TNC Transactions on Networks and Communications
SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
VOLUME 2, ISSUE 4
ISSN: 2054 - 7420

An Agent-Based Model for Cross-Enterprise Supply Chain Management

Tarini Prasad Panigrahy[1], Manas Ranjan Patra[2]
[1]*Gandhi Institute for Technological Advancement, BPUT, Bhubaneswar, India*
[2]*Department of Computer Science, Berhampur University, Berhampur, India*
tarinip@yahoo.com, mrpatra12@gmail.com

ABSTRACT

In order to cope with the dynamic scenario of fast changing business requirements enterprises have embraced web technologies to manage their business processes. However, the ability to integrate business processes like procurement, customer relationship management, finance, human resources and manufacturing in a typical supply chain on the web is a challenging task. Today's virtual enterprises need to integrate different workflows within and across enterprises efficiently so as to provide seamless services. Cross-enterprise workflows can not only streamline and coordinate business processes across organizational boundaries in a dynamic Web environment but can provide low cost and flexible solution to supply chain management. In this paper, we have proposed an agent-based cross-enterprise workflow Management System (WFMS) architecture which can dynamically integrate the workflows and compose a workflow execution community customized to different workflow specifications. The model allows the process agents to update the execution plans dynamically and coordinate the functions of the participating service agents.

Keywords: Supply Chain Management, Workflow Management System, Cross Organization Workflow, Agent based workflow.

1 Introduction

Supply Chain Management (SCM) is referred to as the logistics network, which synchronizes a series of inter-related business processes in order to: (1) acquire raw materials from a supplier, (2) transform these raw materials into finished products(manufacturing), (3) add value to these products, (4) distribute and promote these products to either retailers or customers, (5) make information exchanges among various business entities (e.g. suppliers, manufacturers, distributors, retailers and customers). There are mainly three stakeholders to play their roles in a typical supply chain, namely, the Manufacturer, who produces/provides the products, the

DOI: 10.14738/tnc.24.380
Publication Date: 24th August 2014
URL: http://dx.doi.org/10.14738/tnc.24.380

Customer, who purchases the products, the Supplier, who provides raw materials to the manufacturer based on their demand.

Figure 1 show an inter-organizational co-operation system where each of the stakeholders has its own SCM. So SCMS-supplier, SCMS-Manufacturing and SCMS-Customer should perform all supply chain activities in a coordinated fashion. Here the cooperation process model tries to ensure business interoperability with other enterprises. The logistics supply chain coordination includes both vertical and horizontal logistics supply chain coordination and risk management processes. The SCMS-Manufacturing has the goal to optimize their production planning and resource utilization, SCMS-Supplier have the goal to balance supply and demand, negotiate with the supply and demand process and minimize the inventory and number of stock-outs for the whole logistics supply chain, SCMS- Customer has to provide the demand through an order and receives the stock from the manufacturer by maintaining its own inventory system.

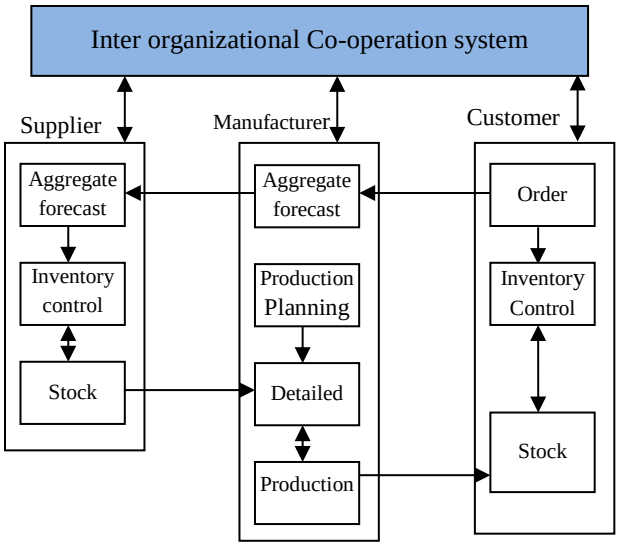

Figure 1: Inter organizational Co-operation system

The main aim of the Supply Chain Management (SCM) is to enhance the operational efficiency, profitability and competitive position of an organization and its supply chain partners. The performance of a SCM is measured in terms of the objectives such as superior quality, cost minimization and delivery on-time etc. All the entities of a logistics supply chain are highly interdependent. As a result, performance of any entity in the supply chain depends on the performance of others, and their ability to coordinate activities within the supply chain [1][2][3][4]. The Supply Chain is a network of several businesses and their relationship [5]. In a typical SCM the independently managed companies coordinate their activities to form a Virtual Enterprise [6]. For an example, the order placement process, order fulfilment process and shipment process in a typical SCM might be done by different companies. These companies provide their offerings as independent functional units called as web services. According to Erl

[7][8] these web services (Sub process) combine together (as shown in Fig 2) to make up a larger part of the business logic automation and each one of them can be distributed.

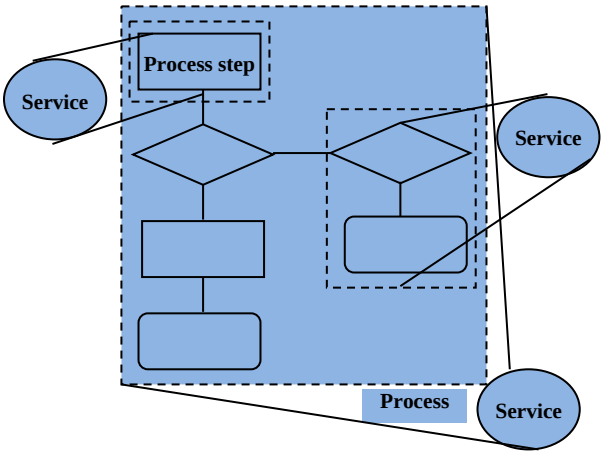

Figure : 2 Communications between Process and Service

Such an environment can be called as a service oriented cross organization environment which incorporates the interoperability of web services from various organizations. For example in a SCM the stakeholders supplier, customer and manufacturer can be considered as independent components, which can be specified with web services technologies, have the capability of being discovered and accessed from distributed locations. We specify these web services through Web Services Description Language (WSDL) [9] and can invoke them using the Simple Object Access Protocol (SOAP). In addition to advertising the specifications of distributed services universally, we use Universal Description Discovery and Integration (UDDI)[9] architectures.

