Enterprise Modelling using Unified Framework supporting Distributed Object Computing

Manuj Darbari
ICFAI Business School, Lucknow

Bhaskar Karn
Birla Institute of Technology, Mesra, Ranchi

Enterprise Modelling using Unified Framework supporting Distributed Object Computing

Manuj Darbari
ICFAI Business School, Lucknow
INDIA

Bhaskar Karn
Birla Institute of Technology, Mesra, Ranchi
INDIA

ABSTRACT

This paper provides a support for the analysis phase in the life cycle of the enterprise distributed computing systems. The major goal of our work is to provide a small but powerful set of enterprise modeling concepts. We will be extending the concept of Enterprise Modeling using EDOC inter-agent system using OPEN Framework.

INTRODUCTION

Our approach of modeling business process is based on our understanding of what are the critical business issues to be addressed when describing processes in the enterprise. The Model supports the composition of business in a suitable way by providing an association of business tasks with business role to execute them. It uses the concept of Unified Modeling (Reenskang, 2006) to support distributed object modeling. Our approach is based on the introduction of modeling concept that represents dynamic, structural and policy aspects of an enterprise. The goal of our work is to provide:

- Small but powerful set of an enterprise modeling concepts.

- An optimistic unified framework for graphical modeling

- Use of intelligent agent methodology for component based enterprise systems.

We will use the terminology **Action** and **Action Sequence** of Unified Modeling language (Selic, 2000) to define business roles.

Overall, a business role can be defined as a collection of action that are involved in performing one or more business tasks and the grouping of these actions corresponds to the definition of business roles. In this paper we have shown how collaboration process component specification can be used for enterprise distributed object computing. The last section of the paper deals with an example of a traffic monitoring cell and traffic control center coordination in an EDOC environment using Unified framework.

The concept of EDOC started with the development of components collaboration architecture (CCA), which details how the concept of classes (Ashley, Meehan & Carr, 2005), collaborations

and activity graph can be used to model at varying and mixed level of granularity. The entities' profile describes a set of UML extensions which can be used to model entity objects that can define compostable components. The Business Process Profile which specializes in CCA describes a set of UML extension which can be used to model distributed components. XML can encode the tagged values and stereotypes of UML EDOC Components which can be stated as << profile>>. There are various consortiums for development and promotion of EDOC like:

CBOP

CBOP is a consortium of Japan which is used for sharing business information. The current work is going on in developing a UML for modeling patterns using EDOC.

DSTC

DSTC has used its dMOF product to develop a MOF repository and Human Usable Textual Notation I/O tools that support modeling of Business Processes conforming to the metamodel. Significant Business Process models have been created using these generated tools, and mapped using XSLT into XML workflow process definitions, which execute on the DSTC's Breeze workflow engine. dMOF is a commercial product installed at many customer sites world-wide, and Breeze is in development and is currently being beta-tested by four DSTC partner organizations.

EDS

EDS developed the Enterprise Business Object Facility (EBOF) product in conjunction with work on the Business Object Facility specification. This product serves as a proof of concept for important aspects of this specification. It incorporated UML models as the basis for generating executable, distributed, CORBA applications. This involved consideration of transactions, persistence, and management of relationships, operations on extents, performance optimization, and many other factors. This product was sold to a major software vendor.

IBM

IBM has extensive experience in enterprise architectures, Java, Enterprise Java Beans, CORBA, UML, MOF, and metadata. The WebSphere, MQ, and VisualAge product lines provide sophisticated analysis, design, deployment, and execution functionality embodying all of the key representative technologies.

IONA

The Relationships Profile is based on many years of modeling experience in industry and in the development of related products and standards.

The «Port» stereotype has been introduced in EDOC for clarity and brevity, defining in a common ancestor, the tagged Values corresponding to attributes of Port in the metamodel, and reused along the stereotypes specialization of «Port» : «FlowPort», «ProtocolPort», «MultiPort», and «OperationPort».The concept of OperationPort in the metamodel is represented by a standard UML operation. The OperationPort is constrained to contain only FlowPorts. The

signature of the UML Operation representing an OperationPort is derived from the type of the one and only FlowPort of the OperationPort, with direction="initiates." For each Attribute of the FlowPort, the UML Operation will have an input Parameter with type equal to the type of the Attribute in the FlowPort.

For each ownedFlowPort with direction="responds" and postCondition="Success," the UML Operation will have return Parameters with same type as the type of the FlowPort.

