Frameworks for m-business Agent Development with CBD

Haeng Kon, Kim

School of Information Technology, Catholic University of Daegu
KyungSan, Taegu, 712-702, Korea.

Abstract: Agent technology becomes more and more importance in the e-business domain. The concepts and technology have been brought to a stage where they are useable in real applications, and there is a growing understanding of how to apply them to practical problems. Component methodologies have proved to be successful in increasing speed to market of software development projects, lowering the development cost and providing better quality.

In this paper, we propose systemical development process using component and UML(Unified Modeling Language) technology to analysis, design and develop e-business agent. The mbA-CBD(m -business Agent-Component Based Development) process is an attempt to consider all of the best features of existing AOSE(Agent Oriented Software Engineering) methodologies while grounding agent-oriented concepts in the same underlying semantic framework used by UML, the standard Modeling language for Object Oriented Software Engineering. Finally we describe how these concepts may assist in increasing the efficiency and reusability in business application and e-business agent development.

Keywords : M-Business Agent(mbA), mbA-CBD Reference Architecture, mbA-Spec., CBD

1. Introduction

Recently the software lifecycle is getting shorter and web service for paradigm of next generation information technology is more focused on while e-business model has developed very rapidly. Therefore, the development of software which is more functionable, various, stable software, the key of business domain. According to these requirements, not only the component having exchangeable module that performs independent business and function in software system but also the utilization of agent in e-business domain become more noticeful. It is important to produce the agent service based on component technology that is change to replacement and portability toward developing the software having high productability[1][2].

In this paper, we propose mbA-CBD process. This proposed posses applies mbA model notation and role model, goal model, architecture model and interaction model in the view of agent. Simultaneously, these 4 models considered as mbA model. The mbA Model can define agent characteristics and the relations among agents. At the same time, component is possibly constructed on mbA specification. In addition the process are presented though of e business agent models the case study of component information search agent.

2. Related Works

2.1 Agent Concept Model

An agent is an atomic autonomous entity that is capable of performing some useful function. The functional capability is captured as the agent's services. A service is the knowledge level analogue of an object's operation. The quality of autonomy means that an agent's actions are not solely dictated by external events or interactions, but also by its own motivation. We capture this motivation in an attribute named purpose. The purpose will, for example, influence whether an agent agrees to a request to perform a

service and also the way it provides the service. Software Agent and Human Agent are specialization of agent[3].

Figure 1 gives an informal agent-centric overview of how these concepts are inter-related. The role concept allows the part played by an agent to be separated logically from the identity of the agent itself. The distinction between role and agent is analogous to that between interface and class: a role describes the external characteristics of an agent in a particular context. An agent may be capable of playing several roles, and multiple agents may be able to play the same role. Roles can also be used as indirect references to agents. This is useful in defining re-usable patterns. Resource is used to represent non-autonomous entities such as databases or external programs used by agents. Standard object-oriented concepts are adequate for modeling resources[4].

