
Implementing enterprise collaboration using web services and software agents

Akbar Siami Namin*

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

Department of Electrical and Computer Engineering,
University of Western Ontario,
London, Ontario, Canada

*Corresponding author

Weiming Shen

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

Hamada Ghenniwa

Department of Electrical and Computer Engineering,
University of Western Ontario,
London, Ontario, Canada

Abstract: Global competition has forced manufacturing enterprises to collaborate together towards fulfilling market demands and satisfying customers. Through collaboration, enterprises will be able to share and effectively utilise their resources. A major challenge in implementing enterprise collaboration is based on the integration of heterogeneous hardware and software platforms. Service-oriented computing, an emerging paradigm in distributed computing, empowers applications integration over the network regardless of platforms and environments. Software agents have been recognised as a promising technology for dealing with cooperation, coordination and decision-making in distributed applications. This paper proposes to implement enterprise collaborations by employing web services and software agents. It presents a multi-layer architecture for enterprises' software infrastructure capable of accomplishing the essential functionalities in typical collaborations such as resource discovery, workflow designing, requests reasoning, tasks assigning and monitoring. In addition, this paper proposes an agent-based multi-layer structure for service discovery to improve the expected functionalities of resource discoveries and their operation more effectively.

Keywords: service-oriented computing; web services; software agents; enterprise collaboration; virtual enterprise.

Reference to this paper should be made as follows: Namin, A.S., Shen, W. and Ghenniwa, H. (XXXX) 'Implementing enterprise collaboration using web services and software agents', *Int. J. Networking and Virtual Organisations*, Vol. X, No. Y, pp.XXX-XXX.

Biographical notes: Please provide biographical notes for authors A.S. Namin, W. Shen and H. Ghenniwa not more than 100 words.

1 Introduction

Global competition has forced manufacturing enterprises, particularly Small- and Medium-sized Enterprises (SMEs), to increase their productivity and profitability through optimal resource utilisation to fulfill market demands and achieve customers' satisfactions. On the other hand, a rapid alteration of business demands and especially manufacturing environments make resource utilisation more and more erratic and unstable. Conventional congregations of enterprises operating together try to solve the given problem by production outsourcing to achieve maximal group benefits. We consider virtual enterprise as one form of enterprise collaboration, which focuses on horizontal collaboration among members of collaborative enterprises (Shen et al., to appear). The advantages of forming virtual enterprises enabled by information and communication technologies supply new ways in facilitating inter-enterprise manufacturing resource sharing and therefore enhancing the profitability of SMEs (Camarinha-Matos and Afsarmanesh, 1999).

However, implementing a virtual enterprise or an enterprise collaboration environment is not a straightforward process and requires some considerations to be taken into account. In terms of developing a virtual enterprise, some of the existing challenges are: locating suitable business partners, constituting common communication protocols, designing coordination mechanisms to explore any interdependency among resources and to resolve any possible conflicts as well as checking and monitoring the status of all committed tasks for re-evaluation purposes. A major challenge in implementing such an environment is based on the integration of heterogeneous hardware and software platforms.

Service-oriented computing is a revolutionary paradigm for modelling and integrating distributed application systems. A service is a contractually defined behaviour that can be implemented as a component for use by another component (The Openwings). Owing to its sophisticated capabilities, service-oriented technology has been supported by industry extensively. IBM, Microsoft and BEA are IT companies leading the standardisation of service-oriented specifications and protocols. Service-oriented technology embodies some outstanding properties such that no other former software technology was able to do so. Interoperability, loose coupling, service discovery and system modularity are among some benefits that might be achievable through using service-oriented technology. In addition, platform independency and interoperability over heterogeneous systems make service-oriented technology as an appropriate solution in implementing a virtual enterprise or an enterprise collaboration environment in general.

Although some sophisticated service-oriented platforms such as openwings (<http://www.openwings.org/>) and Jini technology (<http://www.jini.org/>) have been developed to construct services in any networks, the concept of service-oriented computing has emerged the idea of feasibility of designing as well as figuring the internet as a service-oriented environment. Web services are known as code level or web-based service-oriented computing. They are self-descriptive software components, equipped with some globally established standards, which seems to be suitable for implementing virtual enterprises and in a wider domain designing enterprise collaborative environments over the internet.

Applying service-oriented paradigm in general and web services in particular, the following benefits may be achievable:

- platform independency by bringing up communication to a higher level such as application rather than operating system or machine level
- loose coupling among components so that they can be modified, replaced or removed from the network without affecting the collaboration
- service discovery via registering, advertising and searching through '*Universal, Description, Discovery and Integration (UDDI)*' (<http://www.uddi.org/>) standard
- system modularity and as a result the reusability of reliable services
- encapsulating the internal complexity of applications by allowing service consumers to communicate throughout an interface
- XML-based messaging standard protocol known as '*Simple Object Access Protocol (SOAP)*' (<http://www.w3.org/TR/soap/>) allowing messages passing the fired-wall protected web servers
- service descriptiveness by reasoning about the content of '*Web Service Description Language (WSDL)*' (<http://www.w3.org/TR/wsdl/>) provided for any deployed services to determine how to interact with services.

Despite some revolutionary mechanisms offered by service-oriented computing and specifically web services including their standards, the technology itself cannot exhibit a complete solution for enterprise collaboration.