All other FlowPort in the OperationPort with direction="responds" correspond to raised Exception Signal of the UML Operation. The structure of the Signal is derived from the FlowPort type : the Signal will have Attribute with same name and type of the Attribute of the type of the FlowPort. CCA models may be diagrammed using generic as well as CCA specific notations. The generic notations (as found in UML 1.4) are supported by a wide variety of tools that allow CCA concepts to be made part of the larger enterprise picture without specific tool support. When using generic notations the CCA profile stereotypes should be used. CCA aware design & implementation tools may provide the CCA specific notation in addition to or instead of the other forms of notation.

BUSINESS ROLE

Business role is desired from the UML (Darbari & Medhavi, 2007) concept of Action, which enables a specific behavior. The behavior can   be expressed using different languages with a UML Class to provide it with a structural description. It can be divided into two different types of roles the Performer Role and Artifact Role. A performer role describes behavior for carrying out task in an enterprise. These entities will be responsible for the execution of some aspects of the tasks. While Artifact roles have behavior, it describes the execution of any action. Artifact roles are used to represent inanimate things in the system such as resources (See Figure 1).

Figure 1: Class Diagram.
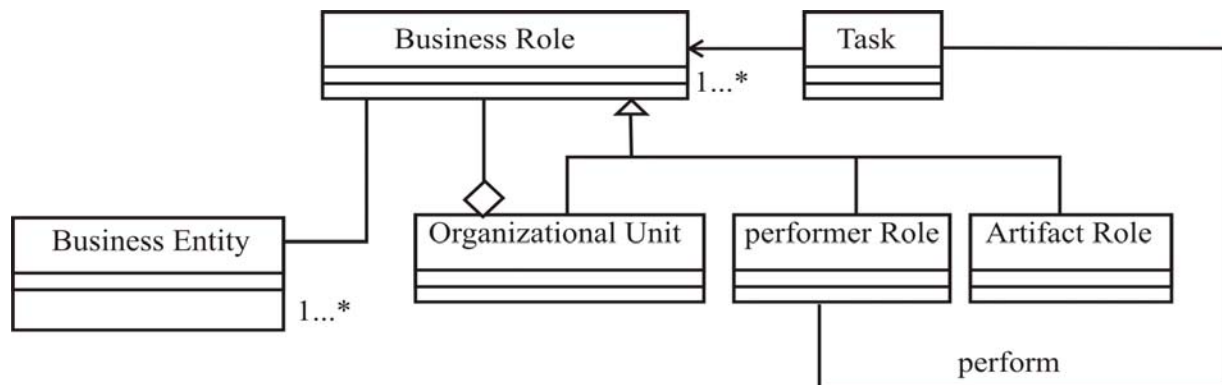
A Business entity describes an actual object that can carry out the business role. A business entity properly set is used for specifying non-functional requirement of the behavior of the Business role. Substantiation of a business role is achieved by binding a business entity that is able to fulfill the behavior specified by business role. Some non-functional characteristics of a business role may be specified as a business entity properly set. Bindings between roles and objects can be statically defined in the business entity model.

A Business event type is a declaration that provides the names and types of properties to be included in events that confirms to this type. Business event types have a name and a domain. Business event type inherits from UML State Machine :: Event, (Shafaei & Aghaee, 2008; Shlaer & Mellor, 1992) which enables the specification of the names and types of the properties that conforming events will contain.

A Business event source defines the Business events type to be committed the conditions under which a business event of that type is committed, Low the values in the event are obtained from the sources state. A Business event sink defines the Business event type to be received, the conditions under which such events should be received, loss the event's values are assigned to the sink's state and the extent. All events are based on actions of interest, but not all actions of interest will be need as events.

DATE FLOW OF EVENTS

The value of inputs and outputs are normally transmitted vial data flows. Events enable the values of inputs and outputs to be transmitted or received beyond the scope of the containing compound task. Such business event sources and sinks behave similarly to the sources and sinks of data flows for the purpose of propagation of data and determining whether an input/output set is satisfied.

The last part in Business Event Model is the definition of task. Activity scan be divided into two sub types : Simple Activity and Compound Activity. A simple Activity refers to an task that is carried out further retirements at this level of abstraction. Simple Activity can be divided into three subtypes : Application Activity, Invokes Activity and Terminates Activity.

Application Activity behaves like wrappers to enable applications while Invoker and Terminator task are used to liaison with other activities within a compound Activity they are generally used to provide dynamic behavior within a particular Activity. An example of simple Activity contains Activity Name, Activity functions, Output Name & Type output set and exceptions. They can be graphically represents by Figure 2.