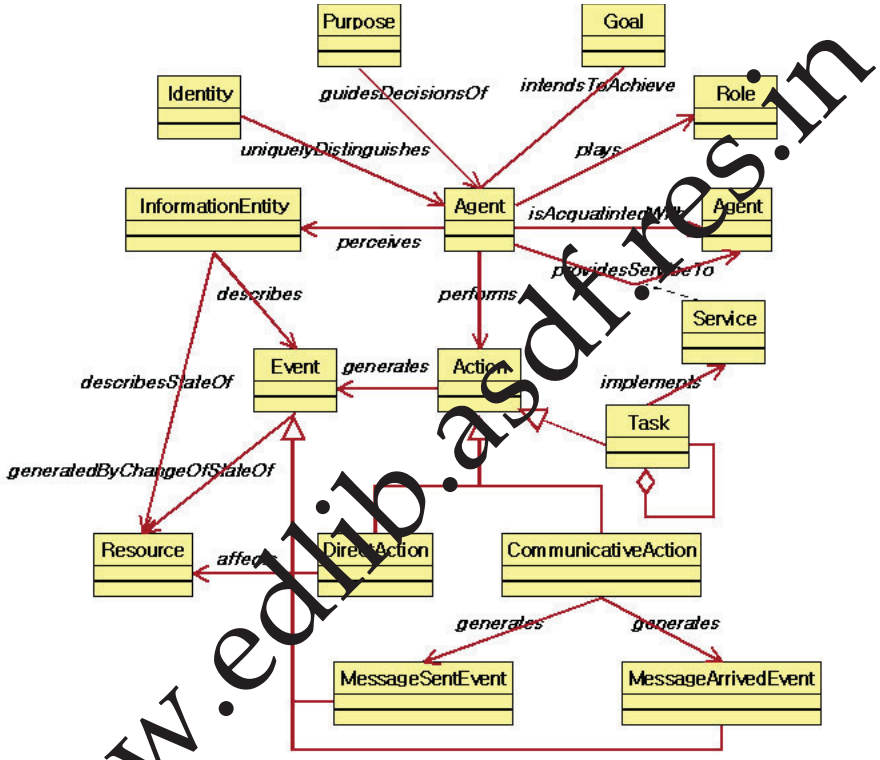

Figure 1. Agent Concept Model

2.2 Reference Architecture of mbA-CBD

In order to construct component reference architecture, agent is classified in general agent type and e-business function attribute. Figure 2 is a component and meta architecture of based on all above described for e-business agent[5][6].

Reference architecture is consisted of dimension, which has 14 general types and 11 concrete business agent types with domain oriented component architecture. These two classification areas tend to be independent for each cross-referenced. Each area has its own horizontal and vertical characteristics. General agent types are corresponding to agent platform and application. It is possible to develop agent system or application by the referencing architecture. The technology of agent can be applied to business domain.

Developed component is classified by the reference architecture and is placed according to general agent type and business attribute. In case agent is applied to the agent system or business domain, system is possibly to build up by identifying component related to business domain and combining it.

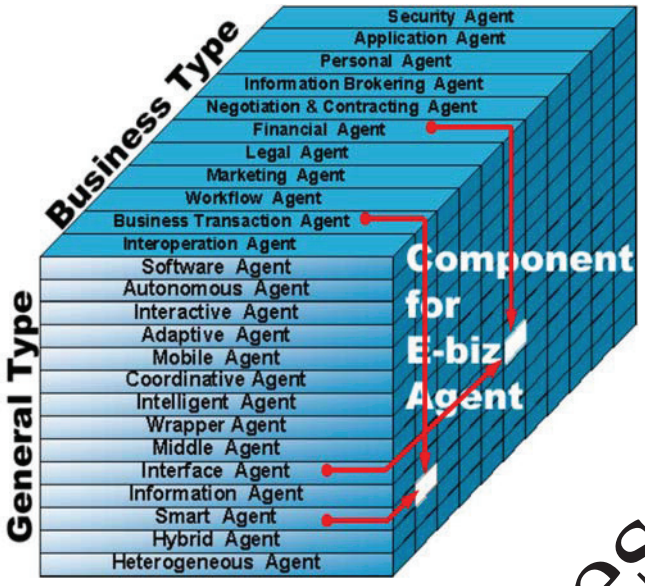

Figure 2. mbA-CBD Reference Architecture

3. mbA-CBD Process

As we suggested mbA-CBD reference architecture in previous research[5], component development process based architecture is a set of activities and associated results, which lead to the production of a component as shown in figure 3. These may involve the development of component from mbA specification by using UML model. Here, our main concern is the specification workflow.