Further, dynamics and unpredictability of the business such as faults or breakdown in utility equipment or an extended delay in taking delivery of raw materials, failure of production facilities, customers change or cancel orders, etc. make real-time cooperative operations on the supply chain complex and difficult [11]. For example, whenever there is a change in a customer order, then the partners in supply chain should be immediately communicated and react to the changes accordingly. Thus the occurrence of an event should be immediately propagated throughout the supply chain so as to do timely coordination and interaction of business processes (according to the changes) within and between the enterprises. So in designing a SCM, the key principle is timely coordination and interaction between various business processes in order to meet the challenge [12]. Workflow management systems have been widely adopted in providing solutions to facilitate SCM implementation in an organization. However, because of the lack of flexible mechanisms with cross-organizational business activities workflow management technology has had little success in achieving dynamic coordination and interaction in a supply chain management. Software agents are autonomous and goal-oriented software entities, which with other agents can operate asynchronously and

co-ordinate when needed [13]. Further in a cross organizational collaboration, the web services have the following shortcomings which we can resolve by using Software agents:

i) When we examine a web service, it is a self-describing software process/component for an application and does not have enough knowledge about its environment, users, software components, and outside world. On the other hand, software agents are capable of reasoning, and interacting with other entities.

ii) When we think of a web services it is discoverable by XML-based UDDI standard. Current standard of UDDI is only able to recognize terms "syntactically". But the main dispute in service discovery is how to find services, which are "semantically" the same as clients' needs. But when we use Software agents, they can operate at the knowledge level, at which they are able to reason semantically on the service requesters.

In this paper we take advantage of software agent technology under the control of a workflow management system, to effectively integrate cross-organization workflows. The difficulty with current workflow technology is its inability to cope with pro-activeness, dynamic interactions and component autonomy which agent-based system can provide.

Rest of the paper is organized as follows. In section 2, we formally specify the workflow model. In section 3 we present an agent-based workflow architecture that we use to describe workflow process management system. In section 4, we present the execution of the SCW for the proposed architecture, in section 5, we present our current implementation. A scenario of using agent based workflow is reported in section 6, finally we discuss some related works and summarize our findings in section 7.

2 Workflow Specification Model

We propose a service oriented workflow model. Using this model, major parts of a workflow process can be represented as web services (i.e., tasks or workflows).Thus, to represent a cross-organizational workflow process we use higher-level specification that can be composed from existing component workflows. Workflow components can be reused and specialized in different organizational settings.

We represent a workflow process as a tuple W:

$W = (S, E_w, R_w)$ where:

1. S is a set of services: $S = \{s_1, s_2 \ldots s_n\}$

2. E_w, is a set of workflow Event-Condition-Action-Monitor (ECAM) rules:

$E_w = \{e_1, e_2, \ldots e_n\}$. The workflow ECAM rule specifies the coordination among the tasks. We have a monitor agent to monitor the execution of the workflow and to inform process agent and other monitor agents if there is any exceptional event during the execution of the workflow. In order to

Define the ECAM rules, we use the following operations:

- ■Δ s: Enables the execution of the service s;
- ■∇ s: Disables the execution of the service s;
- ■ ↪s: Sends a message to the service s;
- ■ ∗W : Starts the execution of the workflow W;
- ■ ⊙W: Finishes the execution of the workflow W.

A workflow ECAM's rule is defined as follows:

(s_i . result == R)→ Δs_i

Which means enable the execution of service s_j when execution the service s_i returns R.

3. R_w, is the result of the workflow execution. It can be success, failure or null .Before the workflow execution begins it is initialized to null. A service can be represented as a tuple s_i:

$s_i = (N_i , O_i , E_{si} , D_i , R_{si})$ where:

1. N_i Represent the name of the service, which is a string to identify the service.

2. O_i is the task object (type of service), which is a tuple (P, T, TR, OP) where:

(a) P Represent set of properties which reflect the general information

about the task object (e.g., the creator of the task).

(b) T Represent a set of possible states: $T = \{t_1, t_2, ... t_n\}$, where:

i) t_i is the initial state: $t_i \in T$

ii)t_f is the final state: $t_f \in T$, if object's State changes to t_f, it implies that the execution of the task is successful.

(c) TR Represent a possible set of transitions (t_i , t_j) , where t_i , $t_j \in T$

(d) OP Represent a set of operations on T, such that TR ⊆ T X O P X T

3. E_{si} is a set of ECAM rules: $E_{si} = \{ e_{i1} , e_{i2}, ... e_{in}\}$. The ECAM rules for a service are used to guide the execution procedure of individual services, for example when to trigger related operations. Further it should be noted that a service can be atomic or complex (i.e a sub process in the workflow). Two kinds of rules are supplied:

C → A: if condition C is true, then operation A might be executed.

C ⇒ A : if condition C is true, then operation A must be executed.

4. Di Represent the deadline to complete a service/task.

5. R_{si} Represent the result of the task execution which may be success, failure, or null. In the beginning we set R_{si} to null. In this model, we distinguish workflow ECAM rules from task/service ECAM rules. Workflows ECAM rules are used to specify the interaction among

workflow services, where as the ECAM rule for a service are used to guide the execution procedure of individual services. In our model, the definition of a cross-enterprise workflow involves the identification of the services/tasks that compose the workflow and specification of the interactions between them. This is different from traditional workflow where it is also necessary to define who will be responsible for the execution of each task and how much time is allocated for each task at specification time. In our approach, tasks are dynamically assigned to service providers (i.e. entities that can perform the tasks, e.g., programs) during the enactment of a cross-enterprise workflow. The dynamic composition of services to provide a cross-enterprise workflow will be discussed further in section 4.

3 Agent based Workflow architecture

Figure 3 describes architecture for an agent based workflow. The entire architecture can be classified into three logical components. They are: workflow definition tool, agent community and actual service.

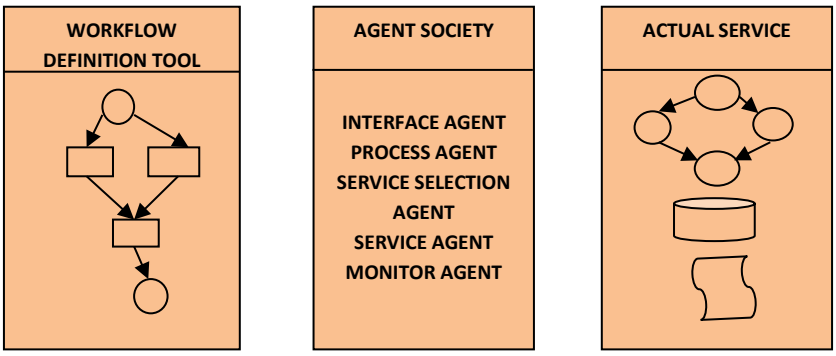

Figure :3 Agent based Workflow Management System

The task of the Workflow Definition Tool is to define cross-enterprise workflow specifications through the end user by using a standard GUI tool. The Agent Community is a set of agents that forms an agent Society which acts cooperatively to provide the general functionalities of a cross-enterprise workflow execution engine. Actual Services offer applications (called as services) from the physical organizations that allow the user to access and use them to perform the specified tasks during workflow execution.