Figure 2: Framework of Simple Activity.
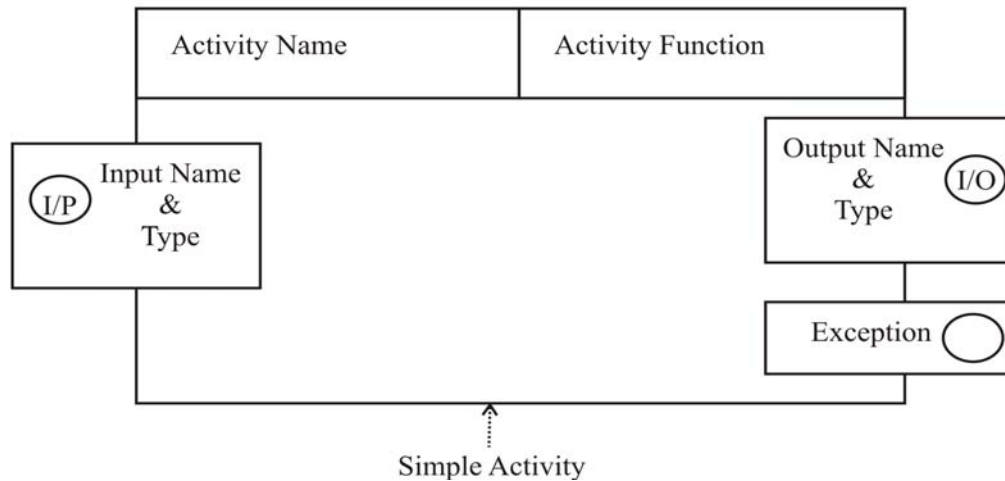
Simple Activity can be divided into three basic features:

- Input Set Model : Which models the information required to commence execution of task as a set of name- value pairs.
- Output Set Model : It models the possible outcome of the execution of tasks.
- Exception Model : It indicates the task status in case of failure of execution.

While the compound Activity Contains various simple Activity inside a single framework and can be represented, see Figure 3.

Figure 3: Compound Activity.
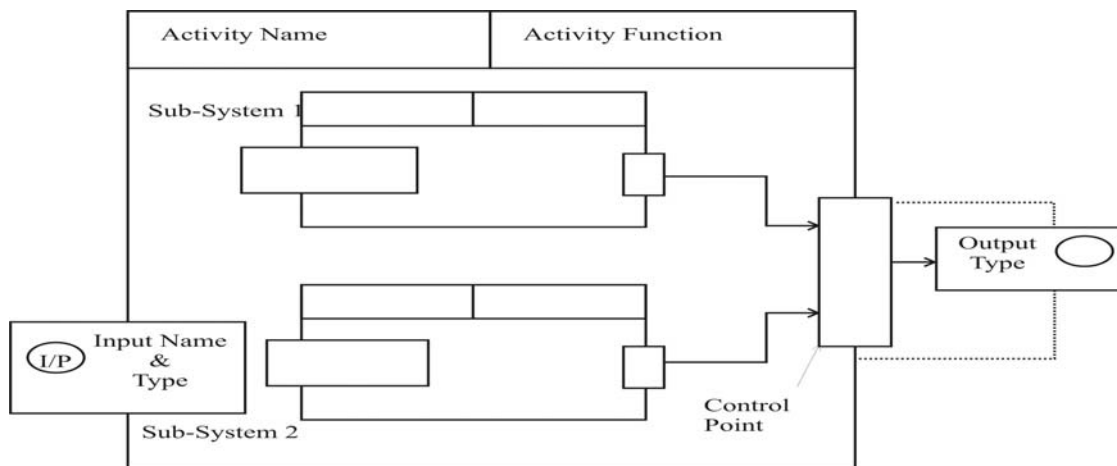
A compound class completes when all of its contained tasks have either completed their execution or are unable to execute. Normally, the completion of a compound task results in one of its output set being suitable to be enabled.

UNIFIED FRAMEWROK FOR MULTIAGENT MODELING

In order to support a Multiagent framework (Zheng, Wang & Yang, 2006) we introduce a modified framework of compound activity with Input being received by Multi Agent Iteration cycle (Yoon, Ko, Han & Youn, 2007).

