2015 International Conference on Advances in Computer Engineering and Applications (ICACEA)
IMS Engineering College, Ghaziabad, India

Modeling of Services and their Collaboration in Enterprise Cloud Bus (ECB) using UML 2.0

Gitosree Khan
B. P. Poddar Institute of Management & Techology
137,VIP Road,Poddar Vihar
Kolkata-700052
Khan.gitosree@gmail.com

Sabnam Sengupta
B. P. Poddar Institute of Management & Techology
137,VIP Road,Poddar, Vihar
Kolkata-700052
sabnam_sg@yahoo.com

Anirban Sarkar
National Institute of Technology
Durgapur, India
sarkar.anirban@gmail.com

Abstract— **Cloud computing refers to a web service model that delivers on demand based services over the network. Agent based Cloud computing technology has gained more importance recently due to the rapid increase in no of clouds and their services. Modeling and Design methodologies for agent based cloud infrastructure has been a challenging domain. In this paper we propose to model the interaction and collaboration of services in an agent based abstraction framework, called Enterprise Cloud Bus (ECB), using UML 2.0. ECB is an abstraction layer of SaaS architecture, and it is modeled using UML 2.0 focusing on service interaction, service collaboration, service discovery, service scheduling and service dispatching.**

Keywords—ECB, CESB, CUDDI, HUDDI, UML

I. Introduction

Cloud Computing has emerged as one of the most challenging paradigms for design and development of information systems today. By providing on demand access of services to a distributed environment of computing resources in a dynamically scaled and virtualized manner, agent based cloud computing offers compelling advantages in cost, speed and efficiency. Therefore the need for organizations to leverage cloud computing as part of their integration strategies is growing exponentially. With the advancement of cloud computing the number of cloud services are also increasing. Though, Enterprise Service Bus (ESB) technology provides an abstraction layer on top of an implementation of an enterprise messaging system, but still it is expected to have limitations like virtualization, resource pooling, scalability and it also increases overhead and slower the communication speed of many compatible cloud services with the increase in no of clouds and no of services in a cloud.

To address performance issues because of increasing number of clouds and services in a cloud, the concepts of Enterprise Cloud Bus (ECB) framework, an abstraction layer of Software as a Service, is proposed for the development of resource pooling, virtualization and dynamic provisioning. Modeling service oriented systems are a key part of all activities that lead up to the deployment of software.

We build models to communicate the desired structure and behavior of a system and to visualize the control systems architecture. The most important characteristic for modeling a system is to seize the dynamicity of the software architecture.

In this paper, UML is used to model the agent based abstraction layered architecture called Enterprise Cloud Bus (ECB) to depict the dynamic, automated and virtualized behavior of the system and also to make the system easy to understand and use by the industry developers and business users. The methodology proposed in this paper, is aimed at building models of Enterprise Cloud Bus services and models of collaborations of such services and the agent. The goal of the methodology is to show how the ECB services are accessible by the consumers according to their service level agreement. In this paper, we have used few structural and behavioral modeling using UML 2.0 to model services, agents and their collaboration using communication, state chart and deployment models and capture the dynamic nature of the ECB framework.

II. Review of Related Work

According to [1], UML is adequate for modeling service oriented (SO) systems. A considerable number of agent-oriented methodologies and tools are available today, and the agent community is facing the problem of identifying a common vocabulary to support them is shown in [2]. In [3], the author presents the modeling and methodology of agent oriented system analysis and design. In [4], the author presents the UML as a standard modeling language for visualizing, specifying, specifying constructing and documenting the elements of systems in general, and software systems.

In [5] the author presents a proposal for a Systematic Approach for Agent Design by using UML. The author in [6] have discussed regarding various methodologies for agent design for requirements. In [7], Kinny et al. propose a technique for modeling agent based system. In [8] the author presents a proposal for modeling agent mobility using UML Sequence Diagram. In [9], the author extended Activity Diagrams in UML 1.5[10] to capture algorithmic behavior of mobile agents with location based on a new structural interpretation of swimlanes. The latest enhancements of UML in 2.0 [11] will provide new concepts for modeling, and their associated new model elements which can be readily used for various system developments.

