JOURNAL OF APPLIED
COMPUTER SCIENCE
Vol. 21 No. 1 (2013), pp. 119-133

Multi-agent System to Assure the Logical Security of Data in Distributed Information System

Aneta Poniszewska-Marańda[1]

[1] *Institute of Information Technology*
Lodz University of Technology
Lodz, Wolczanska 215
anetap@ics.p.lodz.pl

Abstract. *The increased availability of information as a whole became an important problem and threat for its security, especially security of sensitive and confidential information and that is why the necessity to assure the security of such data became undeniable. The developers of applications an information systems put more and more stress on the aspect of their security and safety. Development of information systems has to answer more and more to problems connected to federated data sources and problems of heterogeneous distributed information systems. It is necessary to propose the architecture for secure cooperation of such systems. The paper presents the practical application of concepts of multi-agent systems in domain of logical security of data in distributed information systems. The purpose of presented solution is to support the process management of IT project realization based on the software creation methodologies.*

Keywords: *access control, distributed information systems, intelligent agents, multi-agent systems.*

1. Introduction

Currently, together with the rapidly evolving technological environments that help to simplify and accelerate the business and production processes in the companies, an increasingly important issue becomes the development of appropriate security mechanisms which would provide essential protection and safety of data, applications and intellectual property stored in information systems and used by them and their users The information technology solutions are nowadays commonly used in companies and business. More and more business operations and transactions are conducted on-line as well as more and more critical and sensitive information is being kept on local or remote servers in huge databases of company information systems.

The increased availability of information as a whole became an important problem and threat for its security, especially security of sensitive and confidential information and that is why the necessity to assure the security of such data became undeniable. The developers of applications an information systems put more and more stress on the aspect of their security and safety. Such topic is especially important when we want to ensure that created product or provides service will comply with local, global and international standards of personal data protection and it will guarantee the customers a safety storage and use of data.

As it was said above, the information systems and their databases can store a huge amount of data, personal, sensitive and confidential data that can be accessed by different users, internal and external, with different rights, responsibilities and tasks. For this reason the development of information systems has to answer more and more to the problems connected to federated data sources and problems of heterogeneous distributed information systems [1, 2, 3]. It is necessary to propose the architecture for secure cooperation of such systems. Such architecture has to solve the problems connected with structural or semantic conflicts on information in order to assure the security constraints, defined for local data sources and in order to create the control process on global level of cooperated information systems [2].

Security protection in aspects of data access, realized in federated information systems together with a loose connection among local data sources is difficult to reach because of two main reasons:

- local data sources are heterogeneous (i.e. data, models, access control models, semantic) and

- local autonomy of the systems do not allow to create global integrated security schema.

In order to solve such problems the use of intelligent agents was proposed. These agents can support the process of data access service in real time, realized by defined and undefined users. These data can be stored in different systems, subsystems and applications of federated information systems. Each of such systems or subsystems can be secured by another security policy and the agents can support the integration process of security policy on federation global level. Moreover, the role-based access control model is proposed to use in order to describe the local schema of data access control (mandatory and discretionary models). Global security policy allows to define the rules of data protection and the control of system data flow in two directions:

- import of data – from federation of systems to local system and
- export of data – from local system to the federation.

The solution presented in the paper is based on the cooperation between information system and multi-agent system with the use of interaction among the system agents [4]. It is important to assure the cooperation of local data sources and creation of coherent structure for intelligent agents. It is possible to define different types of agents in order to separate the system functionality, for example information agents that serve the global access requests, security agents, that assure the legality of local access and solve the eventual conflicts during the information access.

Role-based access control model ensure the coherence (i.e. homogeneity) of local security models [5]. This model allows to describe the local models without structural problems in organization. Moreover, the dynamic process of conflict solving can be realized with the use of techniques of multi-agent systems because it is very important to serve the global request with respect to the local security schema.

The paper presents the practical application of concepts of multi-agent systems in domain of logical security of data in distributed information systems. The purpose of presented solution is supporting the process management of IT project realization basing on one of software creation methodologies - Scrum methodology. The support consists in controlling and granting dynamically the access to the code repositories in framework of IT development team, working on certain project, based on actual status of the programmers and based on the progress of works in given sprint of Scrum methodology.