Software agents have emerged as a promising technology for dealing with cooperation, coordination and decision-making in distributed systems. As an application domain, agent-based manufacturing has become a new paradigm for the next generation of manufacturing systems, together with other manufacturing paradigms such as Holonic Manufacturing, Agile and Reconfigurable Manufacturing (Shen et al., 2001). Software agents have been developed with some sophisticated interaction patterns making them proficient in enforcing automatic and dynamic collaborations. Agent-orientation is an appropriate design paradigm for e-business systems with complex and distributed transactions, particularly web services. In services realisation, 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 a complementary technology to overcome some insufficiency of service-oriented technology, particularly for utilising them in collaborations:

- Services are known as passive objects; whereas, software agents behave as active ones. By passive, we mean lacking the capability such as pro-activeness in approaching other entities. Hence, services are not capable of joining collaborations autonomously, which has been considered as a key factor in the success of any collaboration. Service locators must discover services by themselves. On the contrary, software agents are pro-active in presenting their services and capabilities.
- As stated above, an effective collaboration demands a component acting as a coordinator to design, manage and assign sub-tasks to the qualified parties. In addition, the coordinator has to resolve conflicts, monitor activities and detect any failures or interdependency causing a deadlock throughout the life cycle of collaboration. Clearly, services are unable to handle such a coordination. In contrast, software agents can play this role very well.
- Service discovery offered by service-oriented paradigm and particularly by the web services standard in discovery, known as UDDI, is syntax-based. In other words, a service requester locates services based on their syntax and contextually matching rather than any semantic efforts. As a result, services may not be fully discovered and therefore the effectiveness of using service discovery will not be completely achieved. A possible solution is to define domain ontologies for various application domains, which creates some other challenging issues. Therefore, a better solution is to equip ontologies with software agents and employ their power in reasoning about the semantic of requests.

Based on our theoretical analysis and previous experience, we envision the integration of service orientation and agent orientation as an appropriate computational paradigm for open and dynamic collaborative environments. It is a natural way to integrate these two technologies into a cohesive body so that we attempt to avoid the weaknesses of each individual technology, while capitalising on their individual strengths.

This paper presents our recent work on the implementation of enterprise collaboration using web services and software agents. The rest of this paper is organised as follows. Section 2 provides a brief literature review; Section 3 presents an overall view of proposed integration model; Section 4 describes a collaborative enterprise architecture; Section 5 depicts the service discovery model; Section 6 discusses some implementation issues related to the proposed model and Section 7 concludes the paper with discussion on the future work.

2 Related work

Virtual enterprises have been studied extensively in the past few decades. Agent-based virtual enterprises have been an active theme of researches until the service-oriented computing paradigm emerges. Adopting software agents in implementing virtual enterprises are discussed remarkably (Camarinha-Matos and Afsarmanesh, 1999; Marik and Pechoucek, 2003; Rabelo et al., 2001). Some research projects have been proposed

on supply chain management and enterprise integration in recent years. Particularly, web services seems to be the main ingredient of the next generation of computerised virtual enterprises. Moreover, the integration of software agents and web services seems to be promising.

Integrated Supply Chain Management (ISCM) is a research project focussing on decentralising the control in manufacturing system design using software agents (Barbuceanu and Fox, 1995; Fox et al., 1993). ISCM consists of agents working cooperatively and each agent coordinates its own activities with respect to other agents. In fact, a distributed and decentralised coordination is the main theme of the ISCM project.

The Consortium for Intelligent Integrated Manufacturing Planning-Execution (CIIMPLEX) has been developed for supporting intelligent manufacturing and enterprise supply chain integration (Peng et al., 1998, 1999). The system consists of two main categories of agents named as service and special agents. The service agents are responsible for fulfilling some well-known activities and for facilitating collaboration among agents; whereas, the special agents are expert agents for special applications. There are three types of service agents: server agent, broker agent and gateway agent.

The Agile Infrastructure for Manufacturing Systems (AIMS) was developed to facilitate agile manufacturing through negotiation over the internet (Park et al., 1993). AIMS consists of certified manufacturing services linked and formed a network via the internet. A certified supplier employs standard business processes. Customers interact with certified services through exchanging structured messages.

The MetaMorph project was developed to integrate the manufacturing enterprise's activities such as design, planning and scheduling with those of its suppliers, customers and partners in a distributed intelligent open environment (Shen et al., 1998, to appear). In MetaMorph's architecture, the manufacturing system is organised at the highest level through subsystem mediators and each subsystem is integrated in the system through a mediator. As a matter of fact, each subsystem is an autonomous agent-based system. Agents in one subsystem may also interact directly with agents in other subsystems.

The projects given above are dealing with enterprise collaboration including virtual enterprises and supply chain management. The rest part of this section reviews some related projects on integrating web services, software agents as well as workflow management technology.

The use of agents in Workflow Management Systems (WfMS) has been of some interest for researchers (Yan et al., 2001). In an approach presented by Chang and Scott (1996) a particular workflow has been managed by some agents named as personal, actor and authorisation. These agents function on behalf of the involved parties in a workflow and facilitate interaction with others either inside or outside the collaboration.

A multi-agent architecture named as ADEPT consists of a number of autonomous agencies (Jennings et al., 1996). Every single agency consists of a set of subsiding agencies, which are controlled by another agent and each agent is able to perform one or more activities as offered services. A usage of agents to integrate '*cross-enterprise*' dynamic workflow has been presented by Zeng et al. (2001). Three components have been defined as Workflow Definition Tool, Agent Society and Actual Services.

