact and communicate with other entities, including people, other agents, and legacy systems."*

Now we try to describe a Web service in terms of the first definition of an agent: "*A Web service is a computational process (method and procedure) that implements (implementation part) the autonomous (not being controlled by another Web service), communicating (throughout Simple Object Access Protocol (SOAP) messages) functionalities of an application (service)."*

Based on the second definition of an agent, we describe a Web service as: *"A Web service is an encapsulated software entity (by interface) with its own state (some rules), behaviors (reaction to requests), thread of control (internal procedures) and an ability to interact and communicate (through SOAP messages) with other entities including people (client request), other agents (other Web services), and legacy systems (regardless of platforms)."*

Table 1 compares the basic requirements for interactions in both agents and Web services technologies.

Software Agent Technology	Web Services Technology
An agent needs to publish its services and resources to other agents as well as discovering services or resources of other agents [11]. This is achievable through registration in the Directory Facilitator (DF).	A Web service publishes its services or resources in order to be discovered and used by other Web services or clients. This is achievable through registration in the UDDI registry.
Agents need to communicate with each other. They should also be able to use existing infrastructure (TCP/IP networking, HTTP, etc) [11].	Web services communicate with each other through SOAP messages standard. The transportation protocol of Web services is HTTP (regardless of platforms).
Agents need to be able to access (existing) information sources [11].	Web services, as long as the XML based SOAP messages transportation is possible, can access to existing information sources.
The communication should be robust in the sense that recovery from exception conditions should be possible [11].	An asynchronous communication is robust. In the synchronous case, which a Web service needs some information from another one, it suspends the execution until receiving information. However, a Web service is multi-threats and offers services to a large number of users at the same time.

Table 1: Agents and Web services corresponding basic requirements for interaction

Each Web service has both interface and implementation. In terms of agent technology, the service implementation has the role of software system and the service interface plays the role of *wrapper* agent. The service interface or wrapper agent allows a client to connect to a software system. Furthermore, the UDDI registry plays the role of *Directory Facilitator (DF)* in agent paradigm. In Web services paradigm, services register through the UDDI, (*or DF*), and any client, (*or agent*), interested to find a software system queries the UDDI. Consequently, the UDDI returns the service, which offers the desired software system. Simply the client, (*or agent*), queries the interface of service, (*or wrapper*), to use the software system.

5. A Web Services Oriented Agent-Based Approach for Collaborative Manufacturing

Web services are able to provide an integrated solution toward creating an Internet based collaborative environment for resource sharing and interaction over the network. Internet based collaboration allows enterprises to share their resources, integrate their processes, and provide a new kind of service that is beyond the capability of any individual enterprise. The resources consist of a wide variety of categories such as physical machines, databases, applications, software procedures, and even human skills and experts. Therefore, the main objective of designing a collaborative environment is to share resources over the Internet. Moreover, privacy of each involved enterprise, which is concerned about securing its internal policies, capabilities, and resources, has a significant impact on designing a collaborative environment and achieving a successful result.

5.1. System Architecture

Internet based collaboration design must clarify some essential aspects such as: communication mechanism, resource locating, resource sharing, and inter-operational mechanism. We address these requirements by adopting Web services technology. The architecture of an agent-based Web services environment for enterprises collaboration is depicted in Figure 1.

Any individual enterprise, based on its requirements, deploys a number of Web services. Each Web service behaves as an *agent* to manage the enterprise physical resources and inter-operating with others. Enterprises by registering and advertising in UDDI offer their capabilities, resources, and generally services. A Web portal, by discovering related or similar services through the UDDI, achieves some meta-data as knowledge and provides a new kind of service namely as facility service, which is treated as a Web service as well. In fact the Web portal behaves as a *``Web portal agent''* to assist an individual customer, or *``user agent''*, which is interested in using some resources through establishing a collaboration. In other words, the Web portal agent assists the user agent to find its business partners and set up a collaborative environment. The Web portal agent, which is considered as a Web service, can registers through UDDI as well.

In following we discuss the internal architecture of components involved in a simple collaborative environment.