2. Intelligent agents and multi-agent systems

There are some definitions of an agent or a multi-agent system in the literature. Jannings and Wooldrige give the following definition of an agent [6, 7]:

An *agent* is a computer system or application that is situated in some environment and that is capable of autonomous action is this environment in order to meet its design objectives.

An *intelligent agent* is one that is capable of flexible autonomous actions in order to meet its design objectives: reactivity, pro-activeness and social ability.

Intelligent agents are capable of interacting with other agents, are able to perceive their environment and respond to changes that occur in it and they are able to exhibit goal-directed behaviour by taking the initiative. All these functions are made by an agent in order to satisfy their design objectives.

Agents operate and exist in some environment, which typically is both computational and physical. The environment might be open or closed, it might or not contain other agents. At times, the number of agents may be too numerous to deal with them individually and it is more convenient to deal with them collectively as a society of agents. An agent has the ability to communicate. This ability is part perception (the receiving of messages) and part action (the sending of message). Agents communicate in order to achieve better the goals of themselves or of the system in which they exist.

Multi-agent system is composed of multiple interacting software components known as agents, which are typically capable of cooperating to solve problems that are beyond the abilities of any individual member [6, 7].

A multi-agent system is one that consists of a number of agents, which interact with one-another. In the most general case, agents will be acting on behalf of users with different goals and motivations. To successfully interact, they will require the ability to cooperate, coordinate, and negotiate with each other, much as people do. Multi-agent environment provide an infrastructure specifying communication and interaction protocols for agents. It is typically open and contains agents that are autonomous and distributed and may be self-interested or cooperative.

The nature of intelligent agents is connected with the necessity of taking into consideration some specific aspect:

- *cooperation with other objects* – agent has to communicate with external objects that mostly have other interfaces then it; agent that can take data fro external word, it has to know how to change this data into information comprehensible for other agents,
- *cooperation with other agents* – this cooperation has to be realized with the use of some protocols and it has to be possible between tow agents with respect of global constraints,
- *synchronization and concurrency* – an agent often, to realize his tasks, use some thread that have to be served,
- *migration* – agent has to have the source access independent on the place that he is and this situation has to be transparency for an agent,
- *automatic conclusion* – agent chooses a service of an activity plan basing on his knowledge,
- *automatic behaviour* – agent has to able to realize his tasks.

The solutions used on modern development platforms, such as Java or .NET Framework facilitate the work and cooperation during solving of parallel issues. The problems concerning the cooperation of agents with other agents and external applications and the problem of migration can be solved by the use of *Service Oriented Architecture, SOA*.

3. Software creation methodology from the point of view of realized system

Scrum is an iterative methodology of software creation and IT project conducting, known as an agile methodology. The IT product development in this methodology is divided into smaller stages, named *sprints* that has a time-box of one to four weeks.

Scrum methodology consists of *Scrum Teams* and their associated roles, events, artefacts and rules. The Scrum Team consists of a Product Owner, the Development Team and a Scrum Master. Characteristic features of this methodology are the self-organizing and cross-functional abilities of scrum team and the ability to introduce the future system users into the process of its development and building.

Self-organizing team decides how best to accomplish it work, rather than being directed by others outside the team. Cross-functional team has all the competencies needed to accomplish the work without depending on other outside the team.

Scrum team delivers after each sprint the functioning product iteratively and incrementally, maximizing the opportunities for a feedback from the client and the users. Incremental deliveries of ”Done” product ensure a potentially useful version of working product is always available.

The *Scrum Master* is responsible for ensuring the Scrum is understood and enacted by ensuring that the Scrum Team adheres to Scrum theory, practices and rules. He or she is a servant-leader for the Scrum team. The *Product Owner* is responsible for maximizing the value of the product and the work of the Development Team.

The main event of Scrum is a *sprint* that has a time-box of one month or less during which an usable and potentially releasable product is created. Sprint consists of the Sprint Planning Meeting, Daily Scrums, the development work, Sprint Review and Sprint Retrospective.

The *Product Backlog* is an ordered list of all elements that might be needed in the created product and it is a source of requirements for any changes to be made to the product. The Product Owner is responsible for the Product Backlog, its content, availability, and ordering. Product Backlog is never complete and is dynamic because it evolves as the product and product environment.

The Product Backlog lists all features, functions, requirements and fixes that constitute the changes to be made to the product in future releases. The items of Product Backlog have the attributes to describe, order and estimate them. It is often ordered by value, risk, priority and necessity.