Shadow board agent architecture was proposed by Jin and Goschnick (2003). It proposed an orchestration of multiple web services in an agent-based transaction. Each party is wrapped as a web service. The architecture uses an agent-oriented approach

to engineer each web service and treats each of them as a software agent. The model does not address some issues such as initialisation of workflows, monitoring the status of subtasks and a flexible rating mechanism.

An interesting conceptual and generic approach was proposed by Singh (2003). The approach is based on '*temporal logic*', which has rigorous semantics, and yields a naturally distributed execution. A multi-agent system has been taken on the problem of service compositions and emphasising the agent-based distributed computing.

Interleaving web services composition and execution, using software agents and delegation have been discussed by Maamar et al. (2003). A representation composition constraint for semantic web services has been proposed by Cheng et al. (2003). The work proposed the Agent Service Description Language (ASDL), to describe the external behaviour of the agent services. By using ASDL it is possible to define the behavioural characteristics of a web service.

Maximilein et al. discussed a conceptual model for web service reputation (Maximilein and Singh, 2001, 2002). In the model, a '*WSAP*' is defined to access each service. A WSAP is an agent that acts as a proxy for clients of web services.

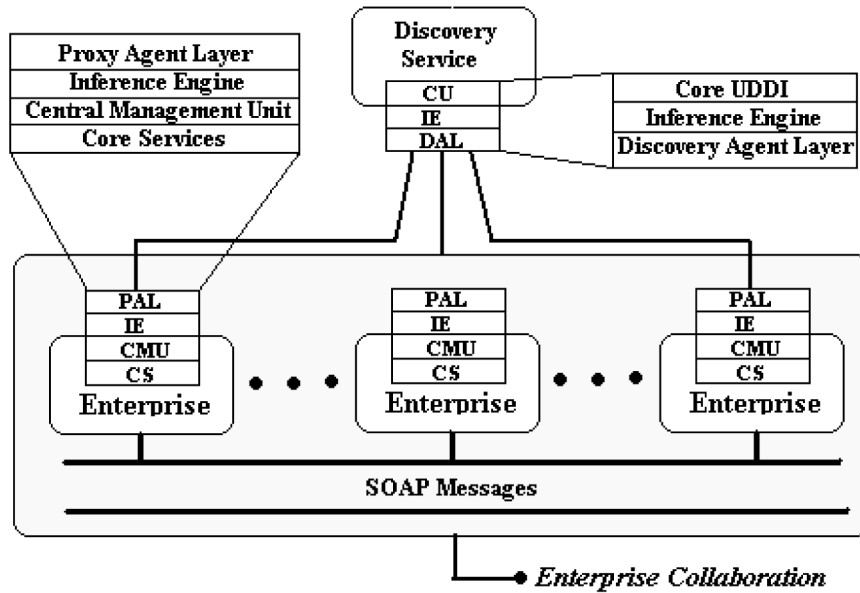
This paper discusses the integration of service-oriented computing and software agent paradigm to take advantages of both technologies and to address the existing challenges regarding the implementation of enterprises collaboration. This paper proposes an integrated model of service-oriented components as well as agent-based entities working collaboratively towards handling the required functionalities in developing a virtual enterprise. We try to address the main challenges of setting up a typical virtual enterprise by defining corresponding entities. In addition, we model an agent-based system for service discovery to assist service locators more effectively.

3 Integrating web-services and software agents for enterprise collaboration

This section presents a novel architecture for enterprises collaboration employing web services and software agent technologies. We model the architecture for application domain of enterprises as well as the internal structure of a service discovery. For the former, we define service-oriented agent-based components with respect to the required roles and functionalities expected in a typical virtual enterprise's life cycle. For the latter, we use software agents to enhance the functionalities of a service discovery. Figure 1 depicts the proposed architecture at a high level.

The proposed architecture addresses the essential functionalities in any collaboration such as enterprise locating, coordination, delegation and monitoring the status of subtasks. By modelling service discoveries as depicted in Figure 1, the expected functionalities of designing service discoveries will be acquired and as a result, the overall effectiveness of service discoveries will be enhanced. Among some of the achievable features are locating '*suitable*' services with respect to their performances as well as the feasibility of searching services in terms of semantics rather than syntax. These features are achievable through designing a multi-layer service discovery and cooperation among designed components sited on the service discovery.

Figure 1 A service-oriented agent-based integration architecture for enterprises collaboration



