AGENT-BASED WORKFLOW MODEL FOR ENTERPRISE COLLABORATION

By

Shuying Wang

SUBMITTED IN PARTIAL FULLFILLMENT OF THE
REQUIRMENTS FOR THE DEGREEE OF
MASTER OF SCIENCE
AT
LAKEHEAD UNIVERSITY
THUNDER BAY, ONTARIO, P7B 5E1 CANADA
APRIL 2005

Your file Votre référence
ISBN: 0-494-10678-6
Our file Notre référence
ISBN: 0-494-10678-6

Acknowledgements

I would like to acknowledge the direction that I received from my supervisor, Professor Weiming Shen. He contributed particularly to the conceptualization of and motivation for the problem. He also added a great deal of insight into various aspects of this work pertaining to software agent architectures, service oriented architecture, enterprise modeling, processes formalization and workflow management concerns. His advices and technical supports added substantially to the completeness of this work.

I would also like to extend my special thanks to Dr. Qi Hao whose valuable supports and suggestions have greatly improved the quality of this thesis.

My research was made possible by the generous support of the National Research Council's Integrated Manufacturing Technologies Institute (NRC-IMTI).

Table of Contents

Abstract

Workflow management system supports the automation of business processes where a collection of tasks is organized between participants according to a defined set of rules to accomplish some business goals. The service-orientated computing paradigm is transforming traditional workflow management from a close, centralized control system into a dynamic information exchange and business process. Moreover, agent based workflow, from another point of view, provides a flexible mechanism for dynamic workflow coordination at run time. In this context, the combination of Web services and software agents provides great flexibility to discover and establish relationships among business partners.

This thesis proposes an agent-based workflow model in support of inter-enterprise workflow management. In the proposed model, agent-based technology enables the workflow coordination at both inter- and intra- enterprise levels while semantic Web and Web services based technologies provide infrastructures for messaging, service description, service discovery, workflow ontology, and workflow enactment. Coordination agents and resource agents are used with a Contract Net protocol based bidding mechanism for constructing a dynamic workflow process among business partners. The agent system architecture, workflow models and related components are described. A prototype system is implemented for the purpose of designing and developing role-feasible agents for simulating the formation process of a virtual enterprise.

Abbreviations

AADE Autonomous Agent Development Environment

ACL Agent Communication Language

EAI Enterprise Application Integration

EDI Electric Data Interchange

ERP Enterprise Resource Planning

ebXML Electronic Business Extension Markup Language

DAML-S DARPA Agent Markup Language for Services

DL Description Logic

DF Directory Facilitator

DAML+OIL Darpa Agent Markup Language with Ontology Inference Layer

FIPA Foundation for Intelligent Physical Agent

HTTP Hypertext Transfer Protocol

OWL Web Ontology Language

OWL-S Web Ontology Language for Services

QoS Quality of Service

RDF Resource Description Framework

SOA Service Oriented Architecture

SOAP Simple Object Access Protocol

UDDI Universal Description, Discovery and Integration

List of Figures

1. Introduction

1.1 Motivation

Today, manufacturing enterprises often focus on their core-competencies and integrate different organization's workflows to quickly respond to the changing market requirements. In this context, workflow technology becomes an efficient approach to assist the collaboration among enterprises. Workflow management systems provide the automation of business process where a collection of tasks is organized between participants such as workers, departments and enterprises according to a set of predefined rules to accomplish some business goals. As a result, the workflow system can quickly react to the customer's requirements, automatically execute by the software tools, and flexibly adjust its processes to improve its efficiency.

Meanwhile, service-orientated computing is quickly becoming a widely accepted technology for enterprise collaboration, enabling business-to-business and department-to-department integration of business processes. Service-orientated architecture (SOA) also provides a loosely coupled computing framework to integrate heterogeneous system resources with the support of XML based technologies. In fact, our motivation in this research is to present an agent based workflow solution for enterprise collaboration under this context.

design, deployment, monitoring, and troubleshooting of complex tasks as a huge challenge to both system designers and developers.

- Time complexity. The development of integrated workflows is still ad-hoc, time-consuming and requires enormous efforts of low-level programming.
- Run-time complexity. Although SOA provides a framework in support of process automation, it is still not enough. We indicate the meaning of run-time flexibility in our work contains the functions of service or partner search and selection, task mediation, coordination, fault handling, workflow reconfiguration, etc. Unfortunately, neither Workflow Reference model nor workflow languages like BPEL4WS or WSFL directly support these crucial functions.

1.3 Scope

Flexible and efficient integration of workflows requires the ability to efficiently discover and exploit business services in a dynamic and constantly changing environment. As much research work has proved that agents can provide a flexible, reconfigurable and coordinated approach to enhance workflow management, the merging of software agents with workflow brings a promising solution. A software agent is an execution entity on behalf of its role, knowledge, or resources to perform tasks of a workflow. Agent-based workflow can be considered as a workflow process that is planed, performed, communicated, and coordinated in a multi-agent environment, in which the workflow is decomposed into multi-level collaborative tasks and each task represents a logical piece of work that contributes to the process.

and the other is ontology based workflow process definition. Consequently, we build the agent based workflow model in two steps:

- First, exploit and establish the relationships among software agents, Web services and workflow. Within this step, Web services are used as external behaviors of agents and software agents for process composition, resource discovery, mediation and interaction. However, deriving the specified workflow description for virtual enterprise formation is not easy, and several steps are required to finalize its concrete description. Our approach is to use an agent-based coordination model with the Contract Net Protocol (Smith, 1980) based negotiation to facilitate this formation process in order to derive an executable inter-enterprise workflow definition.
- Second, develop a semantic model that is based on the previous step to support workflow process definition, generation, reasoning and interpretation. The workflow ontology provides a set of concepts, and meta-data, that can be queried, advertised and used to control the workflow process. Using workflow ontology, agents can find alternative sub workflows, services, and resources when the predefined workflow or resources are abstract or unavailable.

1.5 Organization of the Thesis

In Section 2, we introduce the major concepts that are related to this thesis. These concepts include: enterprise collaboration, virtual enterprise, workflow model, workflow reference model, Web services, semantic Web, ontology, and software agents. Moreover, we review the related research areas in workflow modeling, agent modeling and service based process modeling.

2. Research Literature

Our research is mainly related to four technique domains: virtual enterprise (VE), workflow, semantic Web and Web services, and software agents. In this section, we briefly review these concepts and the related research literature.

2.1 Technical Background

2.1.1 Enterprise Collaboration

In response to the challenges of global competition, the new manufacturing paradigms have emerged such as virtual enterprise, extended enterprise, supply chain management and agile enterprise and challenge the way traditional industrial manufacturing systems are planned and managed. Enterprise collaboration provides a collaborative infrastructure to collect and gather business partners through a coordinate and cooperate approach. In this infrastructure, each partner shares its resources, infrastructure, personnel, research, information, and knowledge, and realizes only a special fraction of the value chain. Therefore, they, combined together, have a clearly synergetic effect to gain a better market position.

The typical forms of enterprise collaboration (except virtual enterprise) are described as follows:

Companies, especially the Small and Medium size Enterprises (SMEs) must join skills and resources in order to survive and gain competitive advantage in a global market environment. Moreover, a dynamic virtual enterprise only emerges when a customer approach with an order and business partners build a temporary relationship for the order processing. Main characteristics of a VE are described as follows:

- Temporary network of independent enterprises or individuals
- Collaboration among the partners
- Goal-oriented
- Commitment-based
- Shared skills, costs, profits, risks and markets
- Geographically distributed
- Limited life span

When analyzing the infrastructure requirements for virtual enterprise, it is important to consider the various phases of its life cycle. Figure 1 represents a life cycle model including the creation, operation, evolution and dissolution stages (Camarinha-Matos and Cardoso, 1999b).