The development team chooses from the product backlog the tasks of the higher priority that allow to realize the project/sprint goal. The realization time of each task is estimated. The list of tasks with estimated work to do represents the *sprint backlog*. The tasks are not assigned to the team members by order from superior authority but they decide about the choice of tasks according to together assessments, own knowledge, skills, experience and other defined preferences.

The idea of Scrum methodology, its roles, artefacts and events are presented in figure 1.

We can consider the following situation during the management of IT projects realized using the Scrum methodology. Project manager manages the team of 30 people. One of the programmer leaves the team for misfortune of fate and the project manager has to find the competent deputy. He has to know exactly which

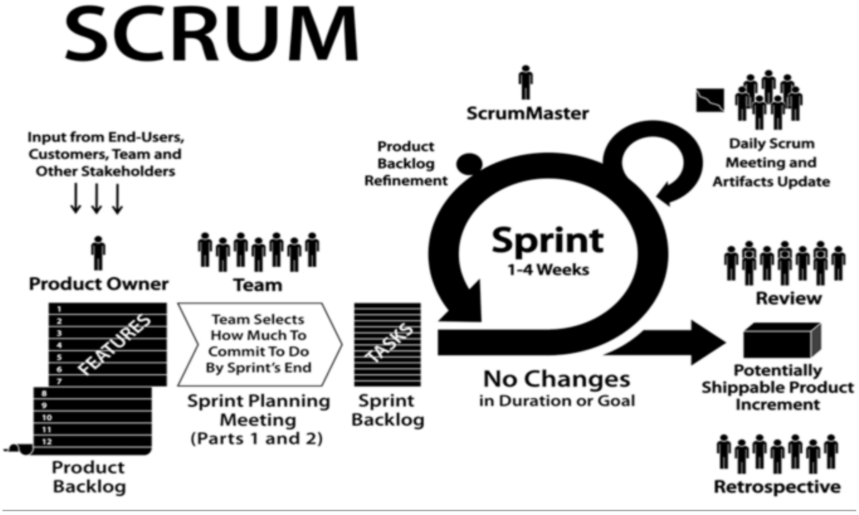
SCRUM
Input from End-Users, Customers, Team and Other Stakeholders
Product Owner
Team
ScrumMaster
Product Backlog Refinement
Daily Scrum Meeting and Artifacts Update
Sprint
1-4 Weeks
FEATURES
1
2
3
4
5
6
7
8
9
10
11
12
Product Backlog
Team Selects How Much To Commit To Do By Sprint's End
Sprint Planning Meeting (Parts 1 and 2)
TASKS
Sprint Backlog
No Changes
in Duration or Goal
Review
Potentially Shippable Product Increment
Retrospective

Figure 1. Idea of Scrum methodology

skills has the leaving programmer, which were his responsibilities in a project. Basing on these information he can find a programmer who probably manage with the tasks of leaving programmer [4].

Solution of this problem can be automatized in great measure by the creation of a system that will takes over part of manager responsibilities such as:

- Verification of knowledge of a programmer to replace – the project manager verifies the experience level of a programmer, average time needed to realize a task, number of errors generated by a programmer.
- Verification of tasks for which the programmer was responsible – the project manager as a person responsible for project progress, has to know the responsibility range of the programmer, has to know the level of task complicity. Basing of these factors, he finds a programmer who manages the best with the given tasks.

The describe situation can be connected with the management of access to system resources. We can grant some user to access the particular, critical data but we can also refuse the other user. The user ”Guest” can also exists in a system but he has very limited access.

To solve this issue and to preserve the integrity and security of distributed information system, the usage of multi-agent system was chosen. The main purpose of such system is to manage and monitor an access to code repositories of project groups that use the specific methodology of software creation. The use of chosen methodology results in the following situations [4]:

- development team assign dynamically the tasks to the developers according to their preferences and qualifications,
- members of development team should be focused only on one project, even a scrum master should take care of one team what is a very rare practice,
- client can actively take part in the process of product development and in consequence he has an access to particular resources of development IT company,
- reduction of costs in IT companies provokes that the use of cloud computing and store of data in clouds become more and more popular and profitable,
- several times a company has a few departments in different countries and the servers storing the data are distributed.