The paper [12] proposes <<agent>> as a stereotype in the object notation to denote an agent class and its instance. The paper [13] describes UML4SOA which extends the UML by providing the possibility for behavioral specifications of services, focusing on service orchestrations. The author in [14] gives an idea of modeling the mobile agent system using UML 2.0. In [15] the author facilitates the business process expert in modeling the security requirements along with the business process modeling. The paper [16] develops a methodology of migration using UML modeling. In [17], the author presents a novel idea about cloud services for UML models verification.

The author in [18] has proposed a Petri Net based model and analysis methodology based on that conceptual framework to analyze the crucial behavioral feature of such system. In [19], a conceptual framework for Multi-Agent System (MAS) architecture has been proposed for quality evaluation of such system at the design level. In [20] this paper, a graph based specification called, Multi-Agent System Architecture Graph (MASAG) has been proposed to model such dynamic aspects of MAS. In [21] a high level Petri Net called HMAP is proposed to model the behavior of MAS. Proposed HMAP is effective towards modeling interactions among the heterogeneous agent in MAS environment. Faisal et.al. [22] [23] have also designed an abstract web service execution machine, which uses web-service family. In [24], we propose an abstraction layer, Enterprise Cloud Bus (ECB) leveraging on multi agent technology and has been modeled using UML 2.0. We, in [25] have proposed a dynamic service discovery from ECB that supports the identification of service during the run time of service-based systems using a relational model.

This paper presents a semi formal representation using UML for modeling services and collaboration of agent based ECB architecture like communication model, state model and deployment model. Here, the purpose of using UML in ECB architecture is to give a conceptual overview regarding system's architecture and covers business process and system functions. In order to build software development process more efficient it is necessary to know the importance of service collaboration, modeling and their applications.

III. A Brief Description of ECB Framework

Cloud Enterprise Service Bus (CESB) and Enterprise Cloud Bus (ECB) in [24] are the abstraction layers of Software as Service (SaaS) architecture in cloud computing environment. CESB is the extension of ESB that enhance the ESB's to register their services for single cloud environment. ECB is a hierarchical layer of SaaS architecture that extends the CESB's to register their services from various locations through cloud agent for inter cloud environment and are shown in figure 1.

The following subsections portray briefly the building blocks of the ECB system:

1) Service Consumer: Service Consumer is the end-users in cloud computing environment. Here, the consumer placed the request for service in Service Level Agreement (SLA) that is made between the consumer and the service provider. In the other hand, a provider agent is deployed in ECB to invoke the request from the customer and publish it in Cloud Universal Description Discovery and Integration (CUDDI), a meta service registry. During this process a timestamp is maintained to track time for several consumer registration requests made by the provider agent in CUDDI.

2) Service Provider: A Cloud Service Provider is the entity responsible for providing web services available to customer. A cloud agent is deployed for collecting various services from different cloud service providers based on various locations, context, etc in SaaS platform and publishes the services in Hierarchical Universal Description Discovery and Integration (HUDDI), an extended meta service registry of CESB's in ECB.

3) Service Scheduler: A Scheduling Agent is deployed in ECB to configure, discover and schedule the cloud services as per Quality of Service (QoS) parameters.
The ECB architecture is shown in figure 1.