Customer (User Agent). A customer behaves as an agent that needs some specific services, locates suitable service providers, and involved in collaborative environment. A user agent is capable of using the Web service communication language, known as SOAP messages over the Internet communication protocol, or HTTP. A user agent, based on its *knowledge* and *preferences*, defines ultimate goals. The designing goal is through *reasoning capability.* In fact, a user agent simply can be a personal assistant application, which by providing some information, assists the end user in finding a suitable business partner.

Enterprise (Service Agent). An enterprise is a knowledge base business entity that provides some resources as services. At the communication level, any enterprise has knowledge about Web services communication standard, or SOAP messages, over the Internet. The main software component of any enterprise is its Web service, which acts as an agent and is responsible for analyzing incoming SOAP messages, or user requests. Thus an enterprise, by analyzing the content of messages and with respect to resources availability, provides resources as offered services. The analyzing mechanism is through the *reasoning* capability of an agent based Web service. A simple Web service consists of following components:

- **Interface**. All communication between the Web service and outside world is handled through an interface, which behaves as a wrapper agent capable of encapsulating the implementation part and providing a privacy mechanism to protect the internal structures and policies of the enterprise from the outside view.
- **Implementation.** The implementation component acts as a reasoning part for the Web service. The implementation part by achieving incoming requests and based on its knowledge and resource availability, reasons about offering a service. Service offering is handled by the scheduler component and based on

resource availability. The scheduler can be considered as a service, which is responsible for assigning jobs to existing resources and keeping track of tasks. The implementation component consists of some methods and rules, which are protected by interface component.

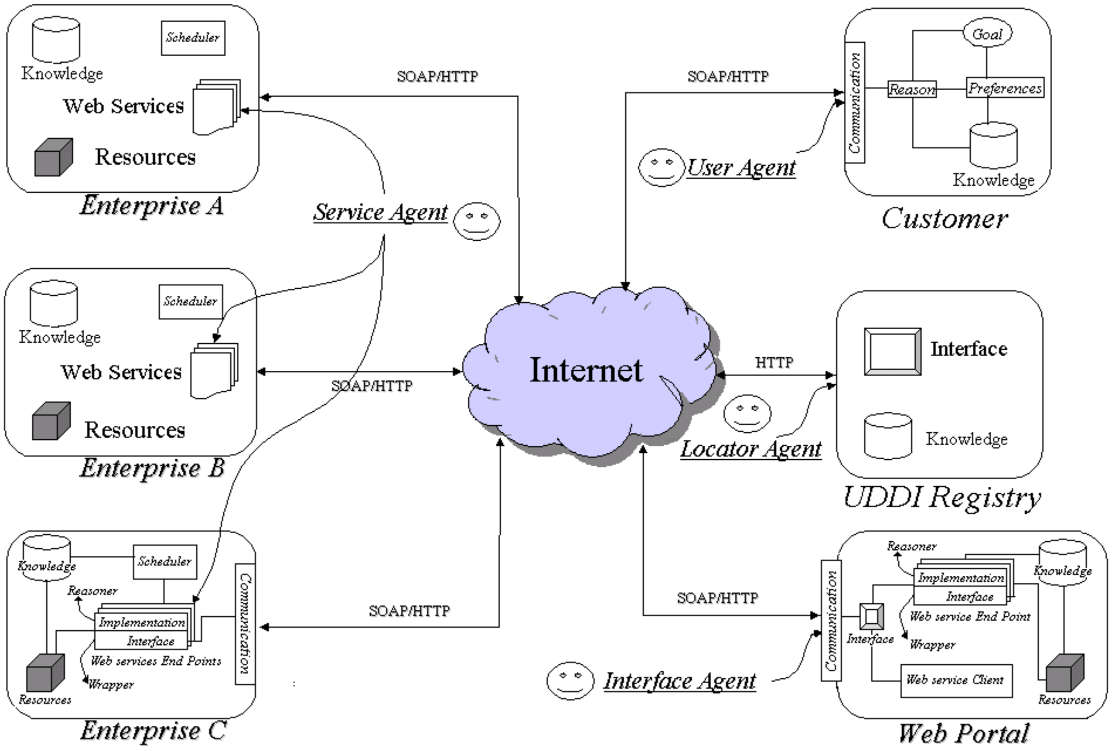

Figure 1: A Web services / agent-based architecture for manufacturing

Web Portal (Agent). A Web portal can be designed either as a Web *server* or a Web *service*. In any cases, it behaves as an agent, which is responsible for locating resources, or services. Moreover, in case of Web service, may be responsible for offering new kind of services, namely as facility service.