The detail architecture of the agent based workflow is presented in Figure 4. It supports cross enterprise workflow that involves multiple organizations. For executing such a workflow we use five types of agents in the system, namely interface agent, service agent, service selection agent, process agent and monitor agent. For each workflow instance these agents form an agent community to execute the workflow.

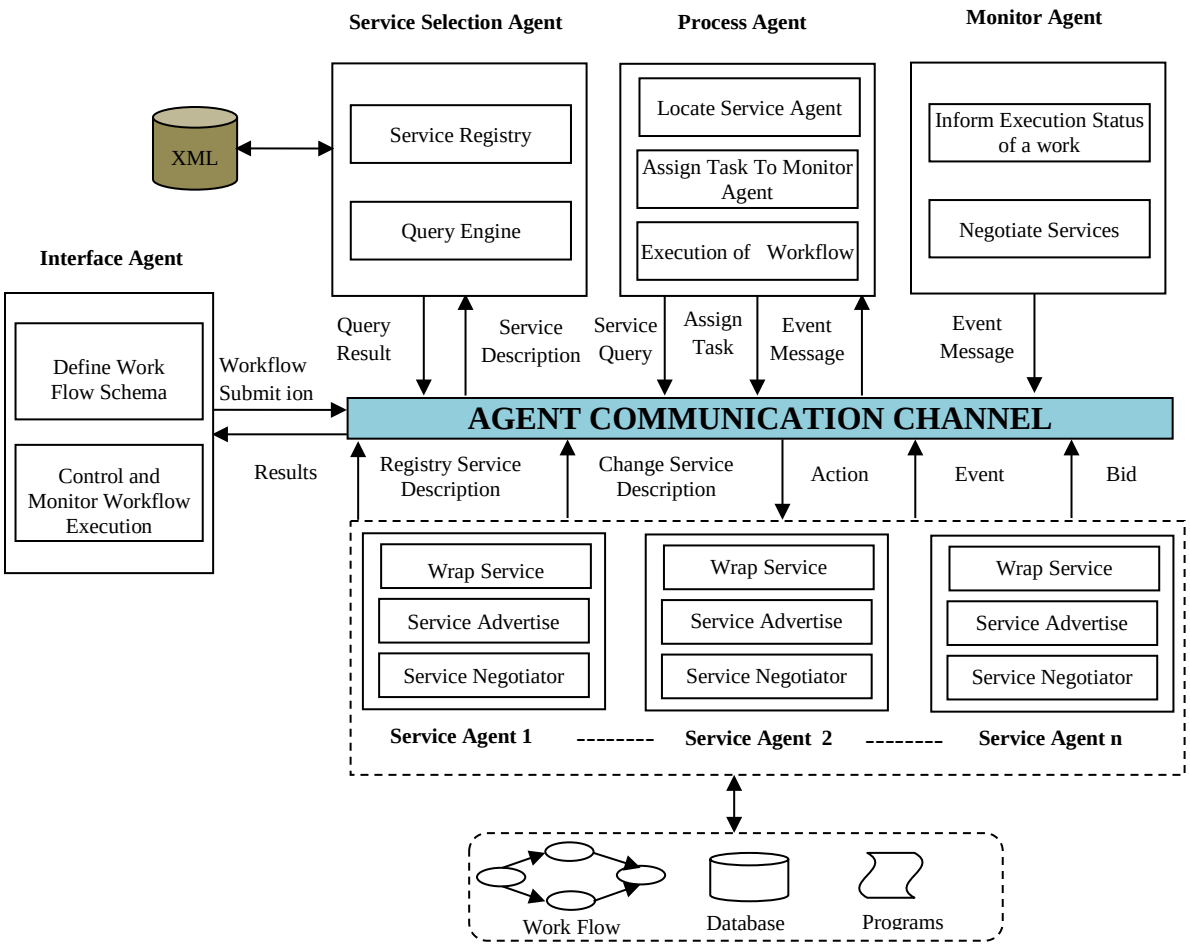

Figure 4: Architecture of agent Based Workflow

3.1 Interface Agent

It is through this agent the user can interact with the system. Here the user plays two types of roles; (i) as workflow process composer to define workflow schema (ii) as end user to create and start workflow instances, control and monitor the execution of workflow instances. The interface agent provides a tool to define the workflow schema, which is represented by UML state chart diagram. In order to draw the workflow schema we follow the following methodology:

1. Identify various tasks that constitute the workflow where each task in a state chart diagram constitutes input data, output data, states and transitions.
2. Draw the workflow by specifying the control flow among the tasks using transitions, fork and Join

Once the process composer draws the UML state chart diagram for the workflow schema, the end user generates an XML document for it. Further the end user provides the parameters required to select the appropriate service agents to execute the workflow. The XML document is submitted to the process agent to execute the workflow.

3.2 Service Agent

Service agents are used to abstract the business processes from their physical organisations. The agents capture different states of a business process which can be expressed in XML as follows:

```
<?xml version=”1.0” encoding=”UTF-8”?>
<!DOCUMENT SUPPLY CHAIN MANAGEMENT “Customer Order”>
<CustomerOrder>
<Service ID=”001” Name=”customer order process”/>
<Services>
<States>SendOrder1, ReceiveItem1,ReleasePayment</states>
<Transitions>
<Transition>
<From>SendOrder1</From><To>ReceiveItem1</To>
<TransitionString>
<Event>ReceiveItem1</Event>
<Condition>Avail(Item1)==true</Condition>
<Action>ReceiveItem1</Action>
</TransitionString>
</Transition>
<Transition>
<From>ReceiveItem1</From><To>ReleasePayment</To>
<TransitionString>
<Event>ReleasePayment</Event>
<Condition>avail Credit >= PriceQuoted</Condition>
<Action>ReleasePayment</Action>
</TransitionString>
</Transition>
</Transitions>
<Services>
</CustomerOrder>
```

(Customer_orderProcess.xml)

A service agent contains the following information:

(i) Service identification, which represents the port that contains the service, the format of the request message that it can understand and process

(ii) Agent capabilities which specify the name and the operations that the underlying service provides, and

(iii) Agent properties which specify constraints associated with the service

For example, the Customer Order Processing in a SCM can be represented in a service agent as shown below:

Agent Identity

Agent name: OrderItem1Agent

Agent address: 203.4.5.101.2323

Agent type: service

Agent interface: XML, SQL92

Agent Capabilities

Supported object: OrderItem1

Supported operations: SendOrder1, ReceiveOrder1,ReleasePayment

Supported query: price

Agent properties

Agent constraint: Delivery period {within 30days}

(Service Agent)

