Fault-Tolerant Adaptive Mobile Agent System using Dynamic Role based Access Control

P.Marikkannu
Department of Information Technology
Anna University of Technology
Coimbatore

J.J. Adri Jovin
Department of Information Technology
Anna University of Technology
Coimbatore

T.Purusothaman
Department of CSE& IT
Government College of Technology
Coimbatore

ABSTRACT
With the advancement in the Mobile Agent Technology, the use of Mobile Agents has extended to a lot of applications, where a coordinated approach is required to achieve certain tasks. In a co-ordinated approach, the roles that are being held by a Mobile Agent play a vital role. There are chances that a mobile agent fails while performing an operation inside a system, due to various factors. The system must be tolerant enough to encounter such situations. This paper introduces a special category of Mobile Agent named Adaptive Mobile Agent, which is designed to accept additional roles, while working inside an environment. This paper also introduces the eXtended Volunteer Algorithm, which performs role assignment in an efficient manner, reducing the overload due to additional role-assignment. This helps the agent to accept roles under dynamic conditions, based on the rules in the agent environment which would help to achieve a fault-tolerant system. The experimental results obtained with the usage of eXtended Volunteer Algorithm over the Adaptive Mobile Agents prove to be more efficient in fault-tolerance.

General Terms
Security, Algorithm, Faut-tolerance.

Keywords
Adaptive Mobile Agent, Dynamic Role Based Access Control, Fault-tolerant systems.

1. INTRODUCTION
Mobile Agents have started their reign in various distributed cooperative tasks. Mobile Agents, as a team are capable of accomplishing various tasks. While working as a team, each agent may be may be assigned some roles which is related to a work that is about to be handled by it. Based on the role, the mobile agent performs some actions and may gain access to some resources from the environment. Consider a circumstance, when one or more mobile agents fail due to some malicious attacks. At such a circumstance, in order to achieve a fault-tolerant mechanism, some mobile agents may have to share or take the roles of the failed mobile agent to accomplish the task. However, this is not possible under an ordinary environment. Hence, we device a context-aware environment which would perform the task of sharing and allocating the roles to the mobile agents present in the environment. The Mobile Agents are also designed to acquire the roles allotted by the environment. The environment generates the rules based on circumstances and the mobile agents acquire the roles based on the instructions given by the environment.

Based on the roles that are acquired by the Mobile Agents, the agents are given access to various resources present in the environment to accomplish a task. The task of achieving this is not a simple one. The Mobile Agents may have been generated from various environments or from some other platform or system. Irrespective of the origin, the mobile agents must cooperate with one another and cooperate with the environment to acquire roles and accomplish various tasks. For this purpose, we have designed a special type of mobile agent named the 'Adaptive Mobile Agent' and their performance and activities with regards to the external environment are discussed in this paper. Also, the technique of dynamic rule generation by the agent environment and dynamic role acquisition by the Mobile Agent are discussed in this paper.

2. BACKGROUND

2.1 Mobile Agent
Agents are software modules which perform an activity on behalf of a user. A Mobile Agent is a special kind of software which possesses the ability of movement. A Mobile Agent is motile and can migrate from one system to another in order to perform a particular job. The mobility of the mobile agent makes it suitable to perform various tasks which require data from varied and diversified locations. The Mobile Agent may be a single module or a program as a whole. The mobile agents can successfully complete any task assigned to it by interacting with other agents present in the environment. The mobile agents are capable of carrying enormous volumes of data along with them while traveling through a network. The various advantages of using a mobile agent in a network are:

- Lowered network traffic
- Autonomous behavior
- Shortened network delay

A Mobile Agent which is present as a team may possess even more functionalities such as

- Synchronous Decision –Making
- Co-operative task accomplishment
- Mutual Understanding

Apart from the above said properties, the mobile agent is much used because of its safe data transfer ability and mobility. Thus a mobile agent can be used to perform various activities.