Figure 4: Modified Compound Activity Diagram.
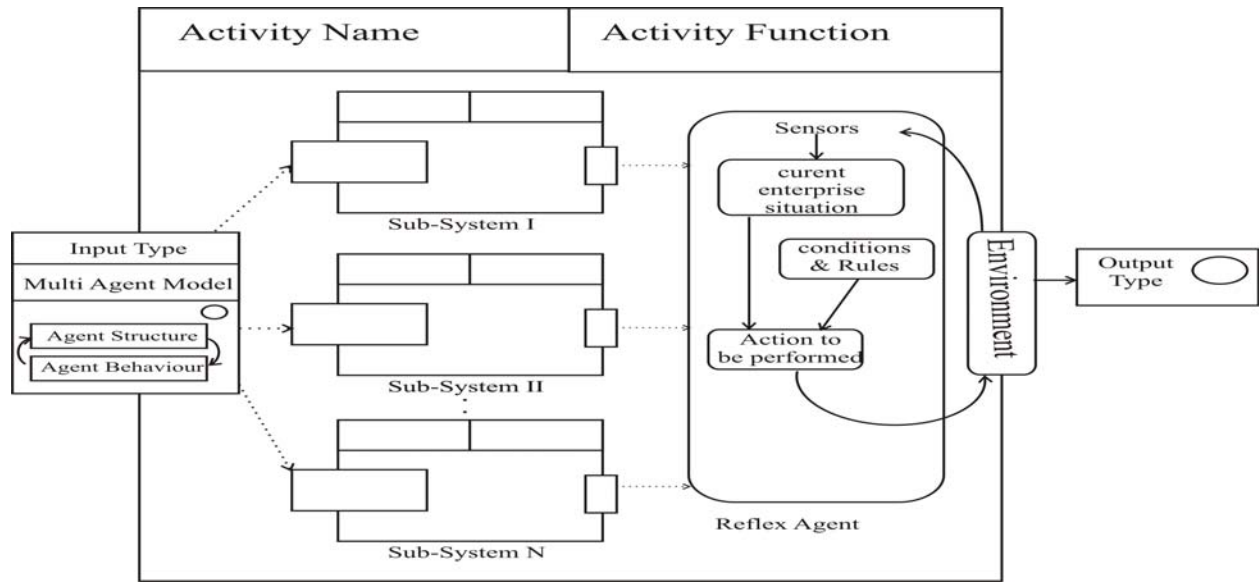
The output of Multi Agent Iteration goes as input to the various sub-systems which then passes the output in cascaded mode to the Reflex Agent in which the real time input conditions are continuously provided by the sensory agents. Based on the output of the various subsystem and the condition and the rules set, it generates the output.

The short algorithm for Reflex Agent can be written as: Function Reflex - Agent returns action (see Table 1).

Table 1: Reflex-Agent Action Algorithm.

State ← UPDATE - STATE- I

Rule ← RULE-MATCH

Action ← RULE_ACTION

State ← UPDATE_ STATE - II

Return Action

The Several roles that an agent can play are introduced as objects in the appropriate sequence diagram. An Agent may participate in several scenario playing distinct roles in each, and may

even appears as more than one object in a single sequence diagram. The Inter-object message in the sequence diagram either represents events generated by the external environment or system or parts of communications between the roles of one or more agents. OPEN sequence diagrams also have distinct icon shapes for the CIRTs at the modes which indicates whether the time baseline is a single, multiple CIRTs. The White-Box sequence diagram represents a more detailed Interaction Diagram representing the end-trend processing, initiated by each major system operations (see Figure 5).

Figure 5: Black-Box Representation of a Modified Compound Activity Diagram.
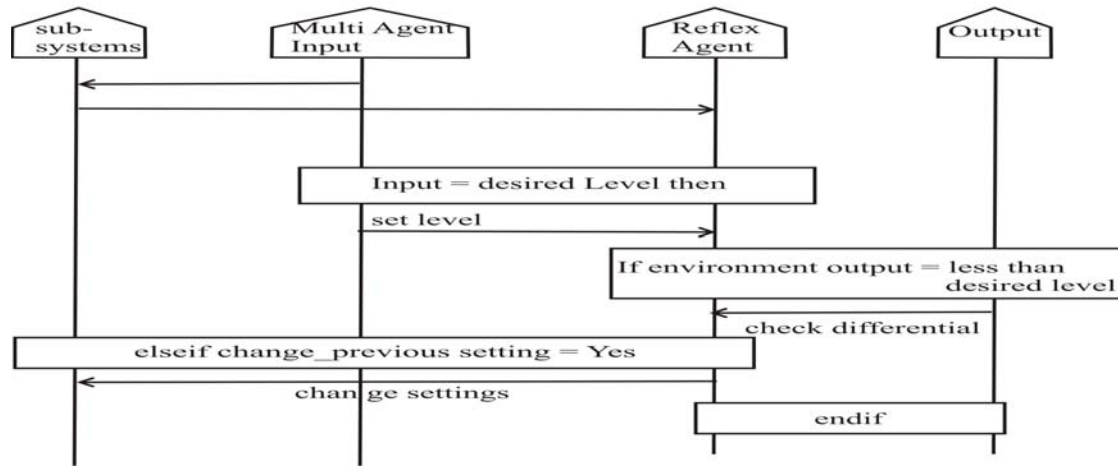
To make the compound Activity Diagram more refined and simple we formalizing the entire sequence   by using Activity Oriented Modeling (AOM) (Wahle, 2001). In this the basic terminology was for Activity Theory and then model it with OPEN (Liu, Dong & Sun, 2006) Class Framework. The linking of various Activity Theory Notations with Business entities in the form of Agents is shown in Table 2.