Creation. In the whole life cycle of the virtual enterprise, the formation is an important phase including partners' search and selection, bid negotiation, configuration, contract, etc.

Operation. This is the phase when the VE is performing its business processes in order to achieve its common goal(s), and which requires functionalities such as: data exchange, order management, order processing, distributed the dynamic planning and scheduling, distributed task management and high levels of task coordination.

other. The communication can involve either simple data passing or two or more services coordinating.

The main characteristics of SOA are loosely coupled computing, messaging, and asynchronous communication. Loosely coupled computing allows a service to be accessed using any arbitrary document format, regardless of syntax, providing the physical business data within the document contents. "Messaging is a technology that enables high-speed, asynchronous, program-to-program communication with reliable delivery." (Hohpe et al., 2004). Asynchronous communication provides a flexible approach for messages to be sent and received: once the message is sent, the sender is free to perform other work and does not have to wait for the receiver to receive and process the message.

The collaborations in a service-oriented architecture follow the so-called "find, bind and invoke" paradigm where a service consumer performs dynamic service location by querying the service registry for a service that matches its criteria. If such a service exists, the registry provides the consumer with the interface contract and the endpoint address of the service. The following diagram (Figure 3) illustrates the entities in a service-oriented architecture that collaborate to support the "find, bind and invoke" paradigm.

a service description from other resources besides a service registry, such as a local file, FTP site or Web site.

Web services are a technology that is well suited to implement a service-oriented architecture. In essence, Web services are self-describing and modular applications that expose business logic as services that can be published, discovered, and invoked over the Internet. Web services can be developed as loosely coupled application components using any programming language, any protocol, on any platform. This facilitates the delivery of business applications as a service accessible to anyone, anytime, at any location, and using any platform. The basic components of Web services are SOAP, WSDL and UDDI.

The Simple Object Access Protocol (SOAP) defines an XML document format that describes how to invoke a method of a remote method. XML together with XSLT, XML Schema, XML Parsers are among the first attempts to solve the data-exchanging problem between heterogeneous web applications. SOAP is a simple and lightweight XML-based messaging mechanism for exchanging structured data. SOAP consists of an envelope, a set of encoding rules, a convention for representing remote procedure calls (RPCs) and bindings with low-level protocols. SOAP can be used in combination with or re-enveloped by a variety of network protocols such as HTTP, SMTP, FTP, RMI over IIOP or MQ.

The Web Services Description Language (WSDL) is an XML vocabulary that describes a Web service. The information in the WSDL file defines the name of the web service, the names of its methods, the arguments to those methods, and other details.

The Universal Description, Discovery, and Integration (UDDI) protocol defines a SOAP interface to a registry of Web services. For a piece of code that a service provider supplement the data widely available in today's Web with "semantic markup" to allow smarter navigation and use of Web data. The semantic web aims for machine-understandable Web resources, whose information can then be shared and processed both by automated tools and by human users. Figure 3 shows the static part of the Semantic Web.

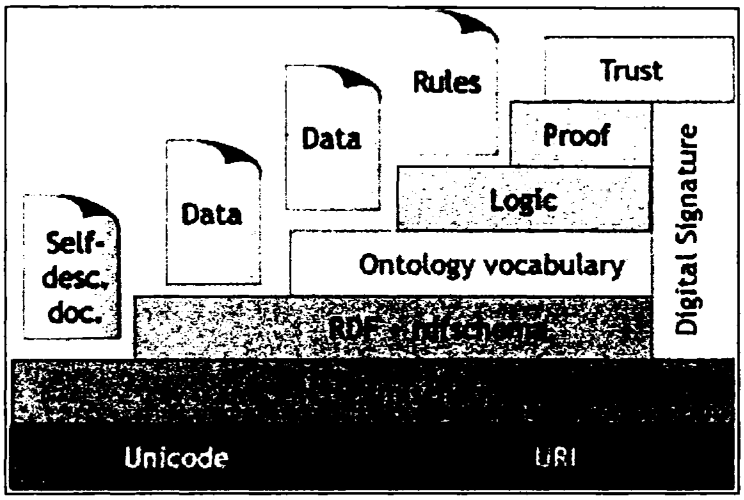

Figure 3. Semantic Web diagram

- Unicode and URI are used as a basis.
- XML provides a surface syntax for structured documents but imposes no semantic constraints on the meaning of these documents. XML Schema is a language for restricting the structure of XML documents.
- RDF is a data model for objects ("resources") and relations between them, and provides a simple semantics for this data model represented in an XML syntax.
- RDF Schema (RDFS) extends RDF by class and property hierarchies that enable the creation of simple ontologies. RDF and RDFS are already standardized by the W3C.

Language), etc., have been proposed on top of those basic components for automating business process management. They are specialized languages for describing some aspects of the workflow and representing the business process logic, defining the sequence of tasks and the routing rules implemented by the workflow system.

Among these languages, BPEL4WS is being widely studied for providing a formal specification of business processes and business interaction protocols. A workflow described in BPEL4WS details the flow of control and data dependencies among a collection of Web services being composed. When enacted, the composition itself becomes available as a meta-Web service, eligible for inclusion in other compositions. BPEL4WS uses the executable business process and the abstract process to ensure that different business processes can understand one another in a Web services environment, and that they can realize a dynamic composition. In fact, there is no direct logical relation between an abstract process definition and an executable process definition except that developers manually define such a relation for them.

OWL-S is an OWL ontology evolved from DAML-OIL for Web services. It describes a set of classes, properties and relations that are specific to the description of Web services. The OWL-S consists of three service models: the service profile model, the process model and the grounding model.

- The profile model describes the properties of a service necessary for automatic discovery, such as what the service offers, its inputs and outputs, and its preconditions and effects.

- *Pro-activeness*: software agents take initiatives and exploit unexpected opportunities where appropriate.
- *Responsiveness*: agents perceive their environment and respond in a timely fashion to change.

The integration of software agents and Web services can be proposed at both the design level and implementation level. At the design level, we encapsulate Web services as agent models so that each agent is on behalf of Web services in its actions and relations to the environment. In this sense, we could treat a Web service as a semi-autonomous agent. On the other hand, Web services can be described as external behavior of software agents. Therefore, agents could be used to build high-level models with flexible interaction patterns, while Web services are more suitable for solving interoperability among various applications in real implementation. At the implementation level, the UDDI, WSDL and SOAP provide the capability of service discovery, deployment and communication, and Web services-based workflow specifications, such as WSFL, XLANG, BPEL4WS, enable the workflow automation. Behind Web services, software agents coordinate the whole workflow process.

2.1.7 Coordination in Multi-Agent System

Multi-agent systems are systems composed of agents coordinated through their relationships with one another. In multi-agent systems, groups of software agents communicate and negotiate shared information and responsibilities in order to accomplish a collective task.

various agent applications such as e-commerce, pervasive computing, workflow management system, integration of heterogeneous information sources and databases.

The mediator agents play the role of system coordinators by promoting cooperation among intelligent agents and learning from the agents' interactions. The mediator agents provide the computational simplicity and manageability by imposing a static or dynamic hierarchy for every specific task.

According to the similar functionalities of software agents, it is very useful to compare the concepts and functions of broker agent and mediator agent. As a matter of fact, such a comparison is based on the context of workflow. In a workflow process, both broker agent and mediator agent play the functions of coordination, mediation, and workflow management. However, there are many differences between broker agent and mediator agent: a broker agent finds a order solution but customers make a decision on it; moreover a broker agent does not have explicit knowledge of its environment like services or agents quantity but knows how to find it; however, a mediator agent within an enterprise has the knowledge to its system and resources, therefore it could make a plan or decision by using its knowledge with some strategies. As a result, we use the broker agent or the mediator agent at different situations: the broker agent is very useful acting in a dynamic environment where the resources may be continually changing while the mediator agent focuses on scheduling in a resource-fixed environment.