3.3 Service Selection Agent

The service space in a web-based environment is large and highly dynamic. In order to search a specific service in such an environment efficiently we use a service selection Agent. The service agents advertise and hence register their offerings like identity, capabilities and constraints in a meta data repository (UDDI). The meta data is used to locate the service agents. The process agent who is responsible for transforming the workflow specification to a workflow instance makes a query to the service selection Agent to find the suitable service agents. Then the service selection Agent finds one or more suitable service agents from the meta data repository for a given task as per the workflow specification. Service agents construct the advertisement message in XML. Such an advertisement message is as follows:

```
<?xml version=”1.0” encoding=”UTF-8”?>
<Message>
<Message Type>Advertise</Message>
<Repository> yellowpages <./Repository>
<AgentIdentity>
<AgentName>OrderItem1Agent</AgentName>
<AgentAddress>203.4.5.101.2323</AgentAddress>
<AgentType>service</AgentType>
```

```
</AgentIdentity>
<AgentCapability>
<SupportDTD>
http://awascm.gita.cse.edu.in/supply_chain.dtd
</SupportDTD>
<SupportService>
http://SupplyChain.gita.cse.edu.in/OrderItem1.xml
</supportServices>
</AgentCapability>
</Message>
```

(Advertisement Message.xml)

3.4 Process Agent

The main task of the process agent is to transform a workflow specification into a workflow instance. It gets the workflow specification from the interface agent and integrates the service agents to execute the workflow. The various responsibilities it is assigned with are

1. The process agent for a particular task in the workflow queries to the service selection Agent and finds the relevant available service agents. Then it negotiates with the service agents about task execution. When the process agent receives many choices to perform a particular task, it allows the user to do selection among the available service agents. Finally, it makes connections to that service agent and assigns the task specifications to it.
2. Once a task is assigned to a service agent, a monitor agent is then created and sent to the task execution site. The process agent instantiates the task ECAM rules only after the monitor agents are migrated to the task execution sites .The process agent do the workflow execution according to the workflow's ECAM rules. It coordinates the tasks to achieve a certain goal and also enable, disable or suspend the tasks according to the workflow ECAM rules.
3. When they do not have the ability to process a task, it forwards the workflow specifications to another process agent. For example, a process agent specialized in workflow implementation may not be able to process a supply-chain workflow.
4. The process agent during the workflow execution receives all the event messages from the monitor agents. This makes the user to know the status of the workflow instance without having to subscribe to any of the service agents

3.5 Monitor Agent

A monitor agent monitors the actual execution of a given task at the site of the service provider. Its functionalities are:

1. During process instantiation the monitor agent supervise the task execution at the corresponding service provider's site where the task is actually executed.
2. It downloads the ECA rules of the corresponding task from the process agent which will guide the service agent in executing the task.
3. It updates the execution plan based on the changes, for example when a service agent fails while executing a task. In such a case the monitor agent can be called back and updated by the process agent and then can re-migrate to continue its monitoring function at the local site of the service provider.
4. When a service agent finished certain action during the execution of a task, it informs the monitor agent regarding its action. Then the monitor agent informs the process agent and other monitor agents about the execution status of the task. The process agent then forwards such messages to the interface agent so as to make the user aware of the progress of workflow execution.

4 EXECUTION OF Cross-organization workflows

In order to execute a workflow, the above five agents form a community according to the particular workflow specification, also the community decides the workload of individual agents as well as the quality of the service they should provide. Once the execution of the workflow completes the agent community disbands.

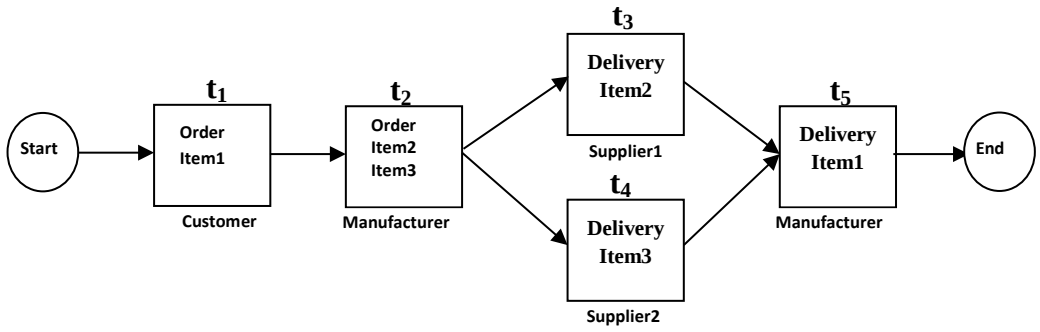

Figure 5: SCM planning Workflow

4.1 An Example

Consider the workflow of a typical Supply Chain Management (SCM) as shown in Figure 5. It involves interaction among agents from a customer company, Manufacturer Company, and supplier company. The business workflow has to execute the following tasks:

OrderItem1 task: where a customer, order for an item called as Item1 to a manufacturer.

OrderItem2Item3 task: where the manufacturer orders for Item2 and Item3 to Supplier1 and supplier2 respectively, in order to Produce Item1.

DeliverItem2 task: where supplier1 supplies the raw material, Item2 to the manufacturer.

DeliverItem3 task: where supplier2 supplies the raw material, Item3 to the manufacturer.

ProduceItem1 task: where the manufacturer produce Item1 and supplies to the customer.

We can define the workflow W for the SCM process as follows:

Workflow W = {T, E_w, R_w}, where:

T= {t_1, t_2, t_3, t_4, t_5}

E_w = {e_{w1} ,e_{w2}, e_{w3}}, where

e_{w1} :*W$\Longrightarrow$ $\Delta t1 \wedge \Delta t2$

e_{w2} : (t_1.result == success) $\wedge$ (t_2.result == success)$\Longrightarrow$ $\Delta t_3 \wedge \Delta t_4$

e_{w3} : (t_3.result==success)$\wedge$ (t_4.result==success)$\Longrightarrow$ Δt_5

Then the tasks:

t_1= {N_1, O_1, E_{t1}, D_1, R_{t1}} ,where

N_1 is customer Order Process

O_1 is OrderItem1 defined as:

States :{callForBid, Negotiate, Assign, ReceiveDelivery, ReleasePayment}

Operations

Negotiation :{CallForBid, Negotiate}

Assignment :{ Negotiate, Assign}

E_{t1} ={e_{11},e_{12},e_{13}},where

e_{11} : (t_1.Item1.state==CallForBid)$\Longrightarrow$ ($\hookrightarrow$ t_2)

e_{12} : (t_2.Item1.state==Bid) $\Longrightarrow$$t_1$.Item1.Negotiate

e_{13} : (t_1.Item1.state==Assign) $\Longrightarrow$$t_2$.Item1.ReceiveOrder1

D_1=20 Sept. 2013

R_{t1}=Null

t_2={N_2,O_2,E_{t2},D_2, R_{t2}} ,where

N_2 is ReceiveItem2Item3 Process

O_2 is Manufacturer OrderItem Process, defined as:

States :{ Bid, ReceiveOrder1,OrderForItem, ReceiveItem,ReleasePayment}

Operations

Receive-Order :{Bid, ReceiveOrder1}

Receive-Item :{OrderForItem, ReceiveItem}

Payment :{ ReceiveItem, ReleasePayment}

E_{t2} ={e_{21},e_{22},e_{23}},where

e_{21} : (t_2.Item1.state==Bid)⟹ (↪ t_1)

e_{22} : (t_2.Item1.state==ReceiveOrder1)⟹t_2.Item3.OrderForItem⋀ t_2.Item4.OrderForItem

e_{23} : (t_2.Item2Item3.state==ReceiveItem) ⟹t_2.ReleasePayment

D_2=20 Sept 2013

R_{t2}=Null

t_3= {N_3, O_3, E_{t3},D_3, R_{t3}} ,where

N_3 is Supplier1 Process

O_3 is DeliveryItem2 Process, defined as :

States :{ReceiveOrder, DeliveryItem, ReceivePayment}

Operations

Delivery-Item :{ReceiveOrder, Deliveryitem}

Send-Invoice :{DeliveryItem, SendInvoice}

ReceivePayment :{SendInvoice, ReceivePayment}

E_{t3} ={e_{31},e_{32},e_{33}},where

e_{31} : (t_2.Item2.state==ReceiveOrder)⟹ t_3.Item2.DeliveryItem

e_{32} : (t_3.Item2.state==SendInvoice)⟹(↪ t_5)

e_{33} : (t_5.Item2.state==ReleasePayment) ⟹t_3.Item2.ReceivePayment

D_3=21 Sept. 2013

R_{t3}=Null

t_4= {N_4, O_4, E_{t4}, D_4, R_{t4}} ,where

N_4 is Supplier2 Process

O_4 is DeliveryItem3 Process, defined as:

States :{ReceiveOrder, DeliveryItem, SendInvoice, ReceivePayment}

Operations

Delivery-Item :{ReceiveOrder, Deliveryitem}

Send-Invoice :{DeliveryItem, SendInvoice}

ReceivePayment :{ SendInvoice, ReceivePayment}

E_{t4} ={e_{41},e_{42},e_{43}},where

e_{41} : (t_4.Item3.state==ReceiveOrder)⟹ t_4.Item3.DeliveryItem

e_{42} : (t_4.Item3.state==SendInvoice)⟹(↪ t_5)

e_{43} : (t_5.Item3.state==ReleasePayment) ⟹t_4.Item3.ReceivePayment

D4=21 Sept. 2013

R_{t4}=Null

t_5= {N_5, O_5, E_{t5}, D_5, R_{t5}} ,where

N_5 is Manufacturer DeliveryItem Process

O_5 is DeliveryItem1 Process, defined as:

States :{ReceiveItem2, ReceiveItem3,Produce,Delivery,ReceivePayment}

Operations

Produce-Item1 :{ReceiveItem2, ReceiveItem3,Produce }

Send-Invoice :{DeliveryItem,SendInvoice}

ReceivePayment :{SendInvoice,ReceivePayment}

E_{t5} ={e_{51},e_{52},e_{53}},where

$e_{51:}$(t_5.ReceiveItem2.state==success)∧(t_5.ReceiveItem3.state==success)⟹t_5.Item1.Produce

e_{52} : (t_5.Item1.state==Delivery)⟹ t_1.Item1.ReceiveDelivery

e_{53} : (t_1.Item1.state==ReleasePayment) ⟹t_5.Item1.ReceivePayment

D_5=22 Sept. 2013

R_{t5}=Null

4.2 Composition Procedure

The workflow integrator is used to specify the supply chain workflow W. An instance of the workflow is executed as described below.

4.2.1 Parsing Workflow Specifications and Searching for Service Agents

The process Agent parses workflow specifications W that consists of five tasks: t_1, t_2, t_3, t_4, and t_5.The main responsibility of the process agent is to assign the different tasks t_1, t_2, t_3, t_4, t_5, to the available service agents on the Web. The process agent queries the service selection Agent for the appropriate service agents which can carry out these tasks t_1, t_2, t_3, t_4, t_5. The

content of the query message that the process agent sends to the service selection Agent for a task t_1, is as given below:

Message Type : query
Agent Identity : ?
Group Identity : ?
Task Object : t1.customerOrder
Search : all
HOP : 3

(Query Message)

When the service selection Agent receives such a message it Searches everywhere in its yellow pages and catalogue repositories, as the query indicates to search in all repositories of the service selection Agent. Here the hop count is set to 3, which indicates that the request will be propagated to at least three service selection Agents. The result is then returns to the process agent as shown below.

Message Type: reply
Yellow page: (ip=172.16.1.7,port=2034, agent name=SupplyItem1)
Catalogue: (ip=230.15.1.7, port=2221, group name=SupplyItem)

(Result Message)

The above message indicates that the process agent discovers one service agent and one group service agents which can execute the task t_1.