With reference to a team action, a Mobile Agent can acquire roles based on the task and nature of the work that need to be done. This flexible property of the mobile agent is used for the construction of a new classification of mobile agent named 'Adaptive Mobile Agent' which could acquire dynamic roles under dynamic environment.

2.2 Role Based Access Control

Access Control is one of the most important issues in security. Access control refers to the process of controlling the access of a user towards a resource that is present in a system. Access Control is critical to preserving the confidentiality and integrity of information [16]. Roles are being assigned to users to restrict or grant access towards a resource. This mode of restricting or granting access to a resource is called Role Based Access Control (RBAC) which plays a main role in managing security of data inside a system. In most systems, an active directory which contains a Access Control List (ACL) is maintained which contains information about various permissions that are granted to an user with respect to a resource. Only users who possess their entry in the Access Control List are allowed to access resources inside a system. This helps to maintain the authentic ability of a user, with respect to that system. In an Access Control List, a username is usually associated with a password which helps the system to overcome various security issues.

Roles can be either allocated statically or dynamically. The static or dynamic assignment of roles depends on the way in which the system is designed. Static assignment is one in which the roles does not change throughout, in the system. Dynamic assignment is one in which the roles change with the situation or as per the internal logic of the system. The dynamic assignment of roles can be used for various tasks which involve teamwork.

2.3 Dynamic Rules

The Dynamic rules may be defined as the set of conditions that are generated as per the context of the environment. Based on the dynamic rules certain actions may be performed. The generations of dynamic rules follow the Event-Condition-Action Model. When an event occurs and a condition is being satisfied in the rules library, an action may be triggered which results in the assignment of roles in our system.

3. RELATED WORKS

The inspiration towards the dynamic rule generation for dynamic roles is obtained from various issues related to the field of robotics, where we would find a team of robots cooperating and coordinating with one another to accomplish a task. The various works related to our work are listed below.

[1] describes a procedure of dynamic role assignment based on the various scenarios. It suggests using a Role Oriented Programming Environment for the role assignment tasks in which each role can be assigned based on the scenario in an office environment. Odell et al. describes the role assignment based on six scenarios which are described below. Classify, which adds a role of manager to role of employee as a result of promotion. Declassify, which removes the role of manager from an employee as a result of demotion. Reclassify, which changes an employee to an unemployed condition. Activate, which take up behaviors of manager role as a part of day to day business activities. Deactivate, which stops manager behavior and take on role of employee alone. Shift, in which a employee is changed from a role relevant to the office to a role irrelevant to the office. Also, three states namely Active, Suspended and Occupied are used. Occupied refers to the state of an entity which has become instance of a particular role. Suspended refers to a state in which the entity is occupied but perform no action. Active refers to a state in which the entity is occupied and performs some action.

[2] discusses the dynamic role assignment for robots in a cooperative environment. The role is performed by a agent when certain constraints are satisfied both externally and internally. The role depends on the internal state of the robot and the information from the environment and other robots. Here, three types of role assignment are produced which are allocation, reallocation and exchange which are almost similar to the action types in [1].

In case of [3], Xu and Xia gives emphasis to the concept of situation calculus for assigning roles. The situation or the circumstance is taken as an input and the roles are assigned based on the input that is given. An assumption that an agent can predict the actions of other agents is made in this work, making the environment suitable for the action of situation calculus over it. But, in real time environment, this is not suitable. An agent may have its own autonomous behavior. Unless controlled by some external force, agents may not be able to know the situation prevailing in the environment.

Costa et al., in their work in [4] describe the context aware services that are being adapted in certain services. The work is based on the ECA model where the events are the occurrences of the interesting events, conditions specify the conditions that must hold prior to the execution and actions are the invocation of arbitrary services. In order to execute service specific behaviors, the controller component in the system observes event notifications asynchronously, monitor condition rules and triggers actions when a particular event occurs and condition satisfied. Also Costa et al. describe three kinds of events namely primitive, temporal and situation events of the three primitive and temporal are not context aware whereas the situation events are context aware and the context-aware services use them for their functionalities.