The interface component allows customers to interact with the Web portal. In other words, the interface is an application, which receives the incoming requests, and delivers them to other internal components. The Web portal has its own knowledge and resources as offered services.

A Web portal should have knowledge about SOAP messages in order to communicate with other Web services. The Web portal may act as a *coordinator agent* as well. More precisely, in case of needs to a new service, the Web portal, by analyzing the goal and looking at UDDI, locates some desired services and delegates the sub goals to each service provider and coordinates their activities.

5.2. SYSTEM INTERACTIONS

The interaction among different involved entities describes the behavioral properties of the system. Clearly, messages are based on the structure of SOAP. In following we describe the interactions among entities.

Registering with the UDDI server. Any service provider, in order to utilize the discovery of service, may register with the UDDI server. The registration facility may cover any kind of services, either elementary or composite. The Web portal may also register through the UDDI server as well.

End User - User Agent. The ultimate goals and their preferences are defined by end user, or service requester. The user agent assists the end user for choosing the most suitable service. In fact, the user agent provides the ability for end users to submit their requests to the Web portal agent or to check for any service that would be of an interest. The user agent, by accepting the goals and their performance, contacts either with the Web portal agent or, by looking at UDDI, contacts with service providers directly. Eventually, by receiving the proposals from service providers, the user agent by its knowledge, reasons about the proposals and assists the end user to choose the most appropriate service provider.

User Agent - Web portal Agent. A user agent contacts with Web portal agent and submits the desired services, or goals including the preferences. The communication takes place through sending some SOAP messages. The Web portal, which provides may act as a service provider, receives the SOAP messages and analyzes the request. In case of being able to provide the requested service by the existing one, the Web portal sends back a SOAP message to the user agent including the proposal. The proposal is based on the reasoning and resource availability in the Web portal. Consequently, the user agent reasons about the proposal and makes a decision. In case of needs to another service provider, the Web portal agent communicates with the UDDI server.

Web portal Agent - UDDI. The Web portal sends some SOAP messages to UDDI in order to search and find suitable services. The UDDI, based on its knowledge, sends back a SOAP message to the Web portal including the service providers and the URL's link to their *Web Service Description Languages (WSDL)* files. The WSDL file describes some meta-data in order to present how to communicate with service.

Web portal Agent - Service Agent. The Web portal by achieving the list of potential services and their providers, sends some SOAP messages asking for their proposals. By receiving the SOAP messages, the enterprises with respect to resource availability reason about offering proposals to the Web portal and consequently the user agent. The intelligent Web portal, by receiving the proposals and using some evaluation parameters, such as user preferences, may select the most appropriate service provider. The Web portal agent may:

- Remove all proposals with services that do not match the requested service;
- Remove all proposals with services that do not satisfy the user preferences;
- Negotiate with the candidate service providers on behalf of the user agent.

Service Agent - Web Services. The service agent assists an individual enterprise to make the most suitable decision for offering a reasonable proposal. The service agent is responsible for reasoning about resource availability, negotiating with service requesters, either user agent or Web portal agent, and placing an order into the Web service .

6. Prototype Implementation

In order to prove the feasibility of the proposed model, a simple prototype is developed. The prototype takes advantages of Java language and the Java based toolkit for Web services, known as JAX-RPC. The prototype is a simplified distributed system, which represents integration and resource sharing through a cooperative distributed system. It consists of following entities:

- A number of 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 on some popular Web programming tools and languages under the Windows NT/2000 environment. Java API for XML-based RPC known as JAX-RPC has been used to create and deploy all Web services.