4.2.2 Assigning Tasks

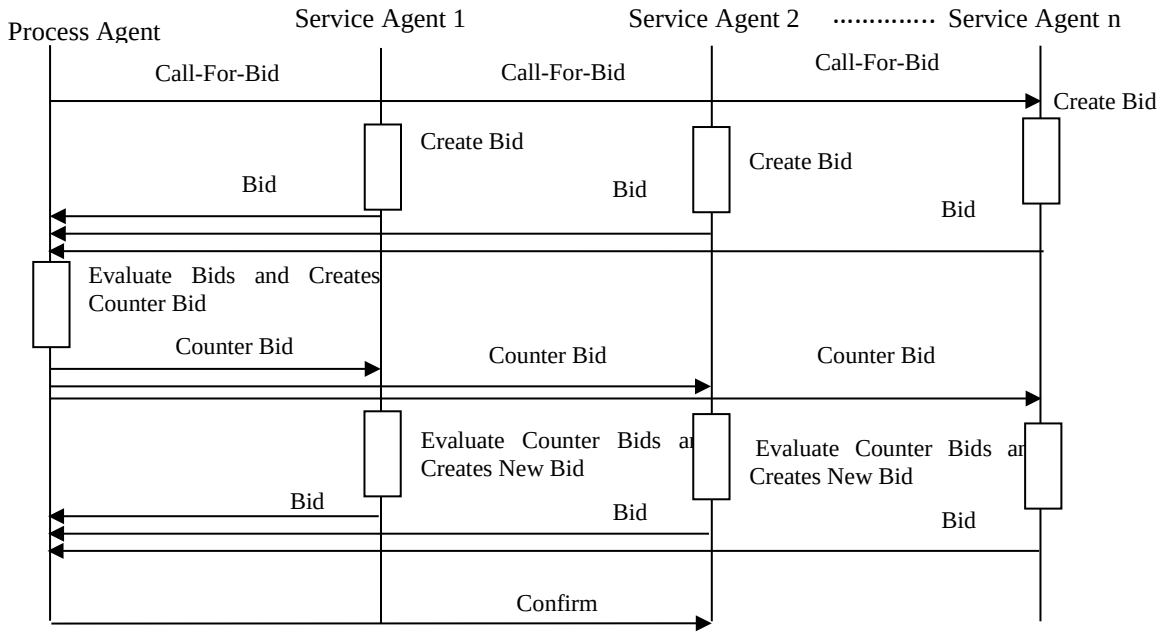

Figure 6: Negotiation Protocol

A task can be assigned to the service agent in two phases.

- Negotiation Phase:

The negotiation protocol is as shown in the above figure 6. In this phase, the process agent sends a Call-for-Bid message as shown below to all the service agents.

Message Type: Call-For-Bid
Sender: 203.5.15.21: 1234: Process-agent
Task Object: t1.customerOrder
Price: ?

(Call-For-Bid Message)

When the service agents receives the Call-For-Bid message from the process agent then depending upon the task specification, it decides whether to respond or not with a Bid in XML. Example of such a Bid is as given below:

```
<?xml1 version=”1.0” encoding =”UTF-8”?>
<!DOCTYPE Bid SYSTEM “bid.dtd”>
        <Bid>
         <TaskBid>
      <Task>Supply_Item1</Task>
      <Cost>200000 Rupees </cost>
     <WorkDuration>2-Days</WorkDuration>
          </TaskBid>
<AcceptDeadline>20/10/2013</AcceptDeadline>
</Bid>
```

(XML Respond Message)

- Task Assignment Phase:

After the process agent receives the bids, it sends a counter bid to the service agents. In the counter-bid the service agent bargains the execution cost and work duration of a task. Upon receiving the counter-bid from all the service agents, the process agent decides and assign to the best service agent based on minimal execution cost and task execution duration. Then the process agent sends task assignment message as given below to the appropriate service agent. After the service agent receives the assignment message, it sends a confirmation message to the process agent.

Message Type : assignment
Sender : 203.5.15.21: 1234:Process-agent

Task : tl

Receiver : 203.5.1.2: 1334:Supplier-agent

(Task Assignment Message)

4.2.3 Monitor Agents and their activities

The process agent creates Monitor agents that must be deputed to the site of service agents. The monitor Agent gets migrated to the site of the service agent where the actual task is executed. Once the monitor agent arrives at the service agent site, it will send a confirmation message to the process agent to indicate that it is ready to monitor the tasks.

When the user defines the ECAM rules, they (ECAM rules) do not have any execution context information and cannot be executed. Once the monitor agents have migrated to the task's execution site, the rules can be instantiated by the process agent. For instance, assuming that the tasks t_1,t_2,t_3,t_4,t_5 have been assigned to service agents *OrderItem1-agent, OrderItem2-agent, DeliverItem2-agent, DeliverItem3-agent, ProduceItem1-agent* and *monitor agents* MA1, MA2, MA3, MA4, MA5 have migrated to tasks' execution sites respectively, then the following ECAM rule is instantiated.

e_{12} :(t_2.Item1.state==Bid) $\Longrightarrow t_1$.Item1.Negotiate

can be instantiated to

(t_2(MA2).Item1.state==Bid) $\Longrightarrow t_1$(MA1).Item1.Negotiate

The monitor agent downloads the ECAM for the instantiated task. Once all the tasks have been assigned to the service agents, all the monitor agents have been created and migrated to the tasks' execution sites and have downloaded the instantiated task ECAM rules then the workflow agent society is ready to execute the cross-enterprise workflow instance. The composed agent society is not a static entity. It might dynamically change during workflow execution. Therefore if the assigned service agent fails to execute the task then the process agent can locate an alternate service agent and execute the workflow.

4.3 Distributed Enactment of a Cross-Organization Workflow

The workflow engine acts as a central task coordinator. Based on the workflow specification, it creates process instance and subsequently the list of different tasks and also controls the execution of the tasks and coordinates task execution as well. The disadvantage with such a central control model is that when there is an exception occurs during the coordination and the task execution it results in a sole point of failure; when an anomaly occurs at the runtime, central server needs to suspend the whole workflow instance to handle it.

We put forward a distributed coordination approach which adopts the dynamic agent technology. We segregate the task control flow and the task coordination: the process agent are responsible for the task execution flow control while the monitor agents do the task coordination for the task those are to be distributed in all the task execution sites. Here to

demonstrate, the working principle of distributed coordination approach we consider a Simple example. Here, we assume workflow agent-community has been formed for the workflow specification W, five tasks have been assigned to three types of service agents (Customer-agent, Supplier-agent, Manufacturer-agent). Five monitor agents MA1,MA2,MA3,MA4,MA5 have been created and deputed to each task's the execution sites , all the tasks' ECAM rules have been instantiated and downloaded by the monitor agents.

Starting the Workflow:

At the start, the process agent will execute the first ECAM rule that is:

e_{w1} : *W$\Longrightarrow \Delta t_1(MA1) \wedge \Delta t_2(MA2)$

The action Δt_1(MA1) and Δt_2(MA2) results in sending messages to monitor agent MA1 and MA2 which informs the service agents, viz., Customer-agent and Manufacturer-agent to start executing their tasks.

Executing the Task t_1 to t_5 :

Once the customer orders for an item(Item1) then the Customer-agent begin to execute its task and also send the enable signal t_2(MA2) message(from process agent) to all the Manufacturer-agent. When the task is assigned to a manufacturing agent , then the manufacturer, order for Item2 and Item3 to supplier1 and supplier2 represented with the task $t_3$and t_4.The monitor agent t_3(MA) and t4(MA) will inform the service agents supplier1 and supplier2 to execute their tasks. After the manufacturer agent gets the results from the supply-Agents, supplier1 (t_3) and supplier2 (t_4) then, the manufacturer-agent will finish the workflow with the monitor agent t_5 (MA) and pass the result to the customer. The monitor agent periodically pings the service agents just to check whether any of them fails or not. If any of the service agent has exhausted the estimated execution time and yet not completed the task, in such case the monitor agent pings it more frequently, because such service agents are more likely to fail. Service agents also sends message to the process agent, when they have completed certain activities during their execution of a task. Then the process agent forwards such message to the interface agent, so as to make the user to know the progress of workflow execution. The process agents also keeps log of execution results of all the tasks which may require selecting the execution plan in future. Once the workflow has been finished then the agent society will dismiss the workflow and a new agent community will be formed to build a new workflow instance.