Dersingh et. al. presents an hierarchy based role assignment in [5]. The contextual terms are represented as ontology by means of OWL. The OWL reasoning engine provides reasoning about class membership, classification and consistency. Based on all the above, the access management is done. In access management, a policy editor which is capable of accessing domain ontology to acquire domain vocabularies is used. Any change in the environment is updated in the semantic knowledge base. The introduction of Extensible Access Control Markup Language makes the task so easier to be performed.

In [6], Ou et. al. introduces a scheme to allocate, manage and enforce authorization policies in a flexible manner. This adopts x.509 attribute certificate based RBAC for mobile agents and agent environment. Ou et. al. discuss only the methodology of security the RBAC and does not concentrate on dynamic roles

which could be stated a disadvantage of the system. [7] proposes a theoretical work of dynamic role assignment in a pervasive environment using agents which focuses on the assignment of a new role by means of checking vacant roles from the roles repository. This system may often go wrong because of the assignment of an irrelevant role to an agent which may be of no use to the system.

Zhang and Yin, in their work in [8] proposes a model for agent teams in which an agent when fails, the other agents reorganize to accomplish a task has been designed. A Multi-agent Logic Language for Encoding Teamwork (MALLET) has been formulated to achieve their task in a successful manner. In this work, hierarchical forms of plans have been designed to define the processes. Plan-Specific role specification is being followed, which at times may lead to failure in the completion of the job when the controlling body fails. Various rule assignments are also discussed and an algorithm named volunteer algorithm has been proposed. The disadvantages in the volunteer algorithm have been overcome in our work by the use of extended volunteer algorithm.

In [9], Sohr et. al. provides various methods for analyzing and managing role-based access control policies. They use an approach of separation of duty by which roles are being separated from one another. The various types of roles and efficient methods of managing the roles have been discussed. [10] proposes a model based on attribute-based user role assignment. The users are associated with appropriate roles based on factors such as their responsibilities and qualification. Some assumptions regarding the authentication, role-permission assignment and the number of users need to be satisfied for this system to work efficiently.

In[11] Wang et. al proposes a certain method of role assignment based on minority game strategies which is usually used in real time environment such as stock exchange. This work does not give details on how work is being assigned to agents. [12] discuss the efficient methods of using temporal RBAC, [13] gives an extension security architecture for sharing information using role-based delegation. The event –driven rules for sensing and responding to business situation using agents is done in [14], which uses event-triggers for processing the services. [15] shows an implementation of rule based agents in the semantic web.

In our work, we obtain some of the base ideas from the field of robotics to achieve a flawless system.

4. ARCHITECTURE

The architecture of the Dynamic rule-role based access control system in an agent environment is shown in Fig 1.

4.1 Context Aware Environment

The context-aware environment is the most important component of the system. The environment is aware of all the activities that take place in the environment. It contains the objects or the resources which need to be accessed by the mobile agent. Also, it has a heart-beat relationship with the Mobile Agents which are present inside it. The context-aware environment has a synchronized any-time communication with the dynamic rule generator to which it intimates all the activities or events that occur in the system. In addition to the above said functionalities, the context aware environment has the capability of creating new mobile agent with a new identity.