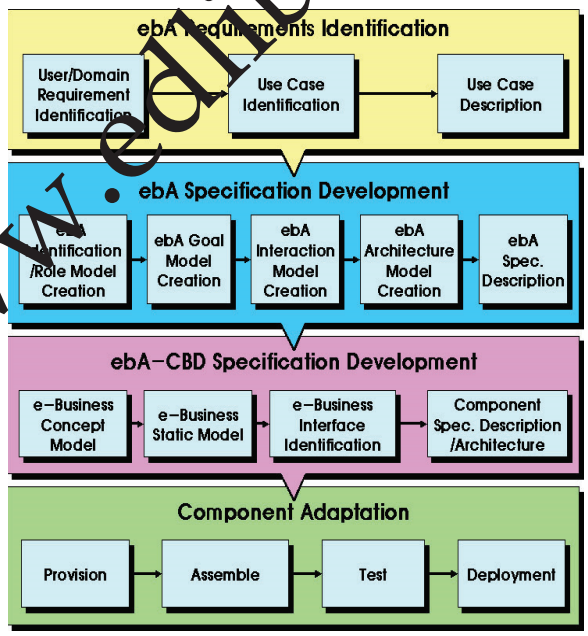

Figure 3. mbA-CBD process

In addition, we consider systemical development process using AUML and mbA model technology to analyze, design, and develop e-business agent. The domain analysis specification, design model, implemented component, which are produced though the process, are stored in the repository[6].

3.1 mbA Requirements Identification Phase

The requirement of agent should be first identified in desired business system. The primary property of agent is able to analyze after that the description for specific agent platform and the sorts of essential properties should be understood. At the same time, it is very important to consider weather the requirement, which is already defined, is corresponding to agent type in reference architecture and what business concept is focused on..

For the e-business domain analysis, UML approach is used. Diagrams used in problem domain analysis are use case diagram. Use case diagram is a diagram that shows a set of use cases and actors and their relationships. It supports the behavior of a system by modeling static aspects of a system.

Also, domain analysis is presented on entire domain concept and scenario using activity diagram. Requirement analysis is defined through use case diagram, and use case description.

3.2 mbA Specification Development

Agent specification based on user's requirement creates mbA Specification and 4 models. These products that are acquired though the specification phase, becomes the main resource to identify and development new components.

As mentioned, in order to provide a further degree of expressiveness, mbA Model extends the UML meta-model by adding a number of elements to it. This section describes the notation that mbA Model represents graphically the instances of these new meta-elements in the diagrams.

Figures 4 provide a summary of the symbols representing the mbA Model concepts and relations respectively. It is notice that symbols are associated to elements natively included in the UML meta-model, which doesn't mention here.

The usages of relationships are as follows:

• Implication: This relation links one or more elements that have an attribute of type state to a single element that has an attribute of type state.

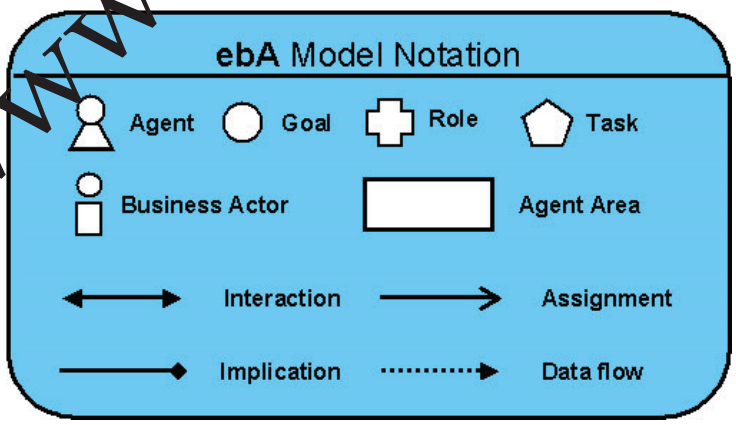

Figure 4. mbA Model Notation

• Assignment: This relation links an element of type Autonomous Entity to an element that has an attribute of type Autonomous Entity. The semantics are such that the assignment from one Autonomous Entity to another following the direction of the arrow.

• Data flow: This relation links a Data Prosumer to an Information Entity that is produced or consumed. This is the same relation as the Object Flow relation defined in UML and therefore the same symbol is used here.