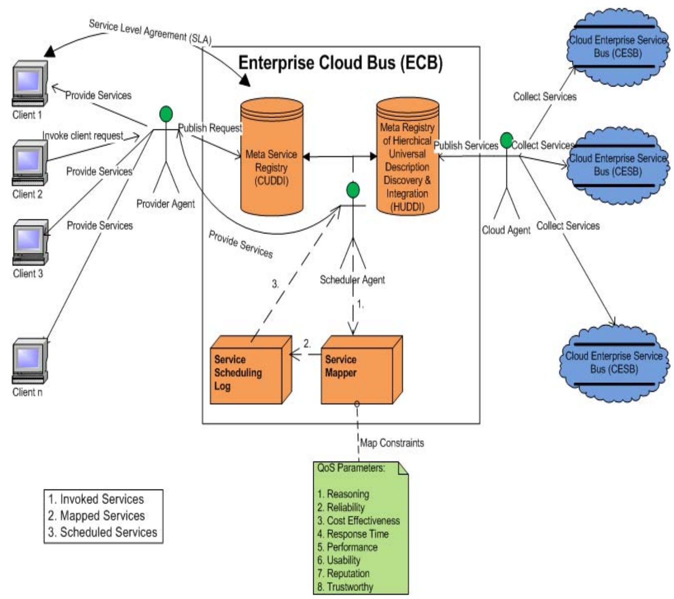

Fig. 1. Enterprise Cloud Bus framework

IV. Scope of Work

In this paper, we have focused on:-

1) Service Modeling Enterprise Cloud Bus (ECB) using UML 2.0.: This paper presents a structural view for ECB. Cloud services encapsulate business services; an enterprise unit focused on delivering web service that is benefit to a client, to enhance communication, coordination and service management.The proposed models is based on a semi- formal, object flow oriented description of cloud services that provide the basis for a formal analysis to verify behavioral properties of services. It has the ability to describe both the structural aspects of an enterprise, the behavioral aspects of an enterprise, and the enterprise rules that affect both structure and behavior. It will be beneficial for both business and software modeling by facilitating communication between business modelers and software modelers. These proposed models provide a simple view of the cloud services, its interaction and collaboration between various object of the framework.

2) Case Study: This approach is explained with the help of a case study on airline reservation system. Here, web services are being used for the coordination of communicating different objects of the airline reservation. This UML structure can be used by the users for collaborating and communicating services for airline reservation system and can be used for other service industries as well.

V. Proposed Modeling of Services & Collaboration in ECB Framework

We have discussed in Section III, the abstraction layer of ESB framework in case of single and multi- cloud architecture that models the services, agent and their interconnections for all the locations. The Enterprise Cloud Bus Architecture (ECB) is a SaaS architecture based on UML models that explicitly addresses the challenge of integration of heterogeneous services, and we therefore choose UML as a basis for our modeling, to ensure compatibility with this approach. In this paper various modeling services and collaboration in ECB helps in visualization, validation and clear communication among the web service.

B. Communication model of CESB & ECB

Cloud Enterprise Service Bus (CESB) and Enterprise Cloud Bus (ECB) are the two hierarchical layers of ESB's in [24][25], that helps in enabling publishing and registry of services in cloud computing environment for SOA-enabled systems to leverage on agent technology of cloud computing. A Communication model depicts the overall situation of communicating objects at an abstract level.

The Communication Diagram represents the interactions between system objects. The objects that are communicating in the ECB framework are Cloud, ESB, Provider Agent and CESB.

The steps for object communication in CESB are as follows:

- Initially several cloud services are registered onto the Enterprise Service Bus.
- The registered services are then collected by the provider agent and publish them into the CESB, the abstraction layer of ESB.

The steps for object communication in ECB framework is as follows:

- Client sends a service request through message call.
- The Provider Agent invokes service requests from several clients and publishes into the Meta Service Registry (CUDDI).
- Simultaneously, several CESBs of various clouds provide services to the Cloud Agent.
- The Cloud Agent collected and publishes those services into the Meta Service Registry (HUDDI).
- The Scheduler Agent initially discovers those services that are requested by the clients from the HUDDI and sends them to the Service Mapper.
- The Service Mapper maps those discovered services on the basis of various Quality of Service (QoS) parameters such as reliability, performance, time, cost, etc. The Service Mapper is activated only after services are invoked by the Scheduler Agent, and deactivates soon after it provides the result to the Service Scheduling Log.
- Once the mapping is done, the Service Mapper sends those mapped services to the Service Scheduling Log. The Service Scheduling Log is activated only after mapped services are provided to it by the Service Mapper. It deactivates soon after it provides the scheduled services to the Scheduler Agent.
- The Scheduler Agent collects the matched service from the service logger and finally dispatched it to the provider agent.
- Finally, the Provider Agent sends the request service to the respective client.

The Communication model for CESB and ECB are shown in figure 2 and figure 3.