As a simple scenario, there are number of companies that possess some expensive machines offered as services to customers. A service requester, interested in using these machines through communicating with Web portal, asks for the bids. Consequently, the Web portal, which has knowledge about the services and their providers, sends a SOAP message to enterprises namely as *"Call for Bids"*. Thus, any service provider based on their facilities and availabilities proposes its bid to Web portal. The Web portal, by analyzing the received bids, suggests the most suitable service to the customer. The configuration of the prototype is depicted in Figure 2.

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 database for both the Web services and the Web portal application is created by Cloudscape database.

The Web portal is created by Java Servlets, deployed on Apache Tomcat Web container as well. The server side programs or Servlets are invoked by Java Server Page (JSP) and HTML programs. The Web portal has its own database, which is created by Cloudscape database.

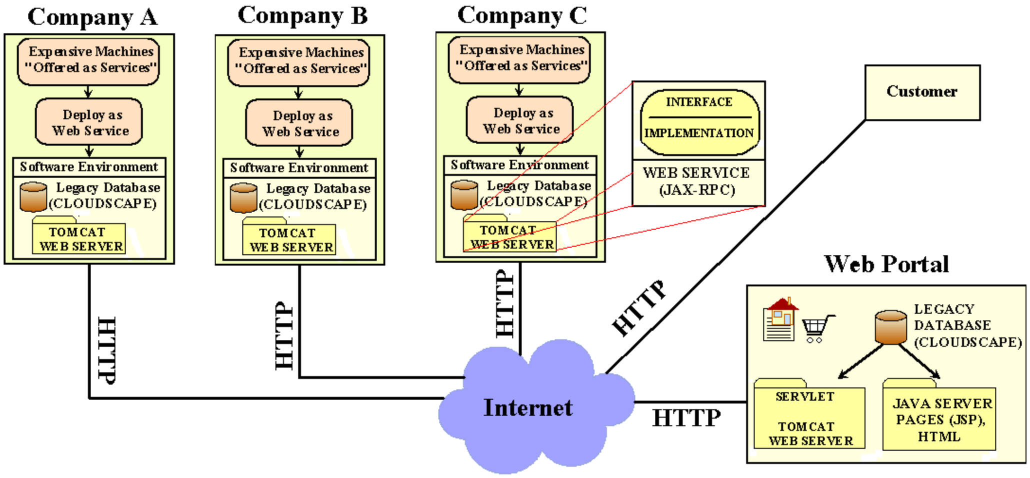

Figure 2: The machine sharing scenario

7. A Scenario For Machines Sharing

In developed prototype, there are some companies that own some expensive machines and like offering as available services or resources through the Internet. Each machine has its own *“speed”* and *“price*.” The speed represents the volume of a machining task, which can be accomplished by the machine in every minute and the price is the cost of work for each machine per minute.

One of the main benefits of implementing a Web service is encapsulating the complexity of services as well as keeping privacy of internal methods, data, and resources. In our prototype, customers have no idea about the internal structures, available machines, or internal policies of a specific enterprise. A customer, via the Web portal, sends its requests. The Web portal, which has knowledge about the *Web Service Definition Language (*WSDL) file for each Web service, delivers the requests to the enterprises' Web services.

Each company has its own legacy database system, which is updated locally in real time. The Web portal consists of some methods to define, submit, and achieve the results back from each Web service. Any customer, by contacting with the Web portal through the Internet, defines any number of jobs, which are packed under one bundle, and then submits the bundle to the Web portal. Consequently, the Web portal sends the bundle to each of the enterprises, via SOAP messages and their Web services and asks them to offer their proposals. Each bundle consists of some tasks, which are defined in terms of machine's type, volumes, and due dates, by which all jobs in the bundle should be completed.

Moreover, we suppose each machine in a company has a special rank that ranks machines with respect to their speed and cost. Assigning jobs to each machine is based on the rank that results in offering bids to customers with the lowest cost as much as possible.