3.2.1 mbA Identification and Role Model Creation

This focuses on the individual Agents and Roles. For each agent/role it uses scheme supported by diagrams to its characteristics such as what goals it is responsible for, what events it needs to sense, what resources it controls, what tasks it informs how to perform, 'behavior rules', etc.

Agent identification uses mbA-CBD reference metrics as figure 5. Agent naming is referenced from use case description and role model is created using mbA model notation. Also, mbA-CBD reference metrics used to give a classification code.

E-Business Agent / Agent Type	System-Level Agent(00)			General Business Activity Agent(10)					Personal Agent (20)	System -Level Agent (30)	Security Agent (40)
	Interoperation Agent (01)	Business Transaction Agent (02)	Workflow Agent (03)	Marketing Agent (11)	Legal Agent (12)	Financial Agent (13)	Negotiation /Contracting Agent (14)	Information Brokering Agent (15)			
Software Agent(SWA)											
Autonomous Agent (AUA)											
Interactive Agent (INA)											
Adaptive Agent (ADA)											
Mobile Agent (MBA)											
Coordinative Agent (COA)											
Intelligent Agent (ITA)											
Wrapper Agent (WRA)											
Middle Agent (MOA)											
Interface Agent (IFA)											
Information Agent (IMA)											
Smart Agent (SMA)											
Hybrid Agent (HYA)											
Heterogeneous Agent (HGA)											

Figure 5. mbA-CBD reference metrics

3.2.2 mbA Goal Model Creation

This shows Goals, Tasks, States and the dependencies among them. Goals and Tasks are both associated with States, so that they can be linked by logical dependencies to form graphs that show e.g. that achieving a set of sub-goals implies that a higher level Goal is achieved, and how Tasks can be performed to achieve Goals. Graphs showing temporal dependencies can also be drawn, and we have found UML Activity Diagram notation useful here.

3.2.3 mbA Interaction Model Creation

This model highlights which, why and when agents/roles need to communicate leaving all the details about how the communication takes place to the design process. The interaction model is typically refined through several iterations as long as new interactions are discovered. It can be conveniently expressed by means of a number of interaction diagrams. This model is interaction centric and shows the initiator, the responders, the motivator of an interaction plus other optional information such as the trigger condition and the information achieved and supplied by each participant.

3.2.4 mbA Architecture Model Creation

This model shows agents relationship to negotiate and coordinate in agent area. It considers the business actor and domain concept. An agent area is where software agents meet and interact in the target architecture. The agent areas can be distributed on different hosts, and facilitate means for efficient inter-agent communication.

There are two main types of agent area. One is the area where the agents advertise their capabilities, communicate with other agents. The other is the user-client where the user interacts with agents.

3.2.5 mbA Specification Description

The mbA specification description is based previous models as role model, goal model, interaction model and architecture model. mbA specification is shown functional and non-functional elements in figure 6. The functional elements are described to use class diagram and sequence diagram.

3.3 mbA-CBD Specification Development

We have attempted summarize the process tasks into the four stages: e-business concept model, e-business static model, e-business interface identification and component spec description.

Item	Description
Agent Name	Identified ebA name
E-Business Type	Identified E-business Type in ebA-CBD reference architecture
General Agent Type	Identified General Agent Type in ebA-CBD reference architecture
Identification Code	Identified code in ebA-CBD metrics
Access Information	Assessed, stored and modified information
Produce Information	File/information which the agent creates while it operates
Related Agent	Agent which negotiate, coordinate or exchange message
Information Model	Represented agent's attribute and operation
Operation Model	Represented sequence of operation among agents

Figure 6. mbA Specification

The specification development takes as its input from requirements a use case model, mbA models and a mbA-spec. It also uses information about existing software assets, such as legacy systems, packages, and databases, and technical constrains, such as use of particular architectures or tools. It generates a set of component specifications and a component architecture. The component specifications include the interface specifications they support or depend on, and the component architecture shows how the components interact with each other.

The identified information based on component users and performance must be provided in specification form for integration. Also, this information can be provided and acquired by producer, consumer and agent in interoperating system. The information of component design and development, and also functional and non-functional information must be provided by producer, and agent must provide the commercial information with this. This information is the important standard for choice and the ground for reuse to acquire the component. Figure 7 shows this information.