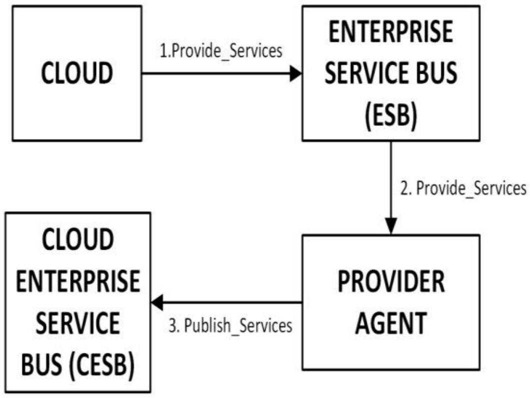

Fig. 2. Communication Model for CESB framework

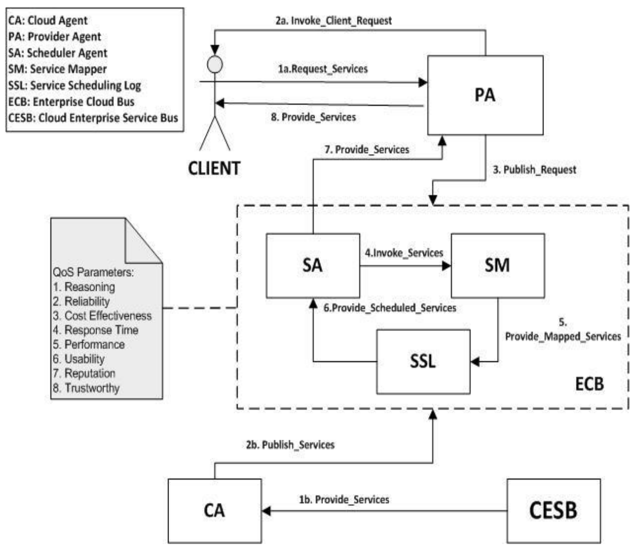

Fig. 3. Communication Model for ECB framework

C. State model of ECB

The State model of ECB shows the dynamic behavior of object or class of objects present in the system. In this section each state represents object condition at certain point of time and the event of the state represent incidents that makes the object to move from one place to another. In figure 4 each transition line is marked with the event that causes the transition. Actions occur when an object arrives in a state.

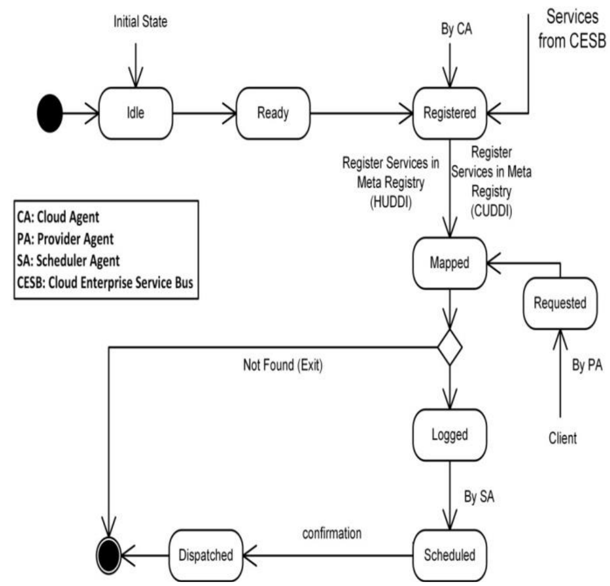

Fig. 4. State Chart Diagram for Service object

Figure 4 shows different states of the Service object such as Idle, Ready, Registered, Requested, Mapped, Logged, Scheduled and Dispatched. The Idle state is the initial state of the object, which then moves into the Ready state. The Cloud Agent collected the service from the CESB and then it moves to the registered state and is being stored into the Meta service registry (HUDDI). Simultaneously, the Service goes into the Requested state on being requested by the Client. The Service object achieves the Mapped state on being invoked and mapped into the ECB. After mapping is done, Service obtains two different states – (i) on successful mapping, the Service obtains the Logged state, then it goes into the Scheduled state. On confirmation, the Service obtains the Dispatched state and then final state, Exit. (ii) On failure, the service goes into the final state, which is Exit.

D. Deployment model of ECB

Cloud services can be deployed in different ways, depending on the organizational structure and the provisioning location. Figure 5 represents the several devices and software that are used for the deployment diagram of ECB framework. The devices are Client, CESB and ECB. ECB and CESB are cloud servers. The ECB can be subcategorized into several other devices – CUDDI, HUDDI, Service Mapper and Service Scheduling Log. The Client program runs on the browser through which service requests can be sent in the form of messages. The browser(s) can be run on and supports HTML, XML, and JSP etc. The CESB is a web server that uses databases containing service descriptions from several service providers. The CESB runs on RDBMS environment that uses MySQL, Oracle, SQL, Enterprise DB, Microsoft SQL Server, Informix etc. The Meta Registries are RDBMSs that store service descriptions in the form of high level query. The Service Mapper is also a RDBMS that maps the services invoked by the Scheduling Agent, on the basis of QoS Parameters.