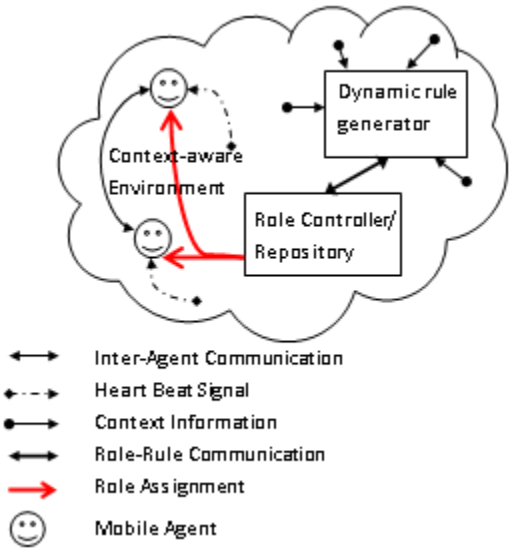

Fig 1: Architecture of Dynamic Rule-RBAC system

This is usually done when a mobile agent which is performing a task fails due to some unavoidable conditions. Once an Adaptive Mobile Agent enters the environment, the environment registers all the characteristics of the agent. When a mobile agent is being created, the intimation is made to the foreign platform, to which the failed mobile agent belonged about the creation. This would help the foreign platform to know about the failure of the mobile agent and would accept the newly created mobile agent. This process is essential when the failed agent requires carrying a vital output of the process that is done within the host environment.

4.2 Dynamic Rule Generator

Dynamic Rule Generator is a module that is responsible for the generation of rules in the system. It is based these rules the Rule Controller acts. The Dynamic Rule Generator may be an external module too, but must be synchronized with the context-aware environment. Dynamic Rule Generators which are used in business solutions like Jafa can be used by making proper customization making it suitable for the application which we are working with. Any strategy could be followed for rule generation.

4.3 Role Controller and Repository

The Role Controller is one which assigns the roles to the Mobile Agents in the system. The roles are assigned based on the rules generated by the Dynamic Rule Generator. The roles which are already available in the role repository can only be assigned to the Mobile Agents. The role assignment is based on the eXtended Volunteer algorithm which is discussed in Section 6. The Role Repository contains all the roles necessary for the agent role assignment and their definitions.

4.4 Adaptive Mobile Agent

In our work, we introduce a new class of Mobile Agent namely Adaptive Mobile Agent which is capable of acquiring the roles based on the instructions from the rule controller. We introduce

a threshold to avoid conditions like Role-overload which would reduce the efficiency of the system as well the efficiency of the Mobile Agent. An Adaptive Mobile Agent has two types of roles present in it. They are

1. Basic Role
2. Adaptive Role

The Basic role is the inbuilt and inherent role that is assigned to the Mobile Agent at the time of its origin. The Adaptive Role is the additional role which is being acquired by the Mobile Agent from the Role Controller.

The Mobile Agent can acquire the roles up to the role-threshold which may be a set by the administrator dynamically or is set by the programmer statically. A Mobile Agent can have a certain number of roles and cannot hold any role beyond that. Based on the role-threshold and weightage of the role, the roles are assigned to the Adaptive Mobile Agents.

Apart from the adaptive Mobile Agents, new mobile agents which are capable of carrying data are generated by the Context-aware Environment

5. COMMUNICATION MESSAGES

The architecture which we have proposed in this paper contains 4 types of messages which are discussed below.

5.1 Heart Beat Signal

This is the interaction signal between the Adaptive Mobile Agent and the Context-aware Environment. It is a message that is exchanged at regular intervals of time. Once the heart-beat signal is not received by the context – aware environment, it is assumed that the mobile agent is dead and is immediately informed to the Dynamic Rule Generator.

5.2 Context Information

This is the message that is being send to the dynamic rule generator each time a change is observed in its environment by the Context-Aware Environment.

5.3 Inter-Agent Communication

It is the message by which a Mobile Agent interacts with another agent which is present in the environment. This is usually needed to coordinate various operations among the mobile agents.