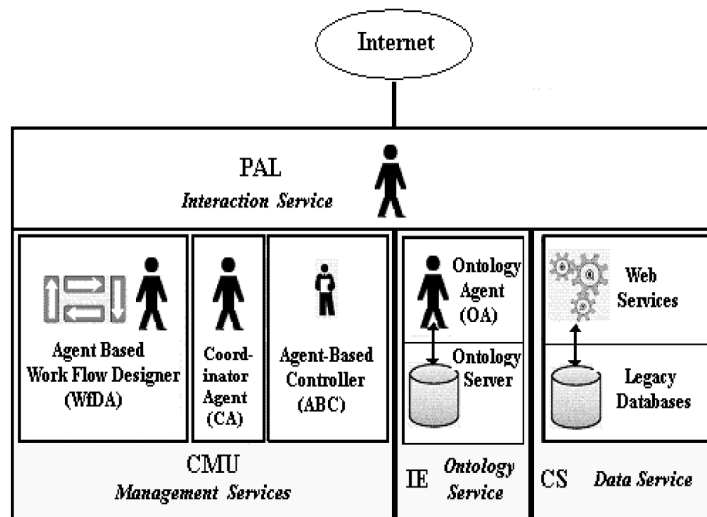
Let us review a simple scenario. A manufacturing enterprise, as a service provider, deploys its resources as services and registers them through a service discovery. A service discovery is a data source of collecting some meta-data, associated with registered services in terms of services, providers, capabilities, categories and their interaction methods. To simplify, we consider the service discovery standard in web services as UDDI. The interaction method of a web service is achievable by reasoning and analysing the content of 'WSDL'. The XML-based WSDL is a descriptive language elaborating the interaction methods for using the service. Any enterprise, as a service requester, searches and consequently browses the matched services that have satisfied the desired criteria and the requester's preferences. More precisely, the result of search and the content of the browse will be the matched services, their providers, their categories and references to their WSDL files as descriptions for interaction purposes. Enterprises, by reasoning on the content of a WSDL file, learn how to interact with the service. A message-oriented exchange communication mechanism using 'SOAP' messages will be employed for interaction purposes throughout services and their providers.

Let us take a quick overview of the defined components in the architecture. An agent-based unit named as '*Proxy Agents Layer (PAL)*' is defined on top of the enterprise software infrastructure. PAL is an interface empowered by reasoning about incoming and outgoing messages. A corresponding agent-based layer named as '*Discovery Agent Layer (DAL)*' with similar functionalities is defined on top of the service discovery. The commonly defined component, both on the enterprise and the service discovery, called as '*Inference Engine (IE)*' is the knowledge-based ontology unit enabling both enterprises and service discoveries reasoning about the domain ontology of the exchanged messages. In the following sections, we elaborate each unit and its functionalities in detail.

4 Collaborative enterprise architecture

Figure 2 shows a service-oriented agent-based architecture for an individual collaborative enterprise. The agent-based nature of the defined interface named as PAL turns a passive enterprise to an active entity equipped with cooperation and integration capabilities. PAL conducts all interactions between enterprises. From the service-oriented perspective, the interface is a service enabling communication and interaction between enterprises at an application level. The 'IE' is in charge of reasoning about the semantic and ontology of incoming messages such as requests, which PAL is not able to parse them. In fact, IE is an ontological service supporting enterprises in reasoning activities. The '*Central Management Unit (CMU)*' is the heart of the enterprise infrastructure that administers the critical activities in collaboration such as coordination, delegation as well as monitoring and re-evaluation. The '*Core Services (CS)*' unit is the same as existing web services technology. The following subsections introduce each unit in detail.

Figure 2 Service-oriented/agent-based architecture for an individual enterprise



4.1 PAL, interaction services

The most significant intuition in the failure of any collaboration might be due to lacking of an intelligent interface for supporting enterprises in communication and interaction effectively. PAL not only enables enterprises to interoperate and interact proactively, but also empowers them to function actively rather than passively. As a result, the model suggests considering PAL as a complementary unit for services offered by enterprises. We may look at PAL from different perspectives:

- As an object, PAL encapsulates the complexity of the internal structure of the application domain from customers' points of views. Therefore, clients deal with an intelligent unit rather than challenge with some complex packages of applications and methods.

- As an agent, it behaves as a wrapper agent packaging all functionalities and features of the internal structure under one umbrella and offering them to customers.
- As a service, it renders interaction services to other enterprises. Accordingly, interaction services may consist of some other services such as ‘*Communication service*’ to exchange messages, ‘*Locator service*’ to search the service discovery as well as ‘*Contract service*’ to provide some collaboration functionalities such as negotiation, service level agreement and contract management.

Furthermore, PAL behaves as a gateway for exchanging data among internal components such as CMU, IE and CS of the enterprise application domain. After all, regarding messaging framework standards, PAL must have enough knowledge to realise the exchanged messages. For instance, in our web services case, PAL must be able to reason about SOAP messages, WSDL descriptions and UDDI standard.

Additionally, PAL also has some other operational activities such as:

- Routing all messages to the related receivers either internally in the enterprise or externally to other enterprises. The routing activity consists of reasoning about the type of messages. In case of ambiguity in meaning of the messages, PAL uses the ontology unit IE and eventually in case of uncertainty about requested services, communicates with the CMU.
- Contacting the service discovery for either searching purposes or reporting the performances of consumed registered services. The service discovery uses the performance report as feedback information usable in reputation of the used services. As a result, any service requester by referring to reputation information will choose its business partner with a lower risk.

4.2 CMU, management services

As stated above, CMU is the heart of the enterprise infrastructure. CMU is an agent-based designer for defining and organising distributed processes in any collaborative system. In other words, CMU is authorised for the activities defined in creation and evaluation stages of the life cycle of a virtual enterprise. These activities may vary from designing workflows and delegation as well as coordination and monitoring the status of the assigned tasks. CMU reasons about the services requested by a customer and decides either to offer services from the existing ones or to embody a new workflow, and consequently forming a new virtual enterprise. We may look at CMU and its roles from different perspectives:

- As an agent, CMU is a set of collaborative agents incharge of creating and administrating further new virtual enterprises. More precisely, CMU is a multi-agent system inheriting all characteristics of a general multi-agent system such as coordination and cooperation. The model adopts these properties to address the required functionalities and roles needed in designing a collaborative system.
- As a service, CMU offers ‘*Collaborative Process Management Services*’ to the enterprise in particular and to the virtual enterprise in general.