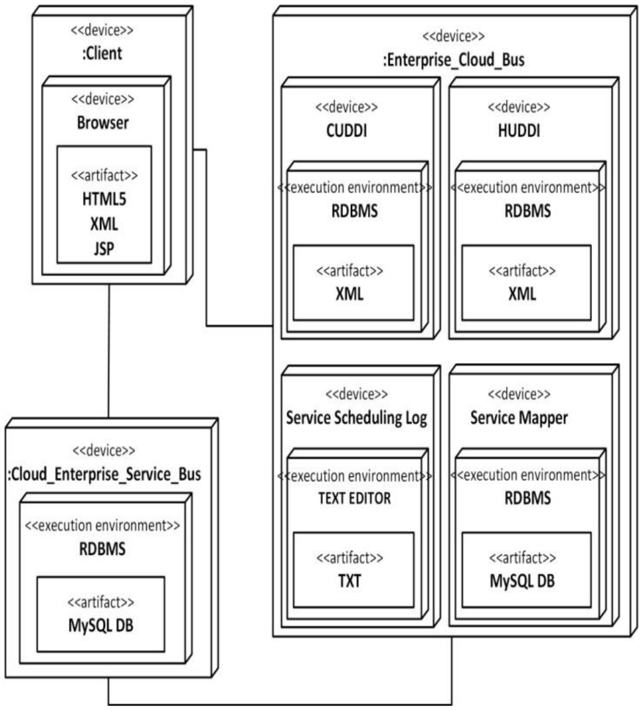

Fig. 5. Deployment Model for ECB framework

VI. Case Study

We establish our approach with the help of a case study of an airline reservation system. Here we provide the UML modeling of ECB. This application is used to maintain flight details, flight status and reservation process. The flight status which maintain flight ID, flight name, arrival time, departure time and also it contain details about the seats such business class seats, economic class seats. The flight detail contains the details about needed flight name as well as the details about the seats. Flight reservation which contains the flight ID, ticket number, passenger name, destination, flight name, business and economic class seats, travel charge, passport number, date of travel are reserved. Finally a report is generated about flight details, flight status, and reservation tickets.

Major features provided by the system are:

- Enquiry about the flight details : The airline reservation system allows the user to perform flight inquiry such as flight scheduling, availability, status, fare details, etc.
- User Registration : The system allows the user to register as a new user who can book or cancels flights.
- Reservation of Flight: The system allows the user to book the flights as per their requirements.
- Flight Cancellation: This functionality can be used by the users to cancel the flight.
- Administration: The administration functionality of the system allows the admin to manage the flight details. It provides the admin to update or change the existing flights.

A. Communication diagram for Airlines Reservation System

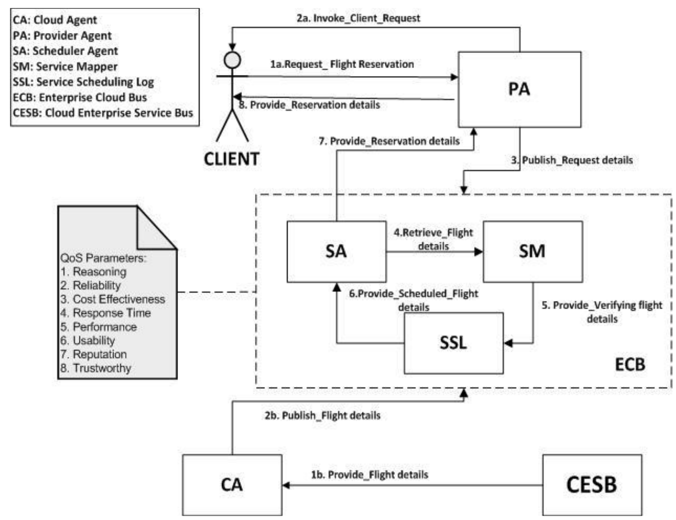

Fig. 6. Communication Model for airline reservation system

Figure 6 shows the objects communication in airline reservation system. This model is of great use in representing the object associations as links.