An example for Inter-agent communication is shown in Fig. 2 which involves a communication between the Grade Evaluation Agent whose task is to assign the grade and the Assessment Agent which has the capability of carrying the various marks based on the assessment of the student. We use the standard FIPA-request protocol to perform the communication [17]. The expected messages which would be exchanged are also shown. The "action" Grade Evaluation Agent requests stud-detail to perform the Grade Evaluation Process. The various parameters which are being used for communication are shown in Fig 2 and Fig 3.

```
(request
: sender(agent-identifier: name GradeEvalAgent)
: receiver(agent-identifier:name AssessmentAgent)
: language FIPA-SL
: ontology(studentmark-ontology)
: protocol(fipa-request)
: content
((action(agent-identifier:AssessmentAgent)
(stud-detail (regno)(college)(dept)(marks))
)) )
```

Fig 2: The message sent by Grade Evaluation agent for requesting agent Role Based Agent to retrieve the student detail information.

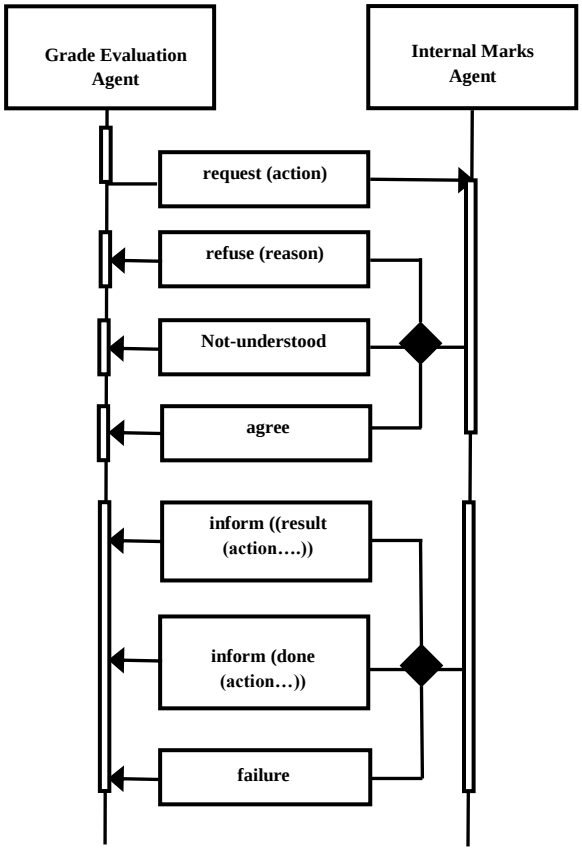

Fig 3: The standard FIPA request interaction protocol

5.4 Role Assignment Message

It is a message which is send by the Role Controller to the Adaptive Mobile Agents in the environment. This is a duplex mode of communication which comprises of the conversation between the Mobile Agent and the Role Controller. The response from the Mobile Agent is usually of Boolean type. TRUE or 1 corresponds to acceptance and FALSE or 0 corresponds to rejection of a request from the Adaptive Mobile Agent.

6. EXTENDED VOLUNTEER ALGORITHM (XVA)

The Volunteer algorithm in [8] focuses on the mobile agents and not the complete scenario. According to the Volunteer Algorithm any agent that is free could acquire a role from the role repository and perform the activities relevant to a role. This may promote the role assignment for mobile agents which may not be authenticated with the platform. This may result in security issues.

In order to overcome these issues, we introduce the eXtended Volunteer Algorithm to assign roles to the Adaptive Mobile Agents. Some of the common terms being used in the algorithm are

role_threshold: This is the maximum number of roles that could be held by the Adaptive Mobile Agent.
threshold_weight: The maximum weightage of roles that could be accepted by an Adaptive Mobile Agent.

The above two factors are usually determined either by the administrator or the programmer based on the nature of the task.

weight_held: This is the actual weight held by the Adaptive Mobile Agent, usually less than the threshold_weight
role_held: This is the number of roles held by the Adaptive Mobile Agent.
role_index: This is the product of weight_held and role_held

i.e. role_index=weight_hold × role_held

The eXtended Volunteer Algorithm applies only for the Adaptive Mobile Agents that are authenticated with the Environment.