Item	Description
Category	Component family of Business domain
Component Diagram	Relationship between component
Component Name	Identified component name
Classification Code	Classification code of component based on ABCD Architecture
Short Description	Describe about component function, motive, constraint, etc.
Glossary	Describe concept of glossary related component specification
Component Context Diagram	Main function of Component
Component Interaction Diagram	Relationship between component
Component Sequence Diagram	Operational sequence of component
Component Diagram	Represent of required and provide interface
Component State Diagram	Represent of operation change
Interface Description	Pre/Post condition, input/output result
Usage Scenario	Scenario for component usage
Quality Attribute	Non-functional(Quality) attribute

Figure 7. Component specification

3.4 Component Adaptation

These output are used in the provisioning phase to determine what components to build or buy, in the assembly phase as an input to test scripts.

The provisioning phase ensures that the necessary components are made available, either by building them form scratch, buying them from a third party, or reusing, integrating, mining, or otherwise modifying an existing component or other software. The provisioning phase also includes unit testing the component prior to assembly.

The assembly phase takes all the components and puts them together with existing software assets and a suitable user interface to form an application that meets the business need. The application is passed to the test phase for system and user acceptance testing.

4. Example of mbA-Specification

In this thesis, we focused on mbA specification so, case study only mentioned mbA requirements identification and specification development. The example is component information search agent. The agent finds the information of the component to be registered newly.

4.1 mbA Requirements Identification for Component Information Search Agent

The requirement identification simply presented by use case diagram as figure 8. The actors are user and agent that is agent family.

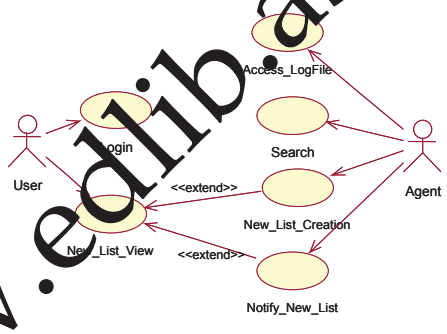

Figure 8. Use Case Diagram of Component Information Search Agent

4.2 mbA-Spec. Development of Component Information Search Agent

The mbA candidate identify based on use case diagram and using mbA-CBD reference metrics. Figure 9 presented identified mbA as user, search and collection agent.

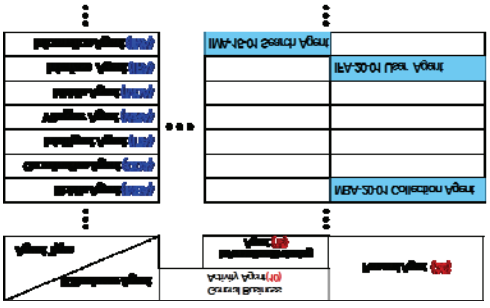

Figure 9. mbA-CBD Metrics of Component Information Search Agent

Figure 10 represents overall role of component information search agent. The roles describe the external characteristics of identified agent. Also, The goal model in Figure 11 shows the main goal of the component information search agent. The CteateNewList goal is achieved when lower goal successfully completed.