2.1.8 Workflow and Workflow Model

The workflow concept has evolved from the notion of process in manufacturing or the office enviroment. The processes exist to increase efficiency by concentrating on the skills of the individuals or information systems) that can perform the specified tasks.

- Workflow management system (WfMS) is a software application that stores process definitions and runs jobs based on those process definitions.

The workflow model has task granularity and can be grouped as a task-flow based workflow. The task acts as the centerpiece of the workflow. At the top level, the workflow system can support a set of processes. Each process can be depicted as a set of activities. An activity can be defined as a single task or a set of low-level tasks. The transformation of data or the invocation of some services can define a task. One task or activity can be dependent on the completion of another task or activity.

A workflow process consists of two phases as business process modeling and the execution of workflow. The business process modeling is to translate a business process from the real world into a formal, computer processable definition by the use of analysis, modeling and system definition techniques. The resulting definition is sometimes called a process model or process definition. At execution phase, the process definition is interpreted by software which is responsible for creating and controlling operational instances of the process, scheduling the various activities steps within the process and invoking the appropriate human and IT application resources, etc.

The advantage of a workflow management system is as follows:

- Improved efficiency: automation of many business processes results in the elimination of many unnecessary steps.

- Better process control: improved management of business processes achieved through standardizing working methods and the availability of audit trails.

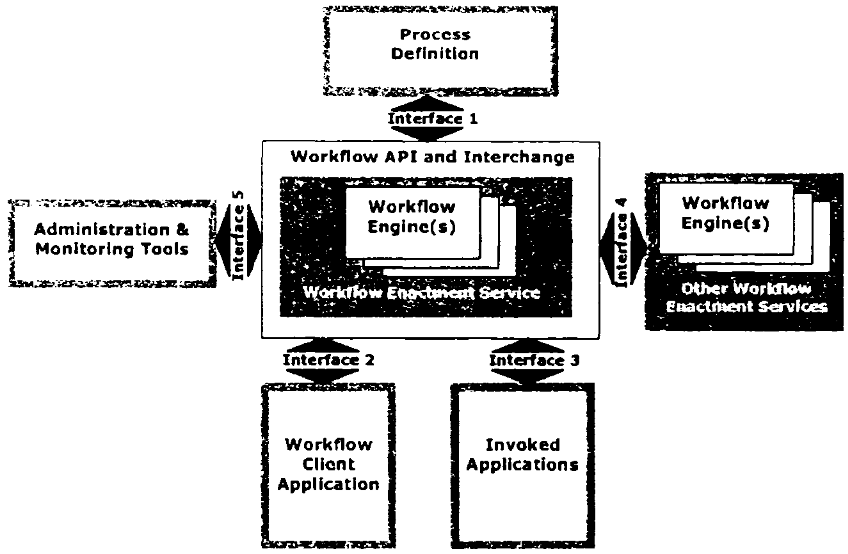

Figure 4. Workflow reference model

Invoked applications: the programs that may be invoked by the workflow system. These programs include interfaces to and between legacy data systems and Enterprise Application Integration (EAI). They also often include image systems, document management systems, and mail systems.

Administration and monitoring: it includes both the history of each case and the monitoring of the total work performed. Individual cases might include the person and processor whose work was assigned the work with the date, time, and disposition. This could be used for reporting and auditing, but could also be used for process control.

The workflow enactment service: it provides the run-time environment in which process instantiation and activation occurs, utilizing one or more workflow management engines, interpreting and activating part or all of the process definition, and interacting with the external resources necessary to process the various activities.

models or to support the representation of additional attributes. Although Petri nets are not directly used, the basic concepts and mechanisms proposed by this formalism can be found in the background of many other tools used in workflow and business process management.

The Unified Modeling Language (UML) is an expressive language for a shared workflow representation. UML has a notation and a well-defined set of syntactic and semantic rules (Eriksson and Penker, 2000). The UML has a great deal of tool supports, which makes it easy to integrate into the workflow framework. The UML is also flexible enough to capture functional and non-functional constraints.

2.2.2 Agent Modeling

2.2.2.1 Agent Encapsulation

Among different approaches used for agent encapsulation in agent-based manufacturing systems, two approaches are distinct: the functional decomposition approach and the physical decomposition approach.

In the functional decomposition approach, agents are used to encapsulate modules assigned to functions such as order acquisition, planning, scheduling, material handling, transportation management, and product distribution. There is an explicit relationship between agents and functional entities. Examples of this type of approach are ISCM (Mark et al., 1993), CIIMPLEX (Peng et al., 1998), and in some recent projects (Choi et al., 2003, 2004; Kuo, 2003).

In the physical decomposition approach, agents are used to represent entities in the physical world, such as operators, machines, tools, fixtures, products, parts, features, and

Brokers are similar to facilitators with some additional functions such as monitoring and notification. The functional difference between a facilitator and a broker is that a facilitator is responsible only for a designated group of agents, whereas any agent may contact any broker in the same system for finding service agents to complete a special task. Broker agents can be found in CIIMPLEX.

The Mediator approach is another type of federation architecture. In addition to the functions of a facilitator and a broker, a mediator assumes the role of system coordinator by promoting cooperation among intelligent agents and learning from the agents' behavior. A detailed description of the mediator concept and architecture can be found in (Maturana et al., 1999). Applications using mediators in intelligent manufacturing systems can be found in (Maturana et al., 1999; Shen et al., 2001).

The Autonomous Agent approach is different. Although different definitions have been proposed for autonomous agents, we argue that an autonomous agent should have at least the following characteristics: (1) it is not controlled or managed by any other software agents or human beings; (2) it can communicate/interact directly with any other agents in the system and also with other external systems; (3) it has knowledge about other agents and its environment; (4) it has its own goals and an associated set of motivations. DIDE used this approach for developing agent-based engineering design systems (Shen and Barthès, 1996). AARIA also used the Autonomous Agent approach, but with fixed negotiation protocols (Parunak et al., 1998).

2.2.2.3 Agent Coordination and Negotiation

Coordination is central to the successful operation of agent-based manufacturing systems which are very complex and whose stability is essential. Without coordination, a group of

(Sycara et al., 2004) proposed a multi-agent architecture for dynamic discovery and coordination of agent-based semantic Web services. They focus on the broker agent with the support of OWL-S for dynamic services discovery, mediation and coordination. The authors also describe OWL-S's extensions, detail their implementation's basic features, and explain how these features address the broker's reasoning problem.

(Zeng et al., 2001) presents an agent-based architecture to support the cross enterprise workflow. The authors describe each of the components of this framework (partially implemented) and how certain agent characteristics and capabilities have relevance to specific operational concerns in the cross enterprise workflow domain.

(Xu et al., 2003) propose the use of a multi-agent system for supply chain management with respect to the perspectives such as planning, control, and execution. In addition, this approach considers nonfunctional concerns such as flexibility and reliability and incorporates the use of industry -standard modeling techniques using the Unified Modeling Language (UML). There is also a positive evaluation of the approach based on the impact on throughput time.

Other advanced approaches to dynamic composition of cross-organization workflows project such as LARKS. Katia Sycara proposes an Agent Capability Description Language (LARKS) for describing the capabilities and requests of software agents and a mechanism to determine the structure and semantic match description in this language. They have implemented LARKS with associated matchmaking process in RETSINA multi-agent infrastructure framework.