Assume a mobile agent fails. As a result, the Context-Aware Environment intimates the Dynamic Rule Generator which communicates with the Role Controller. Now, the Role Controller uses the eXtended Volunteer Algorithm based on the roles present in the Role Repository. According to the algorithm, For a role,
1. The role controller sends a ready_to_receive message to all the Adaptive Mobile Agents in the Environment.
2. The adaptive Mobile Agents respond with response_to_receive message to the Role Controller within a valid time interval t. For an Adaptive Mobile Agent,
    a. If response_to_receive=0, then do not assign role to the agent
    b. else if response_to_receive= 1, then perform step 3
3. For an Adaptive Mobile Agent,
    a. if role_held < role_threshold and weight_held < threshold_weight, then calculate role_index
    b. else do not assign role to the agent
4. Sort role_index of all Adaptive Mobile Agents in ascending order.
5. Send accept_role_request message to the Adaptive Mobile Agent having lowest role_index
6. Receive accept_role_response message
    a. if accept_role_response=1, then assign role to the Adaptive Mobile Agent
    b. else if accept_role_response=0, then discard corresponding Adaptive Mobile Agent role_index from sorted role_index list and perform step 5.

The algorithm is valid inside the Context-Aware Environment. The Adaptive roles are discarded once the mobile agent gets out of the environment. So the Mobile Agent, when it moves out of the environment can carry only the data that is concerned with its basic role. The result that needs to be carried by the mobile agent that has failed will be provided to the mobile agent that has been newly created by the Context-aware environment.

7. EXPERIMENTAL RESULTS

Experimentation with two sections, each with 20 Intel Pentium IV PCs running with Windows XP platform is used. The mobile agents are created using Aglets 2.0.1[18]. The Adaptive mobile agents are designed for this purpose by overriding some specific definitions in the tool. Out of the two sections, one section is made to utilize ordinary agents whereas the other is made to use Adaptive Mobile Agents. Both the sections work for the same task and contain 40 mobile agents each. The task given to the mobile agents is to collect the marks from various systems, process them and provide a result whether a student is pass or fail. Oracle9i is used as the database to store the marks of students. The Ordinary Mobile Agent System creates a new mobile agent once if a mobile agent fails, from the host system where the failed mobile agent had originated. The experiment is performed for three continuous hours and analysis made. Similar type of malicious agents is introduced at the same time into each section to simulate agent failure. The response level is keenly noticed in each system. The overall processing capability of Adaptive Mobile Agent System (AMAS) with XVA implemented over Ordinary Mobile Agent System (OMAS) is illustrated in Fig 4.

Fig 4: Processing Capability Graph

From Fig 4, it could be observed that the Adaptive Mobile Agent System with XVA implemented responds much faster than the Ordinary Mobile Agent System. This is because, the AMAS uses the mobile agents inside the system itself to process the results even if a mobile agent fails. But the same thing is not applicable in case of OMAS. The OMAS environment, when it detects a mobile agent failure, intimates the system that had already created the mobile agent to create a new agent to process

the data to interpret the result. This adds a burden to the network. In AMAS, the system containing the mobile agent just shares the work with the other mobile agents inside the system itself. Since, the mobile agents are already inside the system, it does not require any sort of external communication. As a result, the time to create and dispatch a new mobile agent is saved and the response time becomes less. This in turn enhances the fault-tolerant capability of the system as a whole, thereby making the system much efficient compared to the other.

8. ADVANTAGES

Since the system allows only mobile agents that are authenticated with the Context-aware Environment, the tasks are performed in a secure manner. Also, the mobile agent when it leaves the Environment can contain only its basic role and can carry only data that are relevant or associated with the role. As a result, the users may not receive data beyond or below their role. This system is fault-tolerant and the tolerance is obtained by the participating mobile agents itself without the help of any other new foreign agents. The results obtained are carried to the required system without fail even if the original mobile agent responsible for the task fails.

9. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a fault-tolerant system based on dynamic role assignment which proves to be more secure than the traditional dynamic role assignment systems. Moreover, the eXtended Volunteer Algorithm proves to be more efficient in role assignment. The new type of Mobile Agent namely Adaptive Mobile Agent acts more active in acquiring roles due to its special design. Moreover, the system maintains the degree of authenticity and resource access control by controlling the carriage of resultant data from the system to external hosts.

In future, the system can be extended to support Time Based Access Control and Domain Based Access Control.

10. REFERENCES

[1] James Odell et al. 2003. Changing Roles: Dynamic Role Assignment. Journal of Object Technology, vol.2, no.5, September-October, 77-86.

[2] Luiz Chaimowicz et al. 2002. Dynamic Role Assignment for Cooperative Robots. In Proceedings of 2002 IEEE International Conference on Robotics and Automation, 293-298.

[3] Desheng Xu, Kejain Xia. 2009. Role Assignment, Non-Communicative Multi-agent Coordination in Dynamic Environment based on Situation Calculus. In Proceedings of Global Congress on Intelligent Systems, 89-93.

[4] Patricia Dockhorn Costa et al. 2008. Evaluation of a Rule-Based Approach for Context-Aware Services. In IEEE "GLOBECOM" 2008 proceedings.

[5] Anand Dersingh et al. 2009. Dynamic Role Assignment using Semantic Contexts. In Proceedings of 2009 International Conference on Advanced Information Networking and Applications Workshop. 1049-1054.

[6] Chung-Ming Ou, Yao-Tien Wang, C.R.Ou. 2008. Dynamic Role Assignment based on X.509 PMI Mechanism for Mobile Agent Systems.In Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming, 3699-3703.

[7] Akio Sashima, Noriak, Izumi, Yoshiyuki Kotani. 2006. Dynamic Role Assignment Protocol for Pervasive Computing. In Proceedings of the Third International Conference on Information Technology, New Generations (ITNG '06).

[8] Yu Zhang, Jianwen Yin. 2006. A Role-Based Modeling for Agent Teams. In Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and its applications (DIS '06).

[9] Karsten Sohr et al. 2008. Analyzing and Managing Role-Based Access Control Policies. IEEE transactions on Knowledge and Data Engineering, vol. 20, No. 7(July 2008)

[10] Mohammad A. Al-Katani, Ravi Sandhu. 2002. A model for Attribute-Based User-Role Assignment. In Proceedings of the 18th Annual Computer Security Applications Conference (ACSAC'02).

[11] Tingting Wang, Jiming Liu and Xiaolong Jin. 2004. Minority Game Strategies in Dynamic Multi-Agent Role Assignment. In Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04).

[12] James B.D.Joshi et al. 2005. A Generalized Temporal Role-Based Access Control Model. IEEE Transactions on Knowledge and Data Engineering, Vol.17, No.1.(Jan. 2005)

[13] Gail-Joon Ahn, Badrinath Mohan, Seng-Phil Hong. 2009. Towards Secure Information Sharing using Role-Based Delegation. Journal of Network and Computer Applications, 42-59.

[14] Josef Schiefer et al. 2007. Event Driven Rules for Sensing and Responding to Business Situations. In Proceedings of DEBS '07, 198-205.

[15] J.Dietrich, A.Kozlenkov, M.Schroeder, G.Wagner. 2003. Rule Based Agents for Semantic Web. Electronic Commerce, Research and Applications, 323-338.

[16] David F.Ferraiolo, D.Richard Kuhn, Ramaswamy Chandramouli. 2007. Role-based Access Control. Artech House, Inc.

[17] Fabio Bellifemine, Giovanni Caire, Dominic Greenwood. 2007. Developing Multi-Agent Systems with JADE. John Wiley and Sons Inc.

[18] Aglets Software Development Kit, IBM Tokyo Research Laboratory. http://www.trl.ibm.com/aglets/