4. Intelligent agents supporting the management of access rights in distributed information systems

The exemplary situations presented above cause the usage of intelligent system monitoring an access to the resources of development company operating in distributed environment. The monitoring of access rights to the resources should be realized with the use of intelligent agents from multi-agent systems assigning appropriate access rights to the company resource. These agents can be divided according to their functionality as follows [4]:

- *Managing Agents* – they manages the works of other types of agents, evaluating the data collected by them and transferring the data to other managing agents.
- *Agents monitoring the sprint* – they are responsible for granting the access rights the members of development team, depending of tasks assigned to the

team members that are given in Scrum tickets. The *tickets* define the duties given the particular team members and reflect the needs and requirements of a client.

- *Agents monitoring work division* – these agents are responsible for detection of conflicts connected with assigning too much tasks to the team members in framework of given sprint or in framework of work in some projects of particular person. Moreover, they can inform about eventual lack of task assignment or about the assignment of tasks with a high priority to inexperienced worker. The task of this type of agents is evaluation of number of tasks assigned to each employee.

- *Agents monitoring physical security* of a system – they are responsible for correct communication between the system of internal company security, the system monitoring the presence of employees in IT firm and the multi-agent system. They assign the appropriate right to the files in situation of employee absence. They also define the access rights of team members depending of their connections with an employee server.

- *Agents monitoring resource access* by particular programmers – they are responsible for automatic assignment of access to certain resources, located on other servers, with the use of login and password of team member, based on data placed in tickets, assigned to a given sprint. These agents control the progress of works.

Process of information exchange between the described types of agents is presented in figure 2.

5. Types of intelligent agents and their cooperation in created solution

Five types of agents were proposed in created multi-agent system to assure the logical security of data in distributed information systems – AAC-DIS (*Agents for Access Control of Distributed Information Systems*) system. The flow of data between the agents and another elements of information system is presented in figure 3.

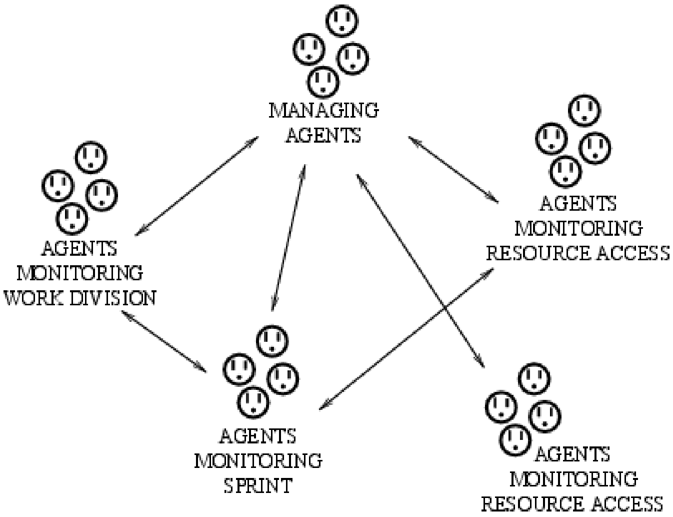
MANAGING AGENTS
AGENTS MONITORING RESOURCE ACCESS
AGENTS MONITORING WORK DIVISION
AGENTS MONITORING SPRINT
AGENTS MONITORING RESOURCE ACCESS

Figure 2. Exchange of information between the agents in AAC-DIS system [4]

5.1. Managing Agents

Managing Agents supervise the work of other groups of agents, checking the data collected by them and transferring this data to other managing agents. Moreover, additional managing agent exists and it is responsible for logging of information to other instance, for example to the administration panel [4].

5.2. Agents monitoring the sprint

Agents monitoring the sprint are responsible for assigning the permissions (i.e access rights) to the members of development team depending on their allocated tasks, named the *tickets*. Tickets determine which duties were assigned to the particular team members and they reflect the needs and requirements of a client [4].

The agents monitoring the sprint operate recursively as other types of agents. There is no need for too frequent monitoring the sprint, so the monitoring can be done once a day, at the beginning or at the end of the work day. The decisions are taken concerning the degree of implementation schedule. The most important information for the Scrum Master is the overall level of the status of work on the project, without going too much into the details of specific tasks, and the more implementation details.