B. State diagram for Airlines Reservation System

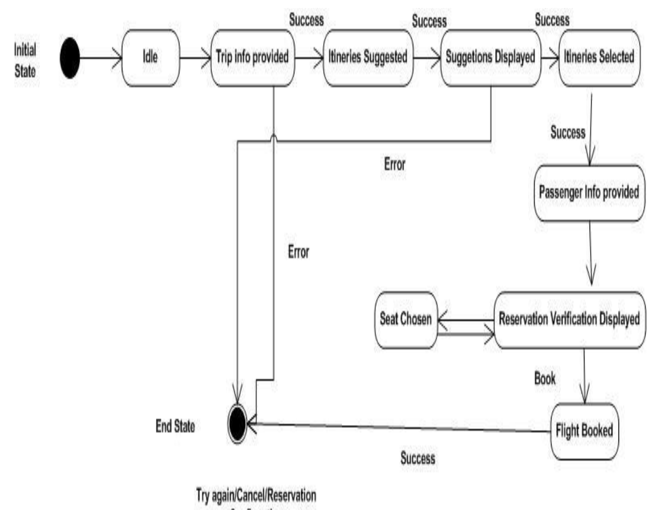

Fig. 7. State Model for airline reservation system

Figure 7 represent the state diagram gives an abstract description of the behavior of an airline reservation system.

C. Deployment diagram for Airlines Reservation System

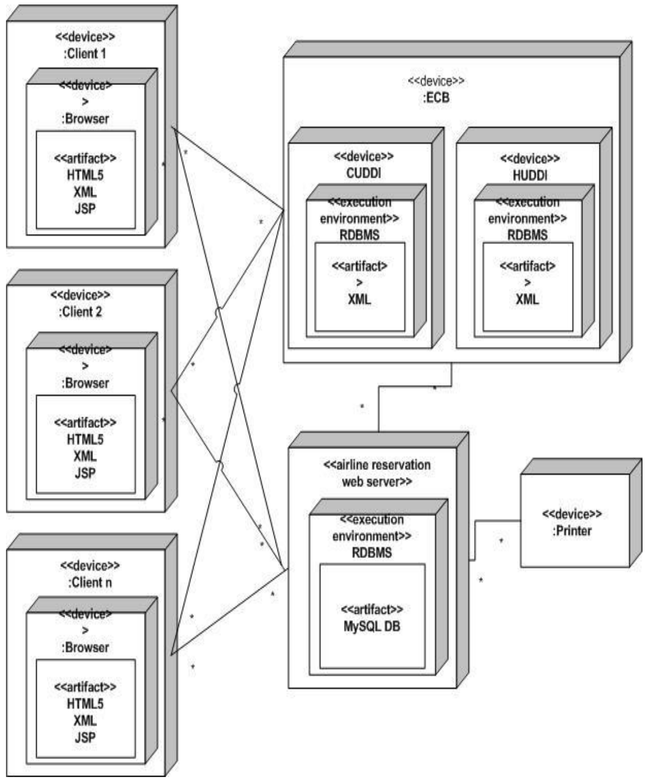

Fig. 8. Deployment Model for airline reservation system

Figure 8 shows the deployment model of the airlines reservation system. The airlines reservation system is deployed in the web server and the client access the software as service deployed web server. Finally the data is being accessed through the database server.

This UML modeling can be used by the users for service modeling and collaboration for airline reservation system and can be used for other service industries as well.

VII. Conclusion

This paper presents a service modeling and collaboration of an agent based Enterprise Cloud Bus framework using UML 2.0. The main motivation of this paper is to standardize the ECB framework in terms of service collaboration, virtualization and communication and to communicate specific service objects in model elements using UML 2.0, which provides an effective modeling method for agent based web service applications.

As our future work, we are planning to formalize our interpretation of UML 2.0 in an ECB framework, which will provide not only the semantic foundations but also the functional verification procedure of the diagrams.

References

[1] OMG, Unified Modeling Language: version 2.0 (UML 2.0), 2003, http://www.uml.org/#UML2.0.

[2] Bauer, B. and Müller, J.P., “Methodologies and Modeling Languages”, Book: Agent-Based Software Development, Luck M., Ashri R. D’Inverno M. (eds.) Artech House Publishers, Boston, London, 2004.

[3] Birgit B, “Models and methodology for agent oriented analysis and design”, Working Notes of the KI’96 Workshop on Agent-Oriented Programming and Distributed Systems, 1996.

[4] Booch G., Rumbaugh J., Jacobson I., The Unified Modeling Language User.

[5] Al_Kholy, M. S. A. R. Khalifa, M. G. Alsaied, “A Systematic Approach for Mobile Agent Design Based on UML (Class and Sequence Diagrams)”, Journal of American Science 6.12 (2010): 284-290.