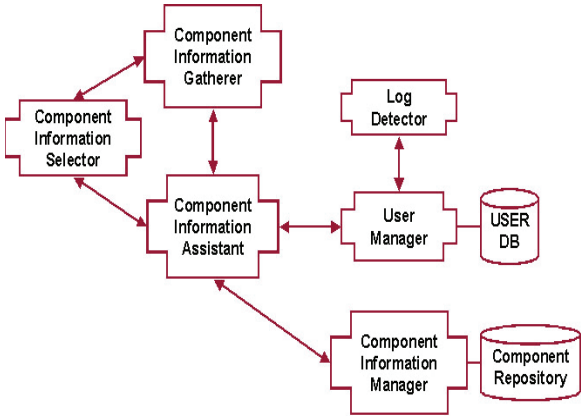

Figure 10. Role Model of Component Information Search Agent

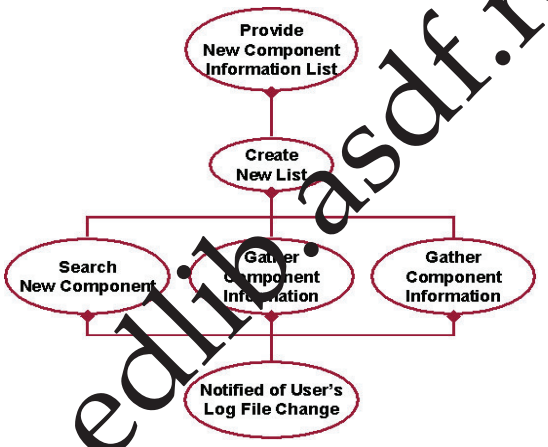

Figure 11. Goal Model of Component Information Search Agent

The following picture shows as an example the interaction model describing the Information Request interaction between the Component Information Gather and Component Information Assistant roles. Figure 13 shows an architecture model. It represents overall system structure of Component Information Search Agent. Search Agent finds the information of the component to be registered newly, Collection Agent periodically update and gather to component information list and User Agent provide to alert service and manage log file.

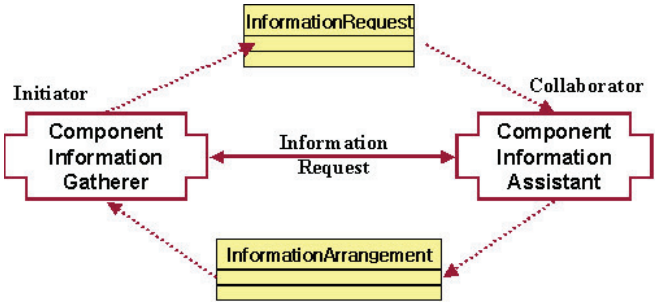

Figure 12. Interaction Model of Component Information Search Agent

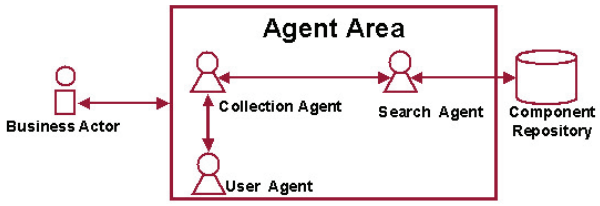

Figure 13. Architecture Model of Component Information Search Agent

Figure 14 present User Agent specifications consists of mbA resource information and two models. The resource information is basic elements such as name, e-business type and general agent type according to mbA-CBD reference architecture and classification code. Information and operation model provide inter/external structure and interoperation of agents. These referred mbA-CBD specification development phase.

Acknowledgement

Following are results of a study on the "Leades INdustry-university Cooperation" Project, supported by the Ministry of Education, Science & Technology (MEST) 2013

5. Conclusion

In this paper, each characteristic in the view of agent and component are defined and mbA-CBD process is proposed to develop e-business agent. Defining mbA specification of whole entire agent has e-business agent information more systemical and intuitional then also includes more information. Introducing the systemical process and mbA-CBD reference model of e-business agent can provide the efficiency by the component easily. Also, the specification can be the guideline to choose desired component and be reused as based more for component creation.

For the further works, the definition and detail methods should be required based on mbA specification for component development and assemble. Furthermore, the comparison and verification are needed though the cases study of implementation.