Nowadays, there is no commercial agent based workflow product. But several prototype systems exist in this research area, namely ADEPT (Advanced Decision

Some researchers have tried to combine the OWL-S with BPEL4WS to use both of their merits. (Liu et al., 2004) build a direct mapping from DAML-S to BPEL4WS for composite processes and atomic processes. Moreover, because agents provide great flexibility in service discovery and interaction in a dynamic workflow process, there are also some research efforts laid upon the use of agents for the composition of Web service based workflow. Korhonen (Korhonen et al., 2003) describes a design of a Web service based agent planner to automatically compose Web service based workflow. Vieira (Vieira et al., 2004, Naveen et al., 2004) presents an ontology-driven architecture to provide a flexible workflow execution. This architecture supports a mechanism to handle presuppositions and chooses alternative sub workflow by defining sub workflow ontology.

- Agent-based workflow (Chang and Scott, 1996; Yan et al, 2001) can be considered as a workflow process that is planed, performed, communicated, and coordinated in a multi-agent environment.
- The workflow is decomposed into multi-level collaborative tasks and each task represents a logical piece of work that contributes to a workflow process.
- The software agent is an execution entity on behalf of its role, knowledge, or resources to perform tasks of workflow.
- The workflow is used to control the interactions between agents.

The inter-enterprise workflow process is across several units in a company or even spreads across several companies. However, a central workflow engine is unable to get all the information of the whole business process in order to control it. Thus the solution is to have one workflow engine residing in each organization or unit and the whole business process is fulfilled through interactions among the multiple workflow engines. In this scenario, agents take full responsibilities of a workflow management system to analyze, automate, integrate, communicate and interact workflow systems.

The agent based workflow technology has the following benefits:

- Providing distributed system architecture. There are several system architectures in multi-agent systems, which can be used in a distributed system for implementing a workflow management system.
- Providing communication methods. Agent communication langue has been widely studied. There are several communication languages based on semantic messages, such as Knowledge Query Manipulation Language (KQML) and

In general, agent-based workflow model takes advantages of agent coordination mechanism. Agent coordination mechanism is a description of actor and role that an actor playing a particular role carries out a set of activities (Weiss, 2000). In fact, coordination model (Omicini et al., 2001) uses a coordinated problem-solving method to solve workflow management where a coordination agent performs the functions of planning, coordinating and managing a process while users and resource agents execute the proposed task. To simplify the workflow, we abstract the workflow process as the enactment of several role players. The main role player in the coordination model is called the workflow coordination agent, which is a software agent responsible for coordinating the workflow process. A coordinative agent-based workflow model is proposed with four layers: workflow management layer, coordination layer, service layer and resource layer (Figure 5).

The workflow management layer contains workflow definition tools that are mainly used to create business process representations, including the definition of business processes, rules, and logics. A workflow engine is a software service that provides a run-time execution environment for a workflow instance, which is capable of initiating utilities to activate appropriate applications for the execution of particular activities.

The coordination layer is the second layer whose responsibility is to coordinate the activities at multiple enterprise levels. In this layer, the broker agent coordinates the activities of workflow at inter-enterprise level and the mediator agent at intra-enterprise level. At inter-enterprise level, when the customer order triggers the co-operative processes, the broker agent selects one workflow pattern, queries ontology agent for potential partner list and sends a call–for-bid on SOAP to potential suppliers. After

At the resource layer, workflow management is implemented by enterprise business systems, human operators, customers, machines, and other applications.

3.3 Agent-based Workflow Model

The key issue to realize a flexible, dynamic workflow process is how to model the workflow properly at design time and how to provide a mechanism to construct a workflow specification for each instance at run-time. To accomplish such purposes, we provide two models: the semantic model to describe workflow ontology for workflow process definition and reasoning; the process model to support process composition, resource discovery, mediation and interaction for dynamic workflow implementation.

3.3.1 Semantic Model

Description logic (Baader et al., 2003) is a family of languages that formally express certain constraints on knowledge representation. Since they have a precise semantics and axiomatization, there are amenable to automatic processing in a manner that is unambiguous across implementations. Description logic begins with primitive concepts and defines further concepts in terms of formal descriptions and then concepts are computed from these descriptions. OWL is a description logic language for describing the web ontology. Although OWL is not specific to workflow process, it has the ability of specifying the process logic. OWL-S is obvious a good example. OWL-S provides three service models for process description and reasoning, advertisement and deployment in a Web services environment. In fact, our semantic model is built upon this basis and extends the concepts towards the whole workflow process domain.

organizational viewpoint; while an ES is executed by outside systems or other enterprises.

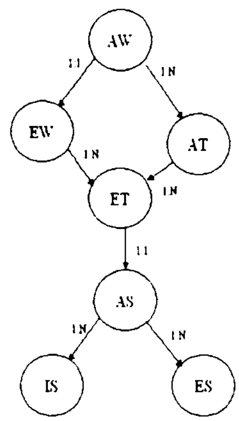

Figure 6. Workflow ontology diagram

As shown at Figure 6, our core workflow ontology entities are defined as follows:

Abstract workflow (AW): AW defines an abstract workflow ontology which describes an abstract workflow process. An AW could be an execution workflow or a collection of abstract tasks. For each particular AW, there is only one sub class instance.

Execution workflow (EW): EW is a concrete workflow ontology that defines a detailed workflow procedure. An EW could be composed into a set of execution tasks without the definition of abstract tasks. Due to the elimination of the logic reasoning procedure, EW is efficient during the execution but less flexible in the run time configuration.

Abstract Task (AT): AT is an abstract task ontology entity that has the property of task name, task type, task deadline, task pre- and post- conditions, task status, and task execution result. An abstract task represents a composite task that has at least one atomic task (i.e. execution task).

example, we define a simple abstract order task which has two sub execution tasks: Task1 and Task2. The Task2 will be triggered if Task1 is executed successfully. We use OWL to express this example as follows:

```
<?xml version="1.0"?>
<rdf:RDF
 xmlns="http://mySite.com/myOntology#"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:owl="http://www.w3.org/2002/07/owl#" xml:base="http://mySite.com/myOntology">

 <owl:Class rdf:ID = "Order">
   <owl:unionOf rdf:parseType = "Collection">
   <owl:Class rdf:about = "#Task1" />
   <owl:Class rdf:about = "#Task2" />
   </owl:unionOf>
 </owl:Class>

 <owl:Class rdf:ID ="Task2">
    <rdfs:subClassOf rdf:resource = "#Order" />
    <owl:intersectionOf rdf: parseType ="owl:collection">
          <owl:Class rdf:about = "#Task1"/>
          <owl:Restriction>
                <owl:onProperty rdf:resource="#status"/>
                <owl:hasValue rdf:resource="#done"/>
          </owl:Restriction>
      </owl:intersectionOf>
 </owl:Class>

 <owl:Class rdf:ID="Task1">
   <rdfs:subClassOf rdf:resource="#Order" />
     ...
 </owl:Class>
</rdf:RDF>
```

Figure 7. Ontology example represented in OWL DL.

model is built upon the basis of process composition, resource discovery, mediation and coordination.

3.3.2.1 Process Composition

A composition process is a process that contains process decomposition and process integration. Comparing with the process of workflow ontology generation that uses the bottom up approach, the process decomposition uses the top down approach in which, given an overall task or goal, the system decomposes it into smaller, more manageable pieces. In our case, the process decomposition is separated to four steps as follows:

- First, a semantic description of the existing Web services and the workflow process is defined in the workflow ontology repository. This work has been done in the semantic model.
- Second, the service requester asks the service provider for a workflow process or a service. The service provider will search the related workflow ontology in the service repository and select an appropriate result to service requester. We call this process "resource discovery".
- Third, the service requester checks whether the request can be satisfied. The request result will change the process status and finally affect workflow process result. We call this process "mediation".
- The last step is interaction process between the service requester and the service provider. In fact, we separate the interaction into two level meanings: the low level communication provides a message channel while the high negotiation assures the workflow to achieve the goal to some extent.