Table l2: Mapping between A.T. Notation and Business Entities.

A.T. Notations	Business Entities (Agents)
Activity	Business Transactions Showing a small circle at I/P & O/P.
Subject	Business objective
Outcome	Desired output level of the Organization
Tools	Subsystems
Artifacts	Reflex Agent

## DEALERS AND RTO B2B COLLABORATION EXAMPLE USING UNIFIED FRAMWORK

In order to support our framework we take the simple example of Vehicle dealers (Quan, 1989) being collaborated by Road Traffic Office (RTO) which is the registration authority of the vehicles sold. The basic internet computing model (Vilain, Schwabe & de Souza, 2007) can be shown by the B2B framework as shown in Figure 6.

Figure 6: General Components of B2B framework.
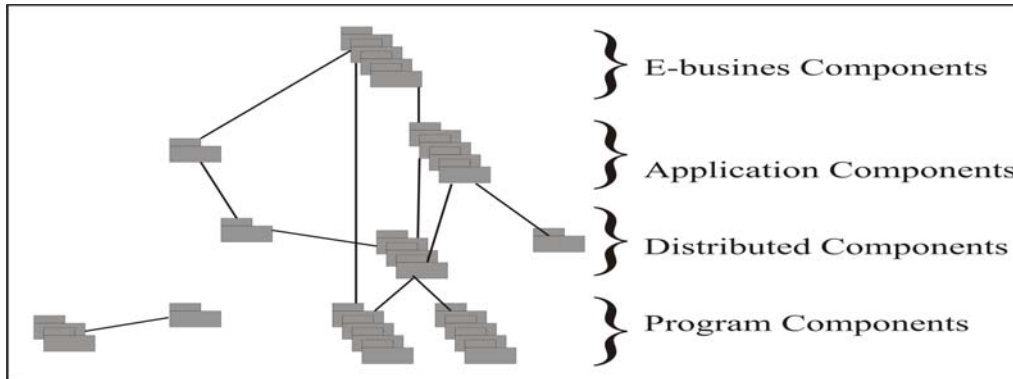
It shows how the necessary document interchange over internet like type of car, its engine number etc. This is followed by Platform independent providing a holistic view of the Unified framework (Foster & Zhao, 1997) shown in Figure 7.