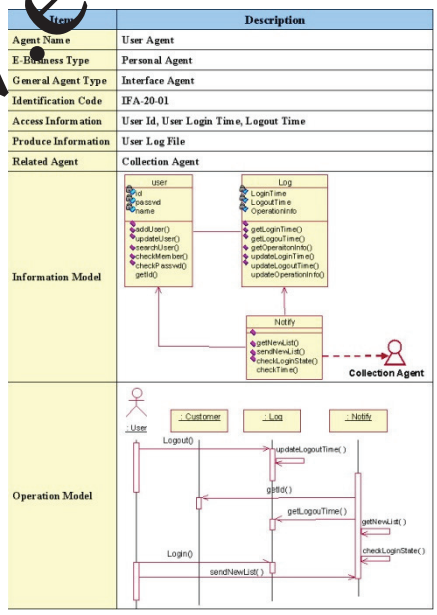

Figure 14. mbA-Spec. of User Agent

References

1. Martin L. Griss, Gilda Pour, "Accelerating Development with Agent Components," IEEE Computer, pp. 37-43, May. 2001.
2. Hideki Hara, Shigeru Fujita, Kenji Sugawara, "Reusable Software Components based on an Agent Model," 7th International Conference on Parallel and Distributed Systems Workshops, 2000.
3. OMG Agent Platform SIG, "Agent Technology Green Paper," http://www.objs.com/agent/, 2000.
4. EURESOMP, "MESSAGE: Methodology for Engineering Systems of Software Agents," EURESCOMP Project P907 Publication, 2000.
5. Ho-Jun Shin, Haeng-Kon Kim, "CBD Reference Architecture through E-Business Agent Classification," Proceedings of 2nd International Conference on Computer and Information Science, pp. 653-658, Aug. 2002.
6. H.K. Kim, "Component Repository and Configuration Management System", ETRI Final Research Report, 2000.
7. H.K. Kim, "Component Repository and Configuration Management System", ETRI Final Research Report, 2000.
8. H.K. Kim, E.J.Han, H.J. Shin and C.H. Kim, "Component Classification for CBD Repository Construction", Proceeding of SNPD'00, pp. 483-493, 2000.
9. Martin L. Griss, "Agent-Mediated E-Commerce Agents, Components, Services, Workflow, UML, Java, XML and Games...," Proceedings of the Technology of Object-Oriented Languages and System, Keynote Presentation, 2000.
10. Hyacinth S. Nwana, "Software Agents: An Overview," Knowledge Engineering Review, VOl. 11, No 3, pp. 1-40, 1996.
11. Nicholas R. Jennings, Michael Wooldridge, "Agent-Oriented Software Engineering," Proceeding of IEA/AIE 1999, pp. 4-10, 1999.
12. George T. Heineman, William T. Councill, Component-Based Software Engineering, Addison-Wesley, 2001.
13. Seoyoung Park, Chisu Wu, "Intelligent Search Agent for Software Components," Proceedings Sixth Asia Pacific Software Engineering Conference, pp.154 -161, 1999.
14. Klement J. Fellner, Klaus Turowski, "Classification Framework for Business Components," Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, pp. 3239-3248, 2000.
15. Nicholas R. Jennings, Michael Wooldridge, "Agent-Oriented Software Engineering", Proceeding of IEA/AIE 1999, pp 4-10, 1999.
16. Odell, James ed., "Agent Technology", OMG, green paper produced by the OMG Agent Working Group, 2000.
17. Mike P. Papazoglou, "Agent-Oriented Technology in support of E-Business", Communications of the ACM, Vol. 44, No. 4, pp71-77, 2001.
18. James Odell, H. Van Dyke Parunak and Bernhard Bauer, "Extending UML for Agents", Proceeding Of the Agent-Oriented Information Systems Workshop at the 17th National Conference on Artificial Intelligence, 2000.
19. Bernhard Bauer, Jörg P. Müller and James Odell, "Agent UML: A Formalism for Specifying Multiagent Interaction", Proceeding of 2000 Agent-Oriented Software Engineering, pp. 91-103, 2001.
20. Pearl Brereton, David Budgen, "Component-Based Systems:A Classification of Issues", IEEE Computer, Vol 33, No 11, 2000.
21. Yariv Aridor, Danny B. Lange, "Agent Design Patterns : Elements of Agent Application Design", Proceeding of Autonomous Agents 98, pp. 108- 115, 1998.
22. Peter Herzum, Oliver Sims, Business Component Factory, OMG press, Dec. 1999.