- The initial step for service providers is to advertise their services to the service registry. Information about these services is stored in a registry until the providers decide to stop advertising their services. At the service requester's side, a service query is created and communicated to the service registry.
- The service registry compares the requested capabilities with the advertised capabilities to determine if they are sufficiently similar. Sufficient similar means that the advertised service satisfies the requested capabilities. Requested and advertised capabilities must thus be formulated so that they can be compared. The service registry makes the service matchmaking, which is based on ontologies that the capabilities are annotated with.
- Finally, the service registry selects the advertised service that best match the requested capabilities. These matches are then reported back to the service requester who makes a decision on which service to use.

The service discovery can be separated to internal service discovery and external service discovery. The internal service discovery is to find services that come naturally from the internal workflow system. The external service discovery is mostly searched from outside Web services. Generally, there are two approaches to find the service – search UDDI registry or OWL-s profile. UDDI is a registry that allows business to describe and register their Web services. It also allows business to discover services. However, the search mechanism supported by UDDI is limited to keyword matches and lack of semantic support. To enable more sophisticated matchmaking presupposes, the OWL-S profile model provides a richer service description.

common agent interactions, such as auctions, iterated contract-net, purchasing, etc. The contract net protocol (Figure 8) is one of the interaction protocols for cooperative problem solving among agents. It is modeled on the contracting mechanism used by businesses to govern the exchange of goods and services. A contract net process is described as follows:

- A coordination agent announces a task that needs to be performed.
- The coordination agent receives and evaluates bids from potential contractors.
- The coordination agent awards a contract to a suitable contractor.
- The coordination agent receives and synthesizes results.

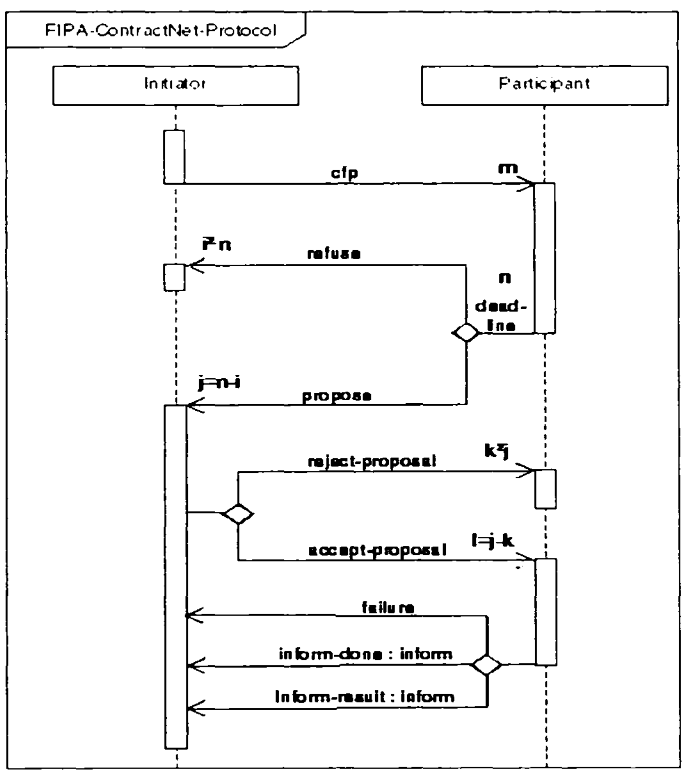

Figure 8. FIPA Contract-Net diagram

The interaction capability, both among agents and between agents and their environment, is one of the basic characteristics of an agent. The definition of high-level

3.4.1 Agents System for Inter-enterprise Collaboration

Web portal is a web interface that is designed for users to access and mange related information. From the web portal, the customers register their information, input orders, select bids and manage the contract information. Behind web portal, the order is received and processed by the broker agent.

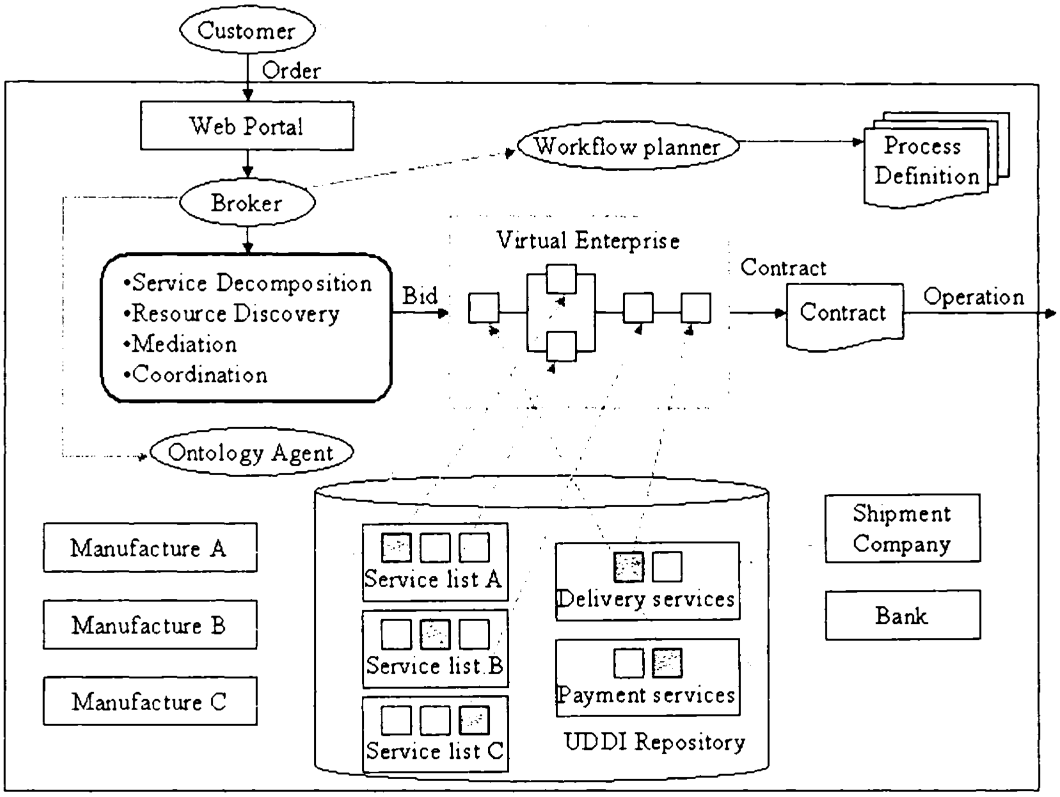

Figure 9. Agent based workflow model at Inter-enterprise level

Workflow planner is a software agent that is responsible for coordinating the workflow process. The workflow planner is responsible for maintaining upper levels of workflow ontology including abstract workflow ontology, execution workflow ontology, abstract task ontology, and task ontology. The abstract workflow ontology and the execution workflow ontology are defined at the workflow ontology repository while the abstract

Supplier agent is an agent on behalf of an enterprise that has capabilities of providing certain services. In a virtual enterprise, there are usually many supplier agents and their coordination relation could be either cooperation (i.e., when they unite for an order) or competence (i.e., when they compete for an order). A supplier agent registers its services at the UDDI registry, receives the order from its web portal, schedules the order tasks using its internal system and processes the contract after the customer makes the decision.

UDDI is a static ontology repository that provides registry and look up services. UDDI can be implemented inside an enterprise and perform like a private UDDI, and it can also be implemented at a shared space outside enterprises for providing public information as public UDDI. Thus, one or more UDDI registries can be employed to provide enterprises with standard terms used in communication languages and knowledge related to these terms' definitions, attributes, relationships and constraints.