Figure 7:   Patterns for Model Driven Architecture.
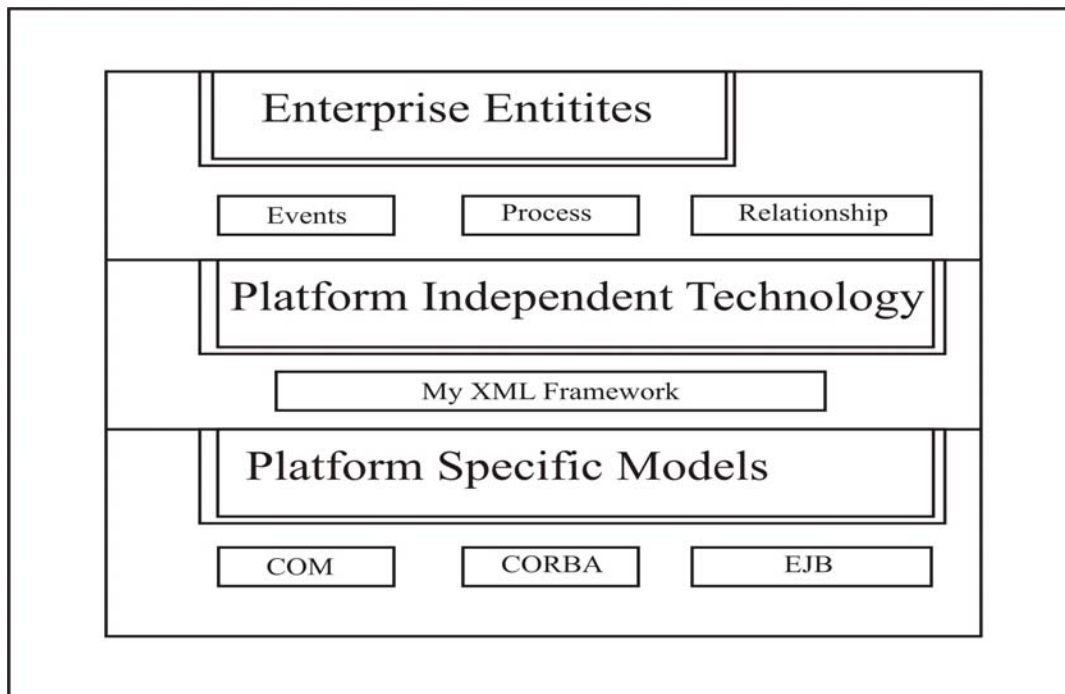
The EDOC structure consists of process components and protocols. It defines configurable behavior units that can be composed in particular way. In this particular section OPEN provide a core architecture for process entities and events at multiple levels of granularity. The framework is also compatible with ebXML business Process specification schema which specifies B2C (Bhatt, Gupta & Sharma, 2007) collaborations.

The B2B process to be represented in OPEN framework can be defined as composite Activity diagram with RTO, Dealer and a Purchaser. This can be further reified by component level diagram giving details about the collaboration of various activities. The Dealer and RTO collaborate by the help of DealerRTO IP Protocol (Li, Chong, Chan & Hallan, 1996) providing a link between the two parties (see Figure 8).

Figure 8: Component Structure of Protocols.
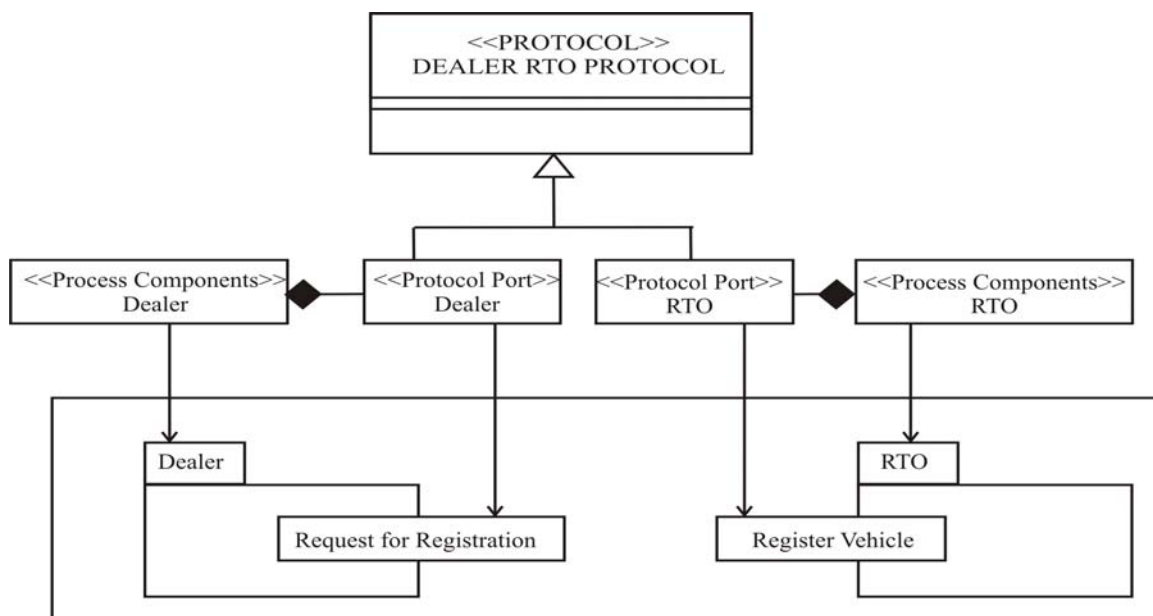
The meta-model structure can be defined by the help of UML and Activity Diagram representing the ports and connectors (Fu, Y., Dong, Z. & He, X, 2006) between dealers and RTO shown in Figure 9.