In general, CMU, as a uniform unit, receives requests from PAL informing the needs for designing new workflows. Consequently, CMU reasons about the customer's request and envisions it as a goal, designs a workflow to achieve it, identifies subtasks by decomposition techniques, informs PAL to locate suitable services corresponding to each subtask through service discovery, delegates the subtasks to qualified service providers, coordinates them and eventually monitors their activities and performances. Indeed, the above processes represent the life cycle of creating a new virtual enterprise. CMU is structured of some subunits as follows.

4.2.1 *Workflow Designer Agents (WfDA), composer service*

Any particular workflow may be envisioned as a common goal to be achieved by entities. In fact, any workflow consists of some subtasks (subgoals) that are identifiable by applying any goal decomposition algorithm. Accomplishing each particular subtask, respectively, results in achieving the overall common goal. Hence, agents have such an outstanding capability on developing these kinds of algorithms. Moreover, the agent-based WfDA is designed to explore a cross-enterprise specification. This design can be handled not only statically but also dynamically.

- As a multi-agent system, the agents sited on the WfDA cooperate together to achieve a common goal. Each agent has its own subgoal and knowledge. However, as the cooperation expectation, they work together supportively to explore a solution for achieving the common goal.
- As a service, WfDA is responsible for composing new services. Service composition is known as one of the main sophisticated result and benefit of service-oriented paradigm and many researchers believe that service-oriented computing without composition is not a revolutionary paradigm. WfDA as a '*composer service*' is authorised to compose new services with respect to customers' criteria and their preferences. For composition purposes, WfDA may take either the existing services deployed by enterprise itself or those offered by other enterprises registered in the service discovery. The composed service may be registered in service discovery as a new service.

As the model heavily relies on the nature of a goal, we give a definition for a goal based on our expectation of this goal in the model.

Definition (Goal (G, Q_G)): A goal is a tuple (G, Q_G) such that:

- 1 G is the desired goal, which might be a set of sub-goals as: $G = \{g_i \mid 1 \leq i \leq k\}$.
 k is the number of sub-goals
- 2 Q_G is the desired preferences in achieving the goal G , which is a set of acceptable qualification parameters as $Q_G = \{q_n^G \mid 1 \leq n \leq m\}$. m is the number of preferences in accepting the goal G .

Now in terms of the above definition for a goal, we define an appropriate workflow process for our goal-based model.

Definition (Goal-Based Workflow Process (GBWf)): A GBWf is defined as a tuple $W \triangleq (\{G_i\}_{i=1}^p, \{R_i\}_{i=1}^p)$ such that:

- 1 p is the number of (sub) goals involved in the workflow
- 2 $\{G_i\}_{i=1}^p$, the set of (sub) goals to define tasks in a workflow
- 3 $\{R_i\}_{i=1}^p \in \{\text{Success, Failure}\}$, is the set of results of fulfilling of (sub) goals
- 4 $o(G_i) < o(G_{i+1})$, is the order of achieving goals, which means the i th goal G_i will happen before $(i+1)$ th goal G_{i+1} .

The above definition for a workflow fits more appropriately for the model than any other traditional definition of workflow. The definition takes into account: workflows as goals, the users' criteria in achieving the goals and also the order of their occurrence.

4.2.2 Coordination Agent (CA), coordination service

As we mentioned earlier, coordination is an essential component in any collaborative environment. In fact, coordination is needed for some critical purposes such as prevailing interdependency problems and conflict resolutions. CA coordinates the enterprises' activities involved in the sequence of processes defined in a workflow while operating their assigned tasks. CA receives a designed and planned workflow from WfDA and locates service providers through PAL that are talented in fulfilling the subgoals. Then, it assigns each task to a qualified enterprise. Eventually, CA informs the 'Agent-Based Controller (ABC)' to monitor the status of each task.

4.2.3 ABC, controller service

Monitoring and evaluating the performances and activities of involved enterprises are necessary to reform and re-evaluate a virtual enterprise. Identifying weak business partners and replacing them with strong ones have a significant impact on the quality of the ultimate products. The ABC creates a corresponding 'controller agent' for each subtask. The controller agent communicates with the target enterprise in charge of fulfilling task and collects some statistical data regarding progress of the task. In fact, the controller agent keeps track of each task. In other words, the controller agent monitors the status of the task and compares the result with the expected and scheduled plan. Eventually, ABC reports the progress of the monitored enterprise to the coordinator agent to make a decision.

The other defined units are 'CS' and 'IE'. CS is the enterprise's web-services technology. IE provides some ontological services for enterprises to reason about the semantics of the exchanged messages. As the designed IE sited on the enterprise software infrastructure is very similar to the one sited on the service discovery, we explain it while describing the service discovery architecture.

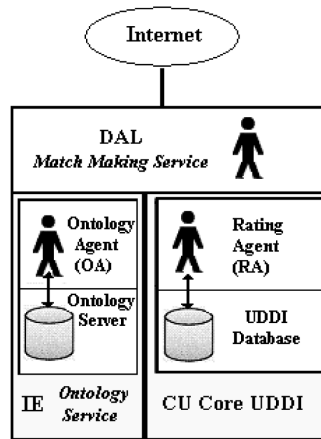
5 Service discovery architecture

The service discovery of service-oriented paradigm is not yet mature. Current technology of service discovery is only a data source for maintaining the industrial identification

information of the registered services and does not expose any further useful information for assisting service requesters to decide about choosing the most appropriate services. To provide a more useful and effective service discovery and switch it from a pure database to an intelligent assistant entity, the model envisions a service discovery as a multi-layer structure with service-oriented standards, particularly the web services standard in service discovery as UDDI, integrated with software agents.