Ontology agent provides the semantic integration while an UDDI registry only performs the function as a static ontology repository. In our system, the ontology agent just plays part of its proposed functions in partner search. When a broker agent submits a service query to the ontology agent, it will check the UDDI repository to find the matching supplier agents. And then the ontology agent returns a matching list to the broker agent. An ontology agent may have some kind of learning ability that can gather service information from business transactions and provide the suppliers' references.

Interface agent aims at preventing direct access of internal systems from outside. It performs functions such as transferring messages, receiving tasks, returning order bids, controlling access security, etc. Simply speaking, the interface agent performs functions similar as an application gateway that is a joint point of the inter-enterprise workflow and the intra-enterprise workflow. When the supplier's Web service receives a message, it is transferred to the interface agent at first. The content of SOAP message is deprived, checked and directed to the responsible agent. The interface agent decides which agent is responsible for the message and sends the message to it.

Mediator agent is a coordination agent at the intra-enterprise level that interacts with the workflow engine, reads process policies, and communicates with other resource agents to exchange the information and control process events like error handling (Weiss, 2000). As shown in Figure 11, when a task is received, the mediator agent takes four steps to accomplish the task: (1) task decomposition, (2) coordination cluster, (3) scheduling and (4) task executing.

- Task decomposition is the process of building goals or sub goals for multi-agent system to be accomplished. In this step, a received order is decomposed into a set of tasks and each task may be further decomposed into subtasks by another domain mediator agent in a hierarchical mediator architecture.
- Coordination cluster is the process of creating virtual agent community. After decomposing the task, mediator agent needs to find possible resource agents to finish the task. It is an iterated process in that the mediator agent finds problem solvers by querying the DF agent for the service list until all the tasks have potential agent solutions.

4. Case Study and Implementation

The inter-enterprise workflow is used to define the business process logics that are shared by the participants of a formed virtual enterprise. The business process in a VE starts with an order by a customer for a composite product or service. If we analogue workflow definition as a "class" in programming, a virtual enterprise could be considered as a running instance of such a class which is triggered by customer's requirement, created by its lifecycle, controlled by workflow management, executed by workflow engine and dismantled while goal is finished. As agent based workflow model we propose in previous section, we have designed a prototype system to demonstrate the use of our model.

4.1 Design Principle

Engineering companies of all sizes are turning to the Internet to find parts suppliers, shaving sales and procurement costs in the process. Often, small companies have a hard time in finding manufacturers, and also these manufactures will charge a competitive price to make a small number of parts. A shop may have the latest manufacturing equipment and software, but neither is much good without incoming work orders. While most shops advertise their manufacturing services in directories, such as the yellow pages, or rely on manufacturing representatives and word-of-mouth to get work, others

- Suppliers register/ subscribe to the site and set up an on-line agent/filter.
- The broker agent behind the web portal is created for buyers and assists them to find potential service suppliers. This lets them to adapt to cyclical workloads and fill roles in their capacity.
- To post a quote, buyers enter data such as delivery dates, part amounts, and intent (e.g. switching suppliers, make-versus-buy comparisons, and so forth). They choose a process they want quoted, and indicate the desired supplier attributes such as only local-area suppliers or those with certain certificates.
- The broker agent searches suppliers and sends the order bid.
- After receiving an order, suppliers study the order by computing the costs at their inner system and send messages to the buyer if needed — both parties have each other's company information. The supplier then prepares and privately submits a quote for the job.
- The buyer reviews the submitted quote and performs due diligence via supplier profiles that include ratings. These ratings come from previous transactions where both buyers and sellers rate each other (one to five stars) based on multiple criteria. For instance, suppliers are rated by quality, delivery, and general responsiveness/customer service.
- Contract could be signed between the buyer and the selected supplier online. After order confirmation, the order commitment and the configuration service are invoked to physically add the order to the resource agents.

The web portals and Web services are supported by Java Web Services Development Pack (JWSDP). The Java Web Services Development Pack (Nagappan et al., 2003) provides a convenient all-in-one package for developers who want to start building and deploying standards-based web services immediately. It is a superset of Java XML pack. The main software tools we use for system implementation are lists here:

- Java APIs for Web services such as JAXP, JAXM, JAXR.
- Tomcat as a web portal test environment for Java Servlet or JSP.
- JSP tag library.
- JSSE (Java secure socket extension) for testing web service access via secure connections.
- Ant build tool for platform independent build management,
- Java registry server which is a private UDDI server that can be deployed internally for service publication and discovery.

Agent Communication Channel in a Java environment. Messages are represented using the FIPA Agent Communication Language. AADE also supports FIPA's standard interaction protocols such as Contract Net Protocol.

4.4 Implementation for creating a virtual enterprise

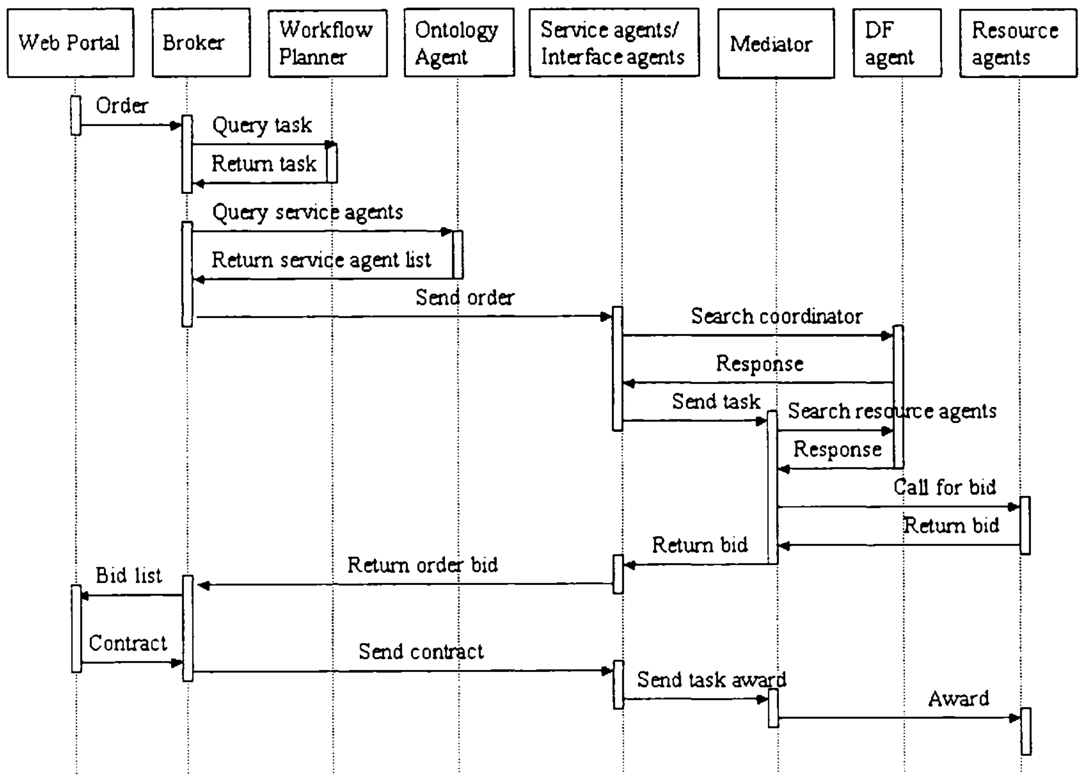

Figure 13. Agent coordination diagram for virtual enterprise formation

In this section, an online order scenario is envisaged to demonstrate the agent based workflow coordination. As a common sense, an online order process could be described as the process of ordering, payment, production, and shipping. For simplicity, we only discuss the first step in this scenario – ordering.

the broker agent needs to query the workflow planner for an order-processing task and the workflow planner agent decides it according to the defined workflow ontology. When the broker agent gets a task indication, it tries to find a service agent to execute the task.