Figure 9: A Metamodel for Protocol interaction of Registration Process.
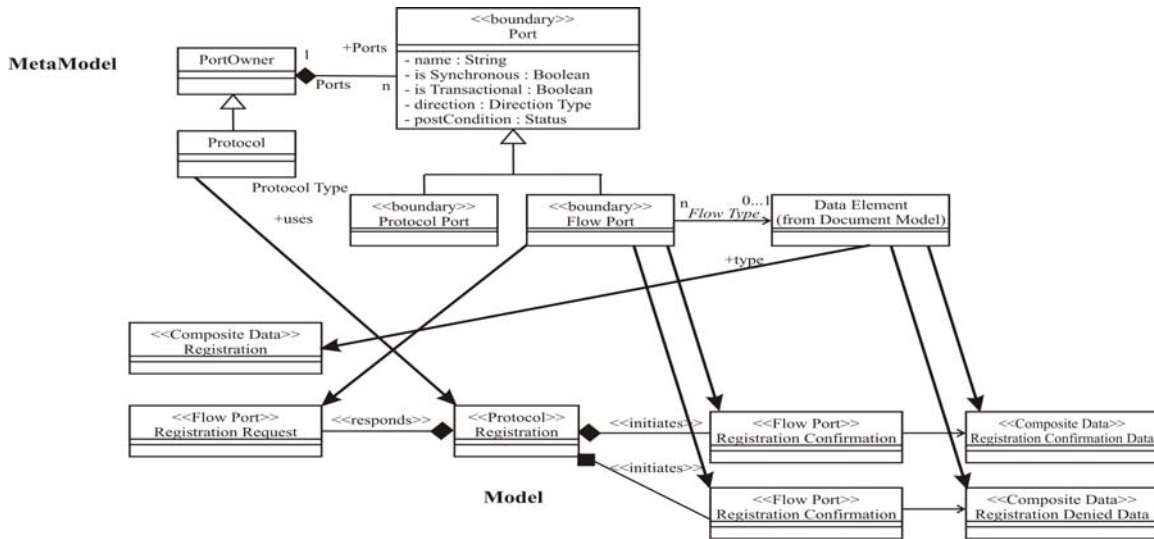
The collaboration between the various protocols describes the sequence of activity taking place. Each step of Activity Diagram can be represented by the help of UML Diagrams (Horling, Lesser & Vincent, 2000). The protocols are divided into two parts the Flow protocol and the Connector protocols which are also responsible for achieving connectivity between the agents collaborating. The last step of collaborative B2B framework consists of Assigning Timing for the port connectivity (see Figure 10).

Figure 10: Initiating Timing in B2B Collaborations.
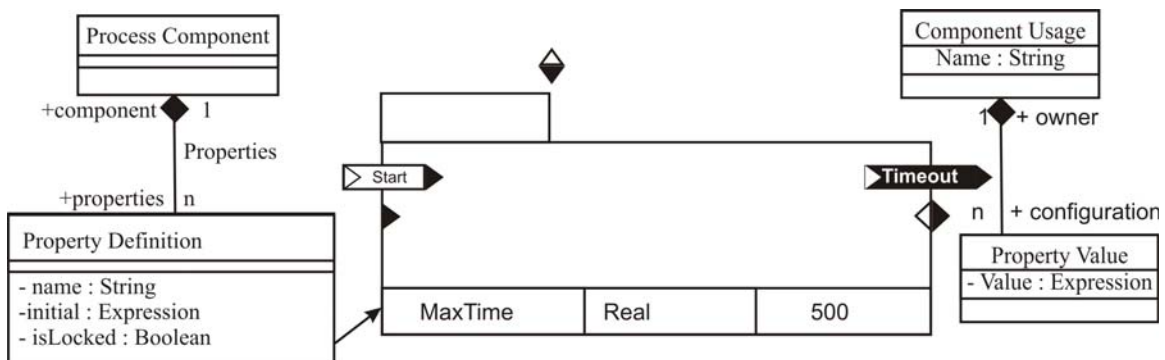
This can be achieved by assigning timing for each port being engaged providing a constraint on the dealers to complete the registration process within certain time frame and free the connectivity port for another dealer for subsequent registration process shown in Figure 10. It makes process reconfigurable and can reset when the components have been used.