5 Prototype Implementation

A prototype implementation of the agent based workflow as depicted in figure 4 has been implemented. The prototype is deployed by using Enterprise Java Beans (EJB). Agent communication channel is implemented using Java Shared Data Toolkit (JSDT). The Service agents and Interface agent are implemented using Java.

In the service agent, the service wrapper provides an interface to the actual business process of an organization. Service description is an XML document that describes the service provided by the business process. Service advertiser registers the services into the service selection Agents. Service Negotiator negotiates with the integration agent about service execution.

In the service selection Agent service registry and Query Engine are implemented as Entity Beans. We use Oracle 8i Database as meta-data repository to store information about the service (Such as content, type, location etc).Each XML document is stored in ORDBMS. The Query Engine takes a query from the process agent and translates it into SQL query. Here we use Oracle XML SQL Utility for java to pass an SQL-Query to the underlying Oracle8i Database. Then the results are sent to the utility which then embed it with XML .This result is a set of service agents descriptions. In the process agent the Locate-service is implemented as entity bean and workflow-Execution is implemented as Session Beans. In the Monitor agent execution-status-of –workflow and Service negotiator is implemented as Session Beans.

6 Scenario

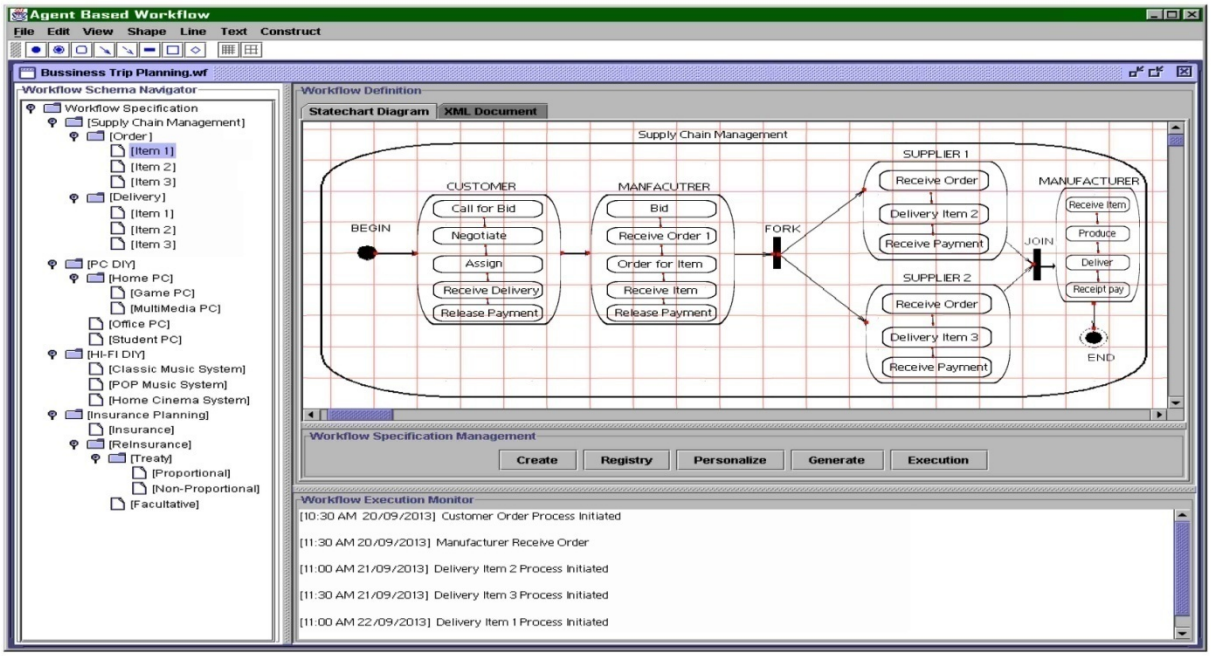

Figure 7: Scenario of Supply Chain Management

We consider the SCM-workflow to explain the use of agent based workflow. The workflow schema consists of five tasks which are OrderItem1 (t1), OrderItem2Item3 (t2), DeliveryItem2 (t3), DeliveryItem3 (t4), DeliveryItem1 (t5).The workflow specification can be explained by taking the business process of M/s HP Ltd, to supply HOME PC,OFFICE PC and STUDENT PC to one of its valued customer. In order to fulfil the order it needs the raw materials from M/s

Fablin Retailer and M/s Radiant Suppliers. The various tasks carried out during the entire supply chain can be explained as below:

The Order PC DIY task where the customer orders for different types of PC to M/s HP Ltd.

The OrderRaw Materials task where M/s HP Ltd orders the raw materials to M/s Fablin Retailers and M/s Radiant Suppliers in order to satisfy client's requirement.

#The DeliverElectronicsRawMaterial task where the M/s Fablin Retailers supplies the raw materials (Electronics Items) to M/s HP Ltd.

The DeliverCabinetRawMaterial task where the M/s Radiant Suppliers supplies raw materials (Cabinets and other mechanical Items) to M/s HP Ltd.

The Produce PC DIY task will be carried out only when M/s Fablin and M/s Radiant supplies Electronics and Mechanical items to M/s HP Ltd.

We created 16-service agents namely a1, a2, a3,.....a16 all of which registered with the service selection Agent. In the following portion we will describe how to define the workflow schema, and then create workflow instance and then to execute it by using agent based workflow architecture.

Describing workflow schema: A workflow panel is provided by the interface agent present in the upper right panel as shown in figure 7. The process composer with the help of this workflow panel draws the UML state chart diagram that defines the workflow schema. These workflow schemas are organized using domain specific hierarchy. There are leaf and non-leaf

For the nodes in the left panel, the leaf nodes represent the schema of specific workflows. For an instance (as shown in figure 7) there are four workflow domains namely Supply Chain Management, PC-DY, HI-FI-DIY, Insurance planning. Home PC is a sub-domain of PC-DY. This sub-domain in turn has workflow schemas Game PC and Multimedia PC. Every non-leaf node in the hierarchy represents a set of workflow schema of a particular domain.