4.4.2 Partners Search and Selection

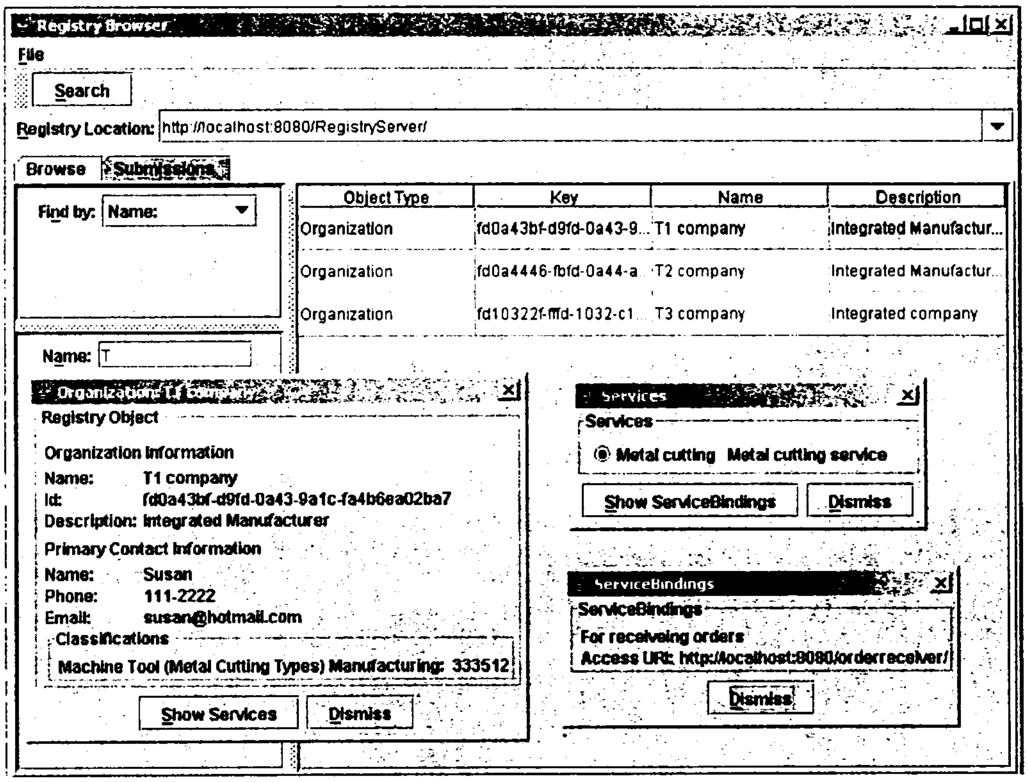

Figure 15.UDDI server diagram

The search and selection of business partners is a very important and critical step in the formation of a virtual enterprise. Partner search can be done based on the following resources: 1) private information resources such as directory facilitator, 2) enterprise's private UDDI repository; and 3) public information resources such as the public UDDI repository. In order to effectively find partners in real time from various resources, we proposed different searching strategies for inter- and intra- enterprise collaboration. At

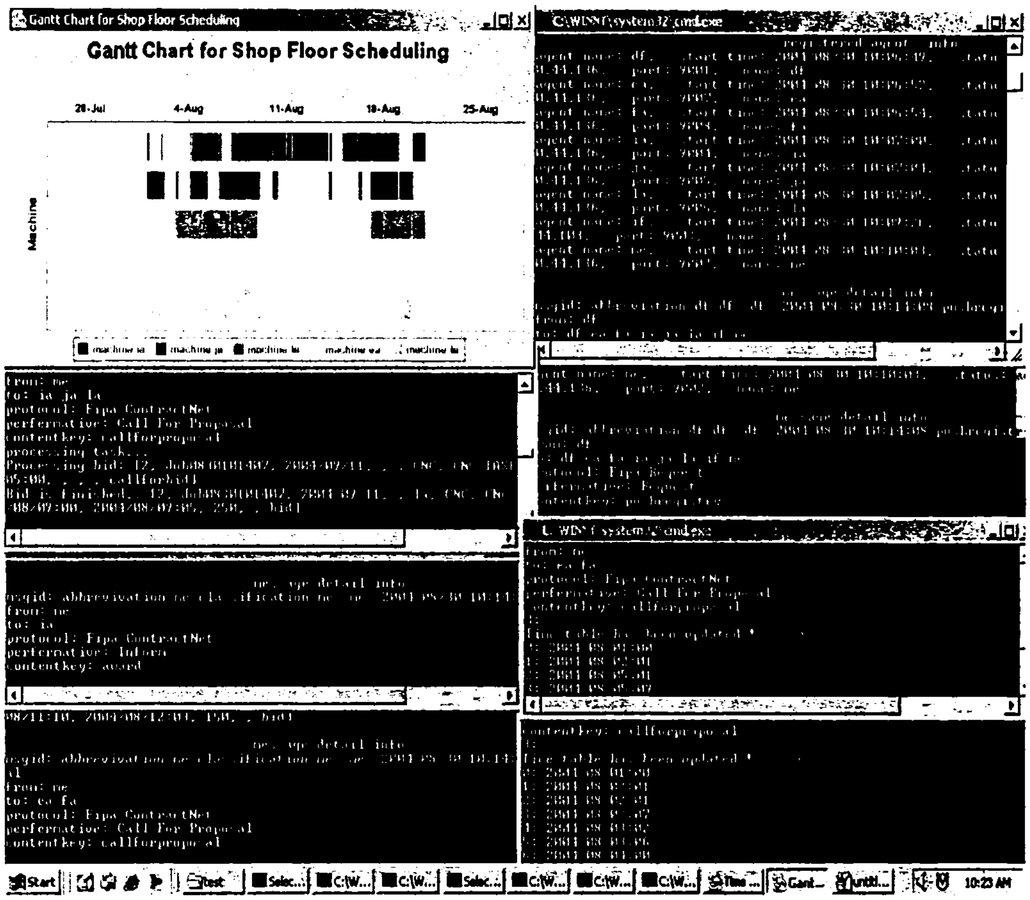

Figure 16. Gantt chart and negotiation example

4.4.4 Contracting

A contracting process is an awarding procedure to designate a service agent to perform a particular task. In our scenario, a business agreement needs to be achieved between customers and service suppliers (external service agents). A contract is generated by the broker agent and mainly based on the customer's order, payment and shipping information as well as the supplier's bid information (Figure 17). The customer keeps the contract and the broker agent sends its copy to the selected supplier for execution.

5. Conclusions

5.1 Summary

Today, workflow management systems often face the problem of runtime process changes and traditional workflow systems are inflexible to capture the dynamic changes. Thus, a flexible and efficient workflow solution requires the ability to efficiently discover and exploit business services in a dynamic and constantly growing environment. It also requires the capacity to dynamically establish relationships among business processes. In this context, we propose the agent based workflow model as a blueprint for orchestrating the business process's dynamics.

The result of this research is toward an approach for distributed workflow process automation. In our agent based workflow model, we propose workflow ontology not only for dynamic runtime interpretation, but also for module reusability at design time. Moreover, the agent based coordination mechanism provides great flexibility for workflow process composition, resource discovery, mediation and interaction. Finally, the automated operational processes can be reused to capture the workflow-based information and automate their processes not only for workflow but also for other enterprise-modeling domains.

5.3 Future Work

The agent based workflow model has some research areas that can be improved through future work.