[6] Garijo F. J., Boman M., “Multi-Agent SystemEngineering”. Proceedings of MAAMAW'99. Springer, ed., 1999.

[7] Herlea D. E., et al., “Specification of Behavioural Requirements within Compositional Multi-Agent System Design”, 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, pages 8-27, Springer, 1999.

[8] Mario K., Jezic G., “Modeling agent mobilitywith UML sequence diagram”, AgentLink III AOSE TFG2 (2005).

[9] M. Kang and K. Taguchi, “Modelling Mobile Agent Applications by Extended UML Activity Diagram”, 6th International Conference on Enterprise Information Systems (ICEIS)’ 04, Portugal, 2004.

[10] OMG,UML 2.0 superstructure Specification, http://www.omg.org/docs/ptc/03-08002.pdf ,2003.

[11] OMG, Unified Modeling Language Specification (version. 1.5), UML Revision Task Force, 2002.

[12] J.Odell, R. Levy, M-P. Huget and M. Nodine, “FIPA Modelling: Agent Class Superstructure Metamodel”, FIPAModelling TC, 2004.

[13] Mayer, P., Koch, N., Schroeder, A., & Knapp, A. The UML4SOA profile, 2010.

[14] Bahri M. R., Mokhtari R., Chaoui A., “Towards an extension of UML2. 0 to model mobile agent-based systems" International Journal of Computer Science and Network Security 9.10, 124-131, 2009.

[15] Saleem M. Q., Jaafar J., Hassan M. F., “Secure business process modeling of SOA applications using UML-SOA-SEC", International Journal of Innovative Computing, Information and Control, 8(2), 2729-2746, 2012.

[16] Chang, Peter H., “A Methodology of Migration to Cloud Computing Using UML Modeling” GSTF Journal on Computing, 1(3), 2011.

[17] Li R., Yang D., Hu H., Luo J., “Towards Description Logics Based Cloud Service for UML Verification”, Information Engineering and Applications, LNEE Springer London, 154, 1184-1191, 2012.

[18] R. K. Chatterjee, A. Sarkar, S. Bhattacharya, “Modeling and Analysis of Agent Oriented System: Petri-net Based Approach”, 11th Intl. Conf. on Software Engineering Research and Practice (SERP 11), Vol. 1, PP 17 – 23, 2011.

[19] Anirban Sarkar, Narayan C Debnath, “Measuring Complexity of Multi-Agent System Architecture”, 10th IEEE Conference on Industrial Informatics (INDIN 2012), PP 998 – 1003, 2012.

[20] Sarkar, Anirban. "Modeling multi-agent system dynamics: Graph semantic based approach." In Service Systems and Service Management (ICSSSM), 2013 10th International Conference on, pp. 664-669. IEEE, 2013.

[21] Neha, Rajib Chatterjee, Anirban Sarkar, “High Level Petri Net based Behavioral Model for Multi Agent System”, 3rd International Conference on Advanced Computing and Communication, Los Angeles, PP 25-30, September, 2013.

[22] Faisal Ahmed, Suvamoy Changder, Anirban Sarkar, “Web Service Execution Model for Cloud Environment”, ACM SIGSOFT Software Engineering Notes, Volume 38, Number 6, PP 1 – 13, November 2013.

[23] Faisal Ahmad, Anirban Sarkar, Narayan C Debnath, "Analysis of Dynamic Web Services", Accepted in IEEE International Conference on Computing, Management and Telecommunications (ComManTel 2014), Vietnam, April 27 – 29, 2014.

[24] Gitosree Khan, Sabnam Sengupta, Anirban Sarkar, Narayan C Debnath, “Modeling of Inter-Cloud Architecture using UML 2.0: Multi-Agent Abstraction based Approach”, In Proc. of 23rd International Conference

on Software Engineering and Data Engineering (SEDE), New Orleans, Louisiana, USA, pp 149-154, October 13-15, 2014.

[25] Gitosree Khan, Sabnam Sengupta, Anirban Sarkar, "WSRM: A Relational Model for Web Service Discovery in Enterprise Cloud Bus (ECB)", In Proc. of 3rd International Conference on Eco-friendly Computing and Communication Systems (ICECCS) , NITK Surathkal, Mangalore, December 18th-21st,2014.