CONCLUSIONS AND FUTURE SCOPE

This paper emphasizes on Business Process Modeling by using the concept of Activity Theory and UML modeling framework (McManus & Snyder, 2003; Luck & d'Inverno, 2001) . It is to describe how different objects are related to each other both in terms of static and dynamic relationships. The use of compound artifact allows a more flexible approach in implementing and modifying business processes. It provides a platform where any organization can be graphically represents by the help of AT + OPEN concept. The use of distributed objects bring new capabilities of object-oriented modeling in that the objects can be developed independently by different parties and they can interact by sending messages over the network. Using the concept of Roles it is possible to describe an enterprise in terms of fragments of behavior that corresponds to positions in the organizations. The benefit of this approach is that they provide many options for assigning objects to roles, even during run-time binding. It also allows the exploitation of event-based interactions between objects. This mechanism is particularly suitable to be used as a way of communicating business events among the parties involved in enterprise.

Our approach also provides an integration of unified methodology(UML) with distributed object computing (EDOC).We also support the role of intelligent agent in EDOC concept which can be used for design and implementation phases of object oriented cycle.

 The future extension could be for distributed object modeling by using the conversion of entire object Modeling representation into XML format using MyXML compiler providing Platform and Device Independent distributed modeling for Ubiquitous environment. It can also be integrated with UML concept of capsule, Port, Connectors and Business Protocols.

REFERENCES

Ashley, M. W., Meehan, T. E. & Carr, N. (2005). UML Activity Diagram Semantics and Automated GUI Prototyping. *Journal of International Technology and Information Management,* 14(3), 43-54.

Bhatt, G. D., Gupta, J. N. D. & Sharma, S. K. (2007). Integrating IT-enabled social networks with transaction cost economics and the resource based view of the firm. *Journal of International Technology and Information Management,* 16(2), 27-44.

Darbari, M & Medhavi, S. (2007). Application of UML for modeling urban traffic system using producer consumer theory to generate process algebra model. *Journal of International Technology and Information Management,* 16(4), 75-82.

Foster, T. & Zhao, L. (1997). Modeling transport object with patterns. Prentice- Hall, India.

Fu, Y., Dong, Z. & He, X. (2006), Formalizing and validating UML architecture description of web systems, ICWE '06: Workshop proceedings of the Sixth International Conference on Web engineering.

Horling, B., Lesser, V. & Vincent, R. (2000, August). Multi-Agent System Simulation Framework, *16th IMACS WorldCongress 2000 on scientific Computation, Applied Mathematics and Simulation.*

Inrona, L. D. (2003). Cooperation, Coordination and Interpretation in Virtual Environments: Some Thoughts on Working Together. *Cognition, Technology & Work*, 3, 101-110.

Li, M., Chong, K.W., Chan, S. & Hallan, J. (1996). Agent –Oriented Urban Traffic Simulation. *The 1st International Conferences on Industrial Engineering Application and Practice, 1996.*

Liu, Y., Dong, L. & Sun, Y. (2006). Multi-agent System Based on the Extended ConGolog, *Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06)*, 2, 913-918.

Luck M. & d'Inverno, M., (2001). A. Conceptual Framework for Agent Definition and Development. *The Computer Journal*, 44(1), 1-20.

McManus, D. J. & Snyder, C. A. (2003). Knowledge Management: The Role of EPSS. *Journal of International Technology and Information Management,* 12(3), 17-28.

Quan, Y. (1989). Urban Traffic Control, China Communication Press.

Reenskang, T. (2006). Working with Objects, Prentice Hall, USA.

Selic, B, (2000). Using UML for developing complex Real Time Systems. In P*roceedings of ECOOP*, USA.

Shlaer, S. & Mellor, S. (1992). Object Lifecycles: Modeling the World in States, Prentice Hall, USA.

Shafaei, S. & Aghaee, N. G. (2008). Biological Network Simulation Using Holonic Multiagent Systems. *Tenth International Conference on Computer Modeling and Simulation (UKSIM-2008),* 617-622.

Vilain, P., Schwabe, D. & de Souza, C.S. (2007). *Use Cases and Scenarios in the Conceptual Design of Web Applications*. Technical Report.

Whale, J. (2001). *A Multi-Agent System for On-line Simulations based on Real-World Traffic Data" in Proceedings of (HICSS),* IEEE Computer Society.

Yoon, Y. S., Ko, H., Han, S. W. & Youn, H. Y. (2007). Priority-Based Message Scheduling for the Multi-agent System in Ubiquitous Environment. *IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology – Workshops*, 395-398.

Zheng, Y, Wang, K. & Yang, J. (2006). A Multi-Agent System for Manufacturing Material Resource Planning. *Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06),* 2, 1118-1123.

This Page is Blank