Each enterprise based on the availability of machines and with respect to the due date of the requested bundle, decides to offer any bid. If the requested bundle can be carried out by the company before the due date, then the enterprise offers its bid.

By receiving proposals, the customer chooses the best and most suitable offer and places its order through the Web portal to make contract and service agreement. Service agreement is just registering customer's information and saving into the company's database and issuing the job reference numbers to show the order and service agreement.

Each customer, by browsing the entry page of Web portal, requests bids and places an order. Communication between entities is carried out by HTTP, the Internet communication protocol.

8. Conclusion and Future Contribution

In order to carry out customers' requests, manufacturing enterprises need to collaborate and share their resources. Enterprise collaboration, permanent or temporary, requests a higher level of technology, which allows enterprises to integrate their applications regardless of platforms, data structures, or data models. Collaboration for small and medium size enterprises is a mandatory issue towards sharing their skills and resources in order to be able to compete. In this paper we proposed a Web services oriented agent-based approach for inter-enterprise collaboration. The main contribution of our approach is on adopting Web services as agent-based units capable of providing application integration over the Internet. The compatibility of Web services technology with the standard specification of FIPA for software agents presents the ongoing approaches toward defining service-oriented paradigm in which each resource is considered as a service capable of interaction and exchanging information through the network and participating in a collaborative environment regardless of platforms and locations.

However, Web services technology is not capable of handling all required capabilities of collaboration. Software agent can be considered as a complementary paradigm besides Web services technology to provide more collaborative environment. Among unique characteristics of agents is pro-activeness. A Web service is not a pro-active entity, which could be able to offer intelligent services. Moreover, a Web service is not an active component and service requesters must have knowledge about them in order to locate and use offered services. As future work, we are concerned about the feasibility of providing integration between software agents and Web services in order to take advantages of both technologies and their properties toward preparing a suitable environment for collaboration. In our view, the unique characteristics of intelligent software agents and revolutionized properties of Web services technology as emerging paradigm, together, can provide a suitable environment for an intelligent collaboration over the Internet.

References

[1] Camarinha-Matos, H. Afsarmanesh: "Infrastructure for Virtual Enterprise", Kluwer Academic Publisher, Norwell, Massachusetts, 1999.

[2] A. Siami Namin, W. Shen: "Application of Web Services and Software Agents to Inter-Enterprise Collaboration", Controlled Technical Report, NRC-IMTI, 2003.

[3] W. Shen, D. H. Norrie, J. P. Barthes: "Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing", Taylor & Francis Inc., 2001.

[4] W. Shen and D. H. Norrie: "Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey", Knowledge and Information Systems, vol. 1, no. 2, pp. 129-156, 1999.

[5] F. Maturana, W. Shen, D.H. Norrie: "MetaMorph: An Adaptive Agent-Based Architecture for Intelligent Manufacturing", International Journal of Production Research, 37(10): 2159-2174,1999.

[6] H. V. D. Parunak, A. D. Baker, S. J. Clark: "The AARIA Agent Architecture: From Manufacturing Requirements to Agent-Based System Design", In Working Notes of the Agent-Based Manufacturing Workshop, Minneapolis, MN, pp. 136-145,1998.

[7] M.S. Fox, J.F. Chionglo, M. Barbuceanu: "The Integrated Supply Chain Management System", Internal Report, Dept of Industrial Engineering, Univ. of Toronto,1993.

[8] Y. Peng, T. Finin, Y. Labrou, B. Chu, J. Long, W. J. Tolone, A. Boughannam: "A Multi-Agent System for Enterprise Integration", In proceeding of PAAM'98, London, UK, pp. 533-548,1998.

[9] W. Shen, F. Maturana and D. H. Norrie: MetaMorph II: An Agent-Based Architecture for Distributed Intelligent Design and Manufacturing, Journal of Intelligent Manufacturing ,11(3), 237-251, 2000

[10] W. Shen and D. H. Norrie: "Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey", Knowledge and Information Systems, vol. 1, no. 2, pp. 129-156, 1999.

[11] http://wiki.cs.uiuc.edu/cs427/Fipa OS.