As enterprise architecture, an agent-based interface named as ‘DAL’ is sited on the top of the service discovery. DAL as an interface, reasons about requests and assists requesters in locating the most desired services and their providers. The ‘IE’ reasons about the semantics of requests. In fact, IE is a supporting unit for the service discovery to provide ontological services. The ‘Core UDDI (CU)’ is the current technology of the service discovery in web services additionally equipped with an agent-based entity incharge of reasoning about the qualification, reputation and ranking of a specific registered service. The architecture of a service discovery is depicted in Figure 3.

Figure 3 An agent-based service discovery



5.1 DAL, match-making service

The DAL moves a service discovery from a silent database to an active and useful agency assisting customers in locating their favourite services and forming an efficient and a reliable collaboration. To handle the above-mentioned features, DAL keeps track of services' profiles and updates the information saved in each of them in terms of performances and feedback information of the former service consumers. The collected feedback information represents the credential and reputation levels of services. Moreover, this information is publicly accessible to all service locators to assist them in choosing a service among a number of candidate services with lower risk. We follow up our routine and look at DAL from different perspectives:

- As an agent, DAL is an assistant agent collaborating with other internal units mainly Rating Agent (RA) towards helping service locator and capturing the profile of the most appropriate service.
- As a service, DAL is a matchmaker service coupling service requesters with suitable services and their providers.

DAL should be knowledgeable about service discovery standards such as UDDI, WSDL and SOAP so that it can be able to parse messages as requests. As some other minor activities of DAL, we may stress routing messages to their receivers either inside or outside service discovery and managing internal sub-units of the service discovery.

5.2 *IE, ontology services*

Effective and consistent communication and interactions among involved enterprises in collaborations require syntactic and semantic agreement about domain ontology. Ontology and especially semantic web are the new emerging research areas. Researchers believe that service automation without the semantic web if is not possible, indeed is limited. The needs for information integration in general and autonomous service composition in specific push both academia and industry to pay more attention to ontology and the semantic web.

Considering the above argument, our model envisions an '*Ontology Server*' administrated by an '*Ontology Agent*'. The ontology server facilitates knowledge representation, whereas the ontology agent manages the ontology server and reasons about the meaning of the terminologies.

From a service-oriented perspective, the IE is a 'Context Service' that provides some features for services and their enterprises in terms of semantics and ontology. As a simple scenario, suppose the ontology agent receives a term included in a request, the ontology agent interprets it into a specific format named as 'context information' and sends this information to the ontology server to analyse. The process is based on the classification of domain ontology that enables IE to realise the meaning of the term.

The success of the semantic reasoning of the IE depends on the following issues:

- the interpretation power of the Ontology Agent in producing suitable contextual information and
- the diversity of designed classification of domain ontology.

5.3 *Core UDDI (CU), data service*

The Core UDDI consists of two sub-units named as the UDDI database and the 'RA'. As the UDDI database is the same as the service discovery standard in web services, we skip its explanation. RA reasons and provides some information regarding qualification, performance and reputation of the registered services. In addition, RA keeps track of services' profiles and updates their ratings, QoS and response times. From the implementation point of view, this information might be as a vector of parameters each representing a qualification issue. As a main benefit of situating this information on the service discovery is that the information will be publicly accessible for all service requesters. We may look at the RA as a service measuring the rating of registered services.

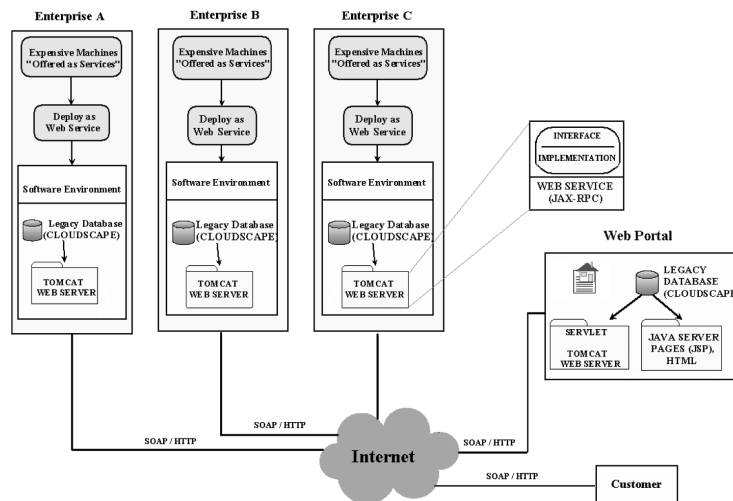
6 **Prototype implementation**

Considering the complexity of the model and as a proof of concept, a simple version of the model has been developed. The prototype implements manufacturing resources sharing among collaborative enterprises. In terms of collaboration, the involved

enterprises form a cooperative distributed system, producing new services and ultimately satisfying market demands. Figure 4 depicts a scenario of the developed prototype briefly. The prototype consists of the following active services or agent-based entities as involved enterprises:

- a number of enterprises as service providers offering manufacturing (resources) services such as EDM and CNC
- a web portal knowledgeable of offered services and their providers, which behaves as a gateway. Hence, any customer interested in using manufacturing resources sends requests to the service providers and interacts with them throughout this gateway and
- a service requester, as the customer.