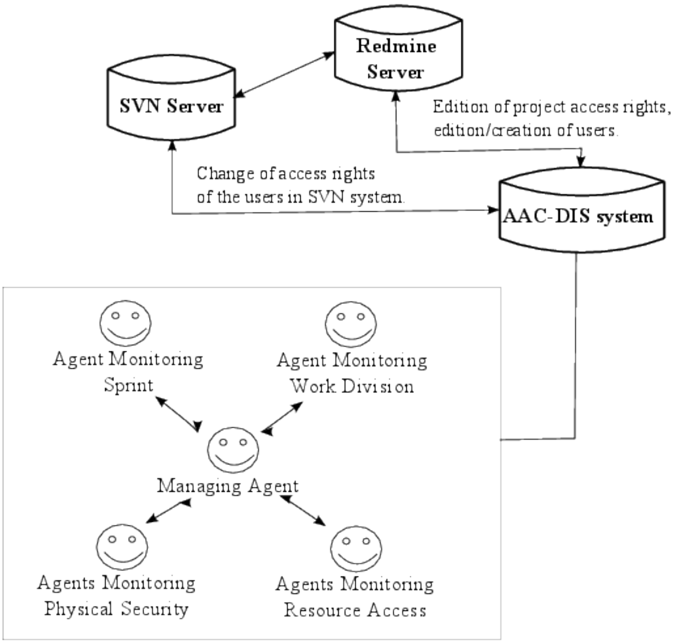
Redmine Server
SVN Server
Edition of project access rights, edition/creation of users.
Change of access rights of the users in SVN system.
AAC-DIS system
Agent Monitoring Sprint
Agent Monitoring Work Division
Managing Agent
Agents Monitoring Physical Security
Agents Monitoring Resource Access

Figure 3. Data flow in framework of created multi-agent system AAC-DIS [4]

The agents monitoring the sprint are very useful tool for controlling the project by the project manager because they provide the information on the status of implementation of tasks in relation to defined project schedule and they allow easily keep track of the work progress and respond to emerging problems [4].

The next task of the agents monitoring the sprint is assignment the new people to a project and deleting selected people from it. In the case when the team has not kept a pace with the realization of the tasks in accordance with the plan, the additional customers are needed. The agents monitoring the sprint can inform of such need, when when they see that some tasks are delayed in relation to the timetable. As a consequence, can ”move” the workers from other project to the project on the basis of their skills and experience and on the basis of the requirements necessary for the implementation of delayed tasks.

In the case when the movement of people between projects is difficult, for example because of staff shortages, the agents may decide to move the workers on the basis of the priorities of individual projects or to report a problem to *project manager* [4].

The agents monitoring the sprint are also responsible for the assignment of the access the relevant branches of project source code to the team member, depending on the tickets assigned to them. The employee, even though the participation in the project, does not have to an access to the whole project. In accordance with the principle of *least privileges* agents may allocate such access to him at the time, when he will begin working on a task and in strictly determined range or they may revoke him an access at the end of the work.

A member of design team may also need an access to the code from outside of his part of an application, as well as the code form another application. Then he may receive the permissions to read only the certain branch of code, without the possibility of its edition[4].

5.3. Agents monitoring work division

Agents monitoring work division are responsible for the conflict detection, related to the allocation of team members too large number of tasks in the Sprint or as part of the work of a certain person in a few projects. Moreover, they can inform about the possible absence of the allocation or assignment of high-priority tasks to a novice employee [4].

The task of this group of agents is to check the quantity of tasks that are assigned to each employee. The maximum number of tasks is calculated on the basis of [4]:

- Experience of an employee – the new employee during a job interview is allocated a certain amount of experience points and depending on this assessment it is employed or not. The obtained points characterize the person's experience. After each month of work the agents monitoring the sprint update (add or subtract) the number of experience points for each employee.

- The number of hours worked in the previous month – if the concerned person works a lot and efficiently, then the agents can assign him/her the additional tasks, the performance of which will increase him/her level of experience. This allows to get a clear and transparent model of reward the employees in the company, while at the same time to increase their productivity.

- Number of tasks in a project – if the number of tasks to be performed in the project is a big, it is necessary to assign more tasks the employees. If at given time, it projects missing the employees do not need to perform the additional work. The number of available tasks affect the coefficient of tasks in the framework of the project, specifying the maximum number of tasks for an employee on the basis of his/her tasks in a project, his/her experience and the number of working hours.

5.4. Agents monitoring physical security

Agents monitoring physical security are responsible for the correct communication between the internal security system of a company, the system monitoring the presence of workers in the work and the multi-agent system. They assign the appropriate permissions to the files in case of absence the employee at work (this fact is recognized by the absence of recording physical access to the door of the room by using his identification card. They determine the team members access rights depending on the connection with the staff server by the employee [4].