Negotiation problem. Although agent technology promises the important features like automation, learning, negotiation, etc, some of them are not mature enough yet. For example, there are already plenty studies in negotiation, but the real world is still very complex to be handled by current techniques. Generally, negotiation contains two kinds of problems: negotiation presentation and negotiation strategies. Negotiation presentation builds the formal interaction definition by negotiation protocols and interaction patterns to realize the negotiation process. On the other hand, negotiation strategies are more dependable on situations where conflicts arise between different sets of beliefs. In our case, we use Contract Net based interaction protocol for both inter-enterprise and intra-enterprise bid negotiation. However, we will consider the iterated Contract-Net negotiation process to change the strategies after learning during each turn.

Ontology problem. We are currently developing tools for mapping the upper level workflow process ontology into lower level service ontology that we can use to generate multi-agent instantiations of the workflow. However, there are two problems (or approaches) in current development of ontology.

The first one is whether we should develop a complex ontology system that takes advantages of the full power of first order logic. In fact, although there are already some ontology tools such as OilEd and Racer, it is still very hard to build and maintain large ontologies without the help of domain experts. However, our solution is to build a small set of ontology in a specific problem domain to minimize the difficulty. For this purpose, and implementing agent-based workflow systems. Agent-based approaches primarily emphasize on the agility and reconfiguability of workflow systems. Different from the mathematical approaches targeting at a global optimization through mathematical formulation, the agent-based approaches attempt to achieve optimization through efficient coordination mechanisms.

Regardless of the above problems, we also identify the following research topics as future research opportunities:

- Adaptive workflow process: creation of new business process logic during runtime through learning.
- Integrated with other technologies: integration with other technologies such as project management, intelligent scheduling, etc.
- Managing and Monitoring: integrated business system and their associated Web services raise new requirements for system management. Business transactions that span several systems also need to be monitored as a whole.
- Quality of Service: Quality of service refers to the quality of the availability, accessibility, integrity, performance, reliability, regulatory, and security capabilities of the service.

[8] Camarinha-Matos, L. M., and Cardoso, T., *The PRODNET Architecture, in Infrastructures for Virtual Enterprises - Networking Industrial Enterprises,* L. M. Camarinha-Matos and H. Afsarmanesh, (eds.), Kluwer Academic Publishers, pp.109-126, (1999b).

[9] Chang, J., and Scott, C., *Agent based Workflow: TRP support Enviroment(TSE),* Computer Networks and ISDN Systems, pp. 7-11, 1996.

[10] Choi, H. R., Kim, H. S., and Park, Y. J., *Multi-Agent-based Negotiation Support System for Order based Manufacturers,* In Proc. of ICEC2003, Pittsburgh, PA, Vol.5, pp. 480-488, 2003.

[11] Choi, H. R., Kim, H. S., Park, B. J., Park, Y. J., and Whinston, A. B., *An Agent for Selecting Optimal Order Set in EC Marketplace,* DSS, 36(4), 371-383, 2004.

[12] Mandell, D. J., and McIlraith, S. A., "Adapting BPEL4WS for the Semantic Web: The Bottom-Up Approach to Web Service Interoperation", www.cs.toronto.edu/ ~sheila/iswc2003sam-djm.pdf.

[13] DAML *DARPA Agent Markup Language* http://www.daml.org/.

[14] ebXML website. http://www.ebxml.org/.

[15] Eriksson, H.E., and Penker, M., *Business Modeling with UML: Business Patterns at Work,* John Wiley & Sons, New York, U.S.A, 2000

[16] FIPA, *Foundation for Intelligent Physical Agents,* http://www.fipa.org/.

[17] Hohpe, G., and Woolf, B., *Enterprise Intergration Patterns,* Addison Wesley, Boston, 2004.

[26] McGuire J., Huokka D., Weber J., Tenenbaum J., Gruber T., and Olsen G., *SHADE: Technology for Knowledge-based Collaborative Engineering,* CEAR, 1(3), 1993.

[27] Munindar P.S., and Michael N.H., *Service-Oriented Computing: Semantics, Processes, Agents,* John Wiley & Sons, Ltd, 2005

[28] Nagappan, R., Skoczylas, R., and Sriganesh, R. P., *Developing Java Web Services,* Wiley Publishing, Inc., 2003.

[29] OilEd http://oiled.man.ac.uk/.

[30] Omicini, A., Zambonelli, F., Klush, M., Tolksdof, R. (eds.) *Coordination of Internet Agents: Models, Technologies, and Applications,* Springer-Verlag, 2001.

[31] OWL *Web Ontology Language Overview* http://www.w3c.org/TR/owl-features/.

[32] OWL-S *Web Ontology Language for Web services* http://www.w3.org/~ Submission /OWL-S/.

[33] Parunak V. D., *Manufacturing Experience with the Contract Net,* In M. N. Huhns, (eds.), Distributed Artificial Intelligence, Pitman, pp. 285-310, 1987.

[34] Peng Y., Finin T., Labrou Y., Chu B., Long J., Tolone W. J., and Boughannam A., *A Multi-Agent System for Enterprise Integration,* In Proc. of PAAM'98, London, UK, pp. 155-169, 1998.

[35] Petrie C., Cutkosky M., Webster T., and Conru A., Park H., *Next-Link: An Experiment in Coordination of Distributed Agents,* Position paper for the AID-94 Workshop on Conflict Resolution, Lausanne, Switzerland, 1994.

[36] Sandholm T., Lesser V., *Advantages of a Leveled Commitment Contracting Protocol,* In Proc. of AAAI-96, Portland, OR, pp. 126-133, 1996.

International Conference on Cooperative Information Systems (CoopIS'99), pp. 115-126, Edinburgh, Scotland, September, 1999.

[46] Van Leeuwen, E. H., and Norrie D. H., *Intelligent Manufacturing: Holons and Holarchies,* Manufacturing Engineer, 76(2), 86-88, 1997.

[47] Vidal, J. M., Buhler, P., and Stahl, C., *Multiagent Systems with Workflows,* IEEE Internet Computing, 8(1), 76-82, 2004.

[48] Vieira, T., Casanova, M., and Ferrão, L., *An Ontology-driven Architecture for Flexible Workflow Execution,* In Proc. of the WebMedia & LA-Web 2004 Joint Conference 10th Brazilian Symposium on Multimedia and the Web 2nd Latin American Web Congress, IEEE, 2004.

[49] Haarslev, V., Möller, R., *Description of the RACER System and its Applications,* In Proc. of International Workshop on Description Logics (DL-2001), Stanford, USA, 2001.

[50] Weiss, G., *Multi-Agent Systems,* (eds.), The MIT Press, 2000.

[51] WfMC, *The Workflow Management Coalition specification,* WfMC Standard, Document Number TC-1011, 1994.

[52] WSDL, *Web Services Description Language,* version 1.1, World Wide Web Consortium recommendation, 2001.

[53] Xu, Q., Qiu, R., and Xu, F., *Agent based Workflow Approach to the Design and Development of Cross-Enterprise Information Systems,* Systems, Man and Cybernetics, IEEE International Conference on, Vol 3, pp. 2633-2638, 2003.

Publications

[1] Qi Hao, Weiming Shen, Shuying Wang, "Agent-Oriented Manufacturing Scheduling Services for Enterprise Cooperation", Accepted for SAE 2005 World Congress, (Paper Number: SAE 2005-01-0511), Detroit, MI, April 11-14, 2005.

[2] Shuying Wang, Weiming Shen, Qi Hao, "Agent Based Workflow Ontology for Dynamic Business Process Composition", Accepted for the 9th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2005), (Paper Number: CSCWD2005_674), Coventry, UK, May 24-26, 2005.

[3] Shuying Wang, Weiming Shen, Qi Hao, "Modeling Agent-based Web Services for Inter-Enterprise Workflow Management", Accepted for the 2005 International Mechanical Engineering Congress and Exposition (ASME IMECE 2005), (Paper Number: IMECE2005-79242), Orlando, FL, Nov. 5-11, 2005.