Create Workflow Instance: It is possible for a user to access the workflow schema with the help of workflow Schema navigator panel. When the user clicks on the leaf node the respective UML state chart will be displayed on the workflow definition panel. An instance of the workflow can be created by clicking on the create button on the workflow management panel. In order to execute the workflow the user gives the appropriate parameters like Item1 and delivery date. Afterwards the user can generate the XML document (that describes the workflow) by clicking on the generate button on the workflow management panel. Once the user clicks the execution button then the workflow (XML document) will go to the process agent to execute the workflow

Dynamic Integration: The integration agent parses the XML document into five tasks as required by our SCM. For each task it queries the service selection Agent for the relevant service agents. Based on the query results from the process agent we separate each group of

service agent based on the task-group they are responsible to perform .For example in our supply chain process

$T_1=\{t_1\},T_2=\{t_2,t_3\},T_3=\{t_1,t_3\},T_4=\{t_2,t_4,t_5\},T_5=\{t_5\},A_1=\{a_1,a_2,a_3\},A_2=\{a_4,a_5,a_6,a_7\},A_3=\{a_8,a_9\},A_4=\{a_{10},a_{11},a_{12},a_{13}\},A_5=\{a_{14},a_{15},a_{16}\}$.Since there are five set of task sets, the process agent creates five negotiation sessions to negotiate with the service agents. The process agent creates a set of execution plans based on the available bids but only one execution plan is selected and executed.

7 Related work and Conclusion

Approaches related to integrating business processes exist in many fields including component-based E-commerce systems, cross-organization workflow and software agents.

An early solution to B2B integration is Component-based E-commerce systems [14], typically rely on distributed object frameworks such as CORBA and DCOM. Various organizations present their high level services as business objects. The combination of middleware technologies, and the business objects, provides services. It is suitable for integration of small number of tightly coupled applications.

Other advanced approaches to B2B integration is the cross-organization workflow. Related projects in this area includes the project at MCC[15] where they proposed a Service Oriented Process model and the idea is to be able to provide a framework for flexible, plug and play approach to cross-organization workflow composition. However they could not addressed the issue of brokering and selection of services that goes beyond what is stated in the service inter-faces.

The use of software agents is one of the promising technologies in B2B integration and E-commerce applications. The use of agents in automating a single organization WFMS have been discussed in [16][17]. In [16], each workflow is represented by multiple personal agents, actor agents and authorization agents. These agents act as personal assistants which carry out actions on behalf of the workflow participants and facilitating interaction with other participants or organization specific WFMS. In [17], the MAS architecture consists of a number of independent agencies. Each single agency consists of a set of subsidiary agencies which is controlled by an accountable agent. A single agent is able to execute one or more services. These atomic agents can be united to form complex services by adding ordering constraints and conditional control. However, neither [16] nor [17] speak about the agent technology to create workflow execution engine dynamically. The workflow processing logic is hard-coded and thus it is difficult to reuse this workflow execution engine, for other business processes.

Even though a little work has been done for agent integration, they can be extended successfully to include integration capabilities. In doing so, the integration solution can take advantage of the agents' negotiation capability and the ability to adapt to dynamic changes in environments.

In our approach, we have implemented agent based workflow system for dynamic B2B integration. In our approach the agent-community for specific workflow is optimally and automatically composed based on the context of workflow execution and can self-adapt and react to changes during the execution. We show how the workflow agent-community gets constructed by taking the example of supply chain management. We illustrate how the agent community executes the workflow specification and modifies themselves during the execution of the workflows. We also used monitor agents for monitoring cross-enterprise workflows. This facilitates the end user to know the status of the workflow instance.

REFERENCES

[1]. K. Kogan, A. Herbon, "A supply chain under limited-time promotion: The effect of customer sensitivity," European Journal of Operational Research, Vol. 188, 2008, pp. 273-292

[2]. C. Lin, H. Chiu, P. Chu. Agility index in the supply chain. International Journal of Production Economics, Vol. 100, No. 2, 2006, pp. 285–299.

[3]. Simchi-Levi, P. Kaminsky and E. Simchi-Levi, Managing the Supply Chain – The Definitive Guide for the Business Professional, New York: McGraw-Hill, 2004.

[4]. D. Simchi-Levi, P. Kaminski, Designing and managing the supply chain–concepts, strategies and case studies, New Jersey: McGraw-Hill, 2006.

[5]. B. Manouvrier, L. Ménard,Application Integration,EAI, B2B, BPM and SOA ,John Wiley & Sons, Inc., pp. 134-142,2008.

[6]. D. Georgakopoulos, editor. Information Technologyfor Virtual Enterprises, Proc. of the 9th Int. Workshop on Research Issues on Data Engineering. IEEE Computer Community, March 1999.

[7]. T.E rl, Service-Oriented Architecture Concepts , Technology , and Design, Prentice Hall professional Technical Reference ,pp. 33-37, 2009.

[8]. T.E rl, Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services, Prentice Hall professional Technical Reference, pp. 48-50, 2005 .

[9]. F.Curbera, M.Duftler, R.Khalaf,W.Nagy,N.Mukhi, S.Weerawarana, IEEE Internet Computing Magazine, "Unraveling the Web Services Web An Introduction to SOAP, WSDL, and UDDI", 2002.,Avialble : http://ieeexplore.ieee.org.www.ezplib.ukm.my/servlet/opac?punumber=4236, (Last accessed: 12 July 2011)

[10]. P. Massimo, K. Sycara, T. Kawamura, Delivering Semantic Web Services, Proceedings of the WWW2003, Budapest, Hungary, 2003 May.

[11]. M. Fox, M. Barhuceanu, and R. Teigen, "Agent oriented Supply-chain Management," The International Journal of Flexible Manufacturing Systems, Vol. 12, pp.165-188, 2000.

[12]. J. Gou, X. Yang, W. Dai, " On Demand Integration of Dynamic Supply Chain Application Based on Semantic Service Oriented Architecture" , IFIP International Federation for Information Processing, Volume 254,Research and Practical Issues of Enterprise Information Systems II Volume 1, eds. L. Xu, Tjoa A., Chaudhry S. (Boston:Springer), pp. 589-598 , 2007.

[13]. T. Hess, L. Rees, and T. Rakes, "Using Autonomous Software Agents to Create the Next Generation of Decision Support Systems," Decision Sciences, Vol. 31, No. 1, pp. 1-31,2000.

[14]. A. Dogac, editor. ACM SGMOD Record: Special Issue on Electronic Commerce, ACMSIGMOD RECORD. ACM, December 1998.27(4).

[15]. D. Georgeakopoulos, H. Schuster, A. Cichocki, and D. Baker. Managing process and service fusion in virtual enterprises. Information System, Special Issue on Information System Support for Electronic Commerce, 24(6):429-456 1999.

[16]. J. Chang and C. Scott. Agent-based workflow:TRP support Environment (TSE). Computer Networks and ISDN Systems, 28(7-11):1501-1511, 1996.

[17]. N. R. Jennings, P. Faratin, M. J. Johnson, T. J. Norman, P. O'Brien, and M. E. Wiegand. Agent- based business process management. International Journal of Cooperative Information Systems, 5(2&3): 105-130,1996.