This type of agents are tasked with distributing or receiving the permissions to certain resources on the basis of the information concerning the presence of each employee in a company. These agents can modify the permissions assigned by *agents monitoring resource access*. Agents monitoring the physical security control the data flowing into the system, that can come for example from the identification card readers placed at the entrances to the company. They determine who the employees are present at work on the basis of the collected information.

In a situation when the agent will confirm the presence of the employee at the company, unlocks for this employee all assigned permissions. If the employee is absent, all permissions remain locked. The exception is when an employee has a permission for remote access to the corporate resources and he/she is at the moment in the business trip. The agent's task is to verify that both of these conditions are met - if so, the agent unlocks the permissions given to the worker, despite its physical absence in the company [4].

5.5. Agents monitoring resource access

Agents monitoring resource access are responsible for the automatic allocation of access to the data resources, stored on other servers, with the use of login and password of a team member, on the basis of data located in *tickets*, assigned to the certain sprint. These agents control the progress of work, giving the possibility of access to the recent code revision [4].

This type of agents perform their work repeatedly, like the other agents. Their goal is to preserve the confidentiality and integrity of stored and provided data. From the point of view of *product owner* in Scrum, it is important that the information on the project, for example the documentation or the code is secure. Therefore, the access to them is restricted only to a limited group of employees. To the resources managed by the agent included the repositories, documentation and hardware platform on which the tests are carried out, the configurations are optimized or implemented the final versions of the product.

The agent can allocate or deprive of the permissions on the basis of the information received from the agents communicating with it and on the basis of the current state of database, contained the information necessary for the operations. This information are included in the *tickets*, assigned to the sprint. They are placed by *scrum master*, who on the basis of his knowledge of realized project and the knowledge, experience and characteristics of the individuals participating, evaluates the need for access to the realisation of a task by a member of the team [4].

Thanks, the *scrum master* does not have to waste a time manually granting the permissions to resources, and he can act the role of expert. Similarly, it is important the automatic control of the permissions and deprive them in the completion of work on a task.

6. Conclusions

The presented paper focuses on the access control security in distributed information systems. Since the information systems are more open nowadays, which means also that more information is easily accessible to users, the task of better protection of confidential information becomes of essential importance. To solve these problems we propose to use the concepts of intelligent agents with their principles and abilities. This solution can preserve the control of data flow in the cooperative systems with respect of all security rules defined in each local system.

The presented example of multi-agent system for the management of logical security in an application or information system shows the possibilities of connection of both concepts, i.e. intelligent agents and access control in order to create the practical solution that is elastic and can be developed depending on specific needs of the particular situation.

The approach of multi-agent systems can be used in different domains of distributed information systems, e.g. electronic commerce, travel applications, public administration, management of university, management of hospital network.

References

[1] Disson, E., Boulanger, D., and Dubois, G., *A Role-Based Model for Access Control in Database Federations, Information and Communications Security*, In: Third International Conference, ICICS 2001, Xian, China, 2001.

[2] Poniszewska-Maranda, A., *Access Control of Federated Information Systems*, In: LNCS 5376, pages 119-130, Publisher: Springer-Verlag Heidelberg, 2008.

[3] Poniszewska-Maranda, A., *Multi-agent systems for access control in distributed information systems*, Scalable Computing: Practice and Experience Journal, Vol. 12, No. 4, 2011, pp. 403–415.

[4] Groschang, K., Margielewski, D., Szymański, R., Urbańczyk, D., Wach, L., and Wawrzyniak, K., *System wieloagentowy do zapewnienia bezpieczeństwa logicznego danych w rozproszonych systemach informatycznych*, Tech. rep., Supervisor: A. Poniszewska-Maranda, Institute of Information Technology, Lodz University of Technology, 2012.

[5] Ferraiolo, D., Sandhu, R. S., Gavrila, S., Kuhn, D. R., and Chandramouli, R., *Proposed NIST Role-Based Access Control*, ACM, Transactions on Information and Systems Security (TISSEC), Vol. 4, No. 3, 2001.

[6] Singh, M. and Huhns, M., *Readings in Agents*, Morgan-Kaufmann Pub., 1997.

[7] Wooldridge, M., *An Introduction to MultiAgent Systems*, John Wiley and Sons, 2002.