Figure 4 A manufacturing resources sharing scenario

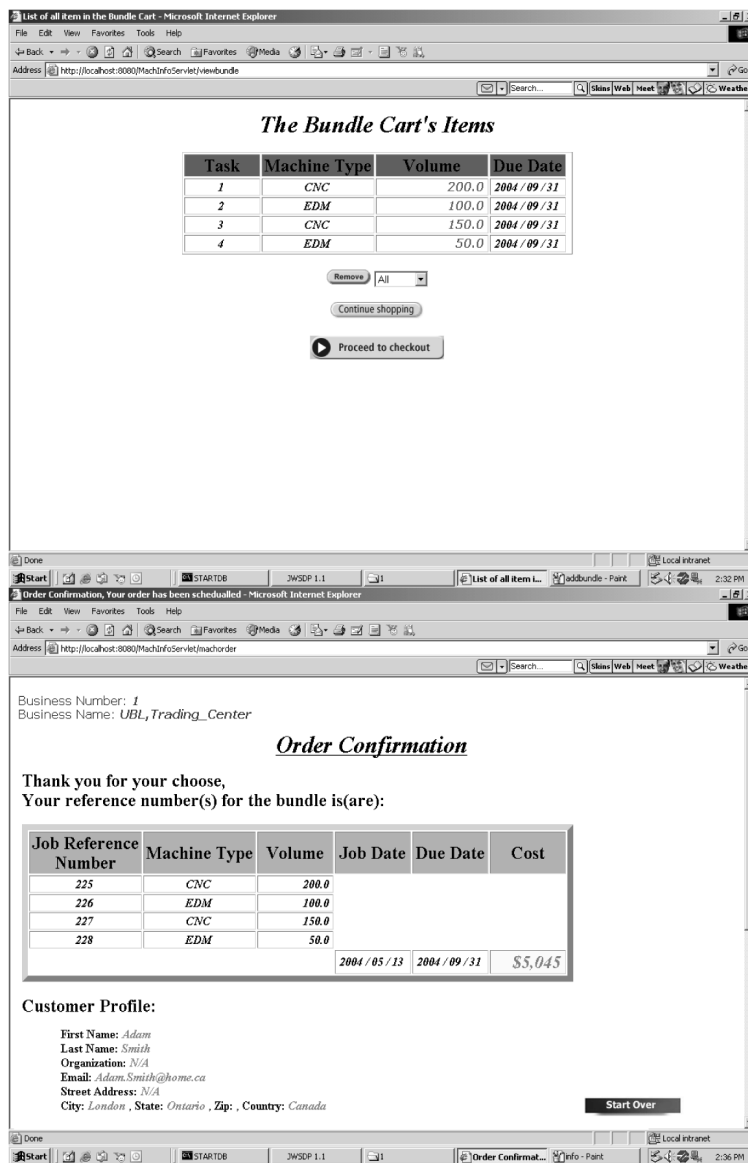


As a simple scenario, suppose some enterprises that possess a number of expensive machines would like to offer them as services to customers. Offering simply means accepting orders via the internet and scheduling the orders as machines' jobs with respect to resource availability. Any customer, who has a manufacturing job and is interested in using these resources, communicates with the web portal and asks for the enterprises' bids. The agent-based web portal, which is knowledgeable about services, their locations and providers, multicasts the customer's message as 'Call for Bid' to the enterprises by sending SOAP messages. Consequently, each service provider based on its local database as knowledge, facilities and resource availabilities proposes its bid including due date and cost to the web portal. The bids are simply SOAP messages submitted through web services to the web portal. As a result, the web portal analyses the received bids and suggests the most appropriate bid to the customer to choose as a business partner. The customer by receiving the suggested bid and with respect to its internal policy chooses the most suitable one and submits its orders to the enterprise directly. Eventually, the customer starts a collaboration with the chosen enterprise over the internet.

As the prototype represents a service-oriented environment, the privacy of each service provider is protected by defining specific web services, private databases, internal policies and local protected configuration files.

The software prototype runs under the Windows NT/2000. Java API for XML-based RPC known as JAX-RPC (<http://java.sun.com/xml/jaxrpc/>) has been used to create and deploy the web services. The services have been deployed on the Apache Tomcat container (<http://jakarta.apache.org/tomcat/>). The web portal is developed by Java Servlet deployed on Apache Tomcat Web container. Figure 5 depicts two snapshots of the developed prototype.

Figure 5 Two snapshots of the developed prototype



7 Conclusion and future work

Global competition forces SMEs to collaborate and share resources to enable them for both competition and market satisfaction. However, in terms of speed and quality, the traditional approaches and technologies for collaboration are not effective any more. Enterprise integration and collaboration needs to be fast and precise; therefore, as an alternative choice, enterprises have to adopt some other media such as the internet as new and fast collaborative environment.

This paper proposes an architectural solution for enterprise collaboration, particularly for manufacturing resources by using web services and software agents. On the service discovery side, we defined a DAL behaving as an interface between the service discovery and service locators. We argued the needs for an ontology server and how adopting semantic web technology may improve the efficiency and expected functionalities of service discovery with respect to semantics and meaning of the requests rather than their syntax.

On the enterprise side, we present a multi-layer architecture consisting of some units to take part in playing the required roles in any collaboration so that they fulfill the essential functionalities such as negotiation, designing workflows and processes, planning, coordination, evaluating and replacing the weak business partners with the strong ones.

This paper reports some results of our ongoing research and development work. Among our main concerns, as future contribution is implementing an ontology server, providing a generic rating algorithm for supporting user-defined attributes, developing an autonomous service composer as well as addressing the security and trust in distributed systems such as virtual enterprises, which are particularly important for industrial applications.

References

- Barbuceanu, M. and Fox, M.S. (1995) 'The architecture of an agent based infrastructure for agile manufacturing', *Proceedings of IJCAI'95 Workshop on Intelligent Manufacturing*, Montreal, Canada.
- Camarinha-Matos, L.M. and Afsarmanesh, H. (1999) *Infrastructure for Virtual Enterprise*, Norwell, MA: Kluwer Academic Publishers.
- Chang, J. and Scott, C. (1996) 'Agent-based workflow: TRP support environment (TSE)', *Computer Networks and ISDN Systems*, pp.1501–1511.
- Cheng, Z., Singh, M.P. and Vouk, M.A. (2003) 'Composition constraints for semantic web services', *Proceedings of WWW2002 Workshop on Real World RDF and Semantic Web Application*.
- Fox, M.S., Chionglo, J.F. and Barbuceanu, M. (1993) 'The integrated supply chain management system', Internal Report, Department of Industrial Engineering, University of Toronto.
- Jennings, N.R., Faratin, P., Johnson, M.J., Norman, T.J., O'Brien, P. and Wiegand, M.E. (1996) 'Agent-based business process management', *International Journal of Cooperative Information Systems*, pp.105–130.
- Jin, T. and Goschnick, S. (2003) 'Utilizing web services in an agent based transaction model (ABT)', *Proceedings of Workshop on Web Services and Agent-based Engineering*, Melbourne, Australia.
- Li, Y., Ghenniwa, H. and Shen, W. (2004) 'Agent-based web services framework and development environment', *Computational Intelligence*, Vol. 20, No. 4, pp.678–692.

- Maamar, A., Sheng, Q.Z. and Benatallah, B. (2003) 'Interleaving web services composition and execution using software agents and delegation', *AAMAS'03 Workshop on Web Services and Agent-based Engineering*.
- Marik, V. and Pechoucek, M. (2003) 'Agent technology for virtual organization', in L.M. Camarinha-Matos and H. Afsarmanesh (Eds). *Processes and Foundations for Virtual Organizations*, Kluwer Academic Publishers, pp.243–252.
- Maximilein, E. and Singh, M.P. (2002) 'Conceptual model for web service reputation', *ACM SIGMOD Record*, Vol. 31, No. 4, pp.36–41.
- Maximilein, M. and Singh, M.P. (2001) 'Reputation and endorsement for web services', *ACM SIGecom Exchange*, Vol. 3, No. 1, pp.24–31.
- Park, H., Tenenbaum, J. and Dove, R. (1993) *Agile Infrastructure for Manufacturing Systems: a Vision for Transforming the US Manufacturing Base*, Defence Manufacturing Conference.
- Peng, Y., Finin, T., Chen, H., Wang, L., Labrou, Y., Cost, R.S., Chu, B., Cross, V., Russell, M., Tolone, B., Boughannam, A. and McCobb, J. (1999) 'An agent system for application initialization in an integrated manufacturing environment', *Proceedings of SCI'99/ISAS'99*, Orlando, FL, Vol. 7, pp.415–421.
- Peng, Y., Finin, T., Labrou, Y., Chu, B., Long, J., Tolone, W.J. and Boughannam, A. (1998) 'A multi-agent system for enterprise integration', *Proceedings of PAAM'98*, London, UK, pp.533–548.
- Rabelo, R.J., Afsarmanesh, H. and Camarinha-Matos, L.M. (2001) 'Federated multi-agent scheduling in virtual enterprises', *Proceedings of PRO-VE'01*, pp.145–156.
- Shen, W., Hao, Q., Yoon, H.J. and Norrie, D.H. (to appear) 'Applications of agent systems in intelligent manufacturing: an updated review', *International Journal of Manufacturing Technology and Management*.
- Shen, W., Norrie, D.H. and Barthes, J.P. (2001) *Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing*, London, UK: Taylor and Francis.
- Shen, W., Xue, D. and Norrie, D.H. (1998) 'An agent-based manufacturing enterprise infrastructure for distributed integrated intelligent manufacturing systems', *Proceedings of PAAM'98*, London, UK, pp.533–548.
- Singh, M.P. (2003) 'Distributed enactment of multi-agent workflows: temporal logic for web service composition', *Proceedings of the Second International Conference on Autonomous Agents and Multi-Agent Systems*, pp.907–914.
- Yan, Y., Maamar, Z. and Shen, W. (2001) 'Integration of workflow and agent technology for business process management', *Proceedings of CSCWD 2001*, London, Ontario, Canada, pp.420–426.
- Zeng, L., Ngu, A., Benatallah, B. and O'Dell, M. (2001) 'An agent-based approach for supporting cross-enterprise workflows', *Proceedings of the 12th Australian Conference on Database Technologies*, pp.123–130.