Scalable Computing: Practice and Experience
Volume 12, Number 4, pp. 403–415. http://www.scpe.org
ISSN 1895-1767

MULTI-AGENT SYSTEMS FOR ACCESS CONTROL IN DISTRIBUTED INFORMATIONS SYSTEMS

ANETA PONISZEWSKA-MARANDA*

Abstract. The modern information systems evaluate very quickly. The information is more and more distributed through the networks or federation of numerous information systems located in different places on the globe. Also, the control domain of information system is very important in the times of very fast networks, telecommunication protocols and telecommunication equipment.

The paper presents the proposal of cooperation between the information systems and multi-agent systems. It is necessary to assure the cooperation of local data resources and create the coherent structure for intelligent agents. The dynamic process of conflict solving can be realized by using different techniques that come from the multi-agent systems.

Key words: access control, distributed information systems, access control models, multi-agent systems

AMS subject classifications. 68N02, 68T02

1. Introduction. The modern information systems evaluate very quickly. The information is more and more distributed through the networks or federation of numerous information systems located in different places on the globe. Also, the control domain of information system is very important in the times of very fast networks, telecommunication protocols and telecommunication equipment.

Not less important is the data protection against improper disclosure or modification in the information systems. This requirement is always obligatory in the development process of information system and its security domain as well in the other phases of information system lifecycle. But nowadays the information engineering, as well as another information science domains, changes very quickly and the new products appear every day. Also the access modes to the information have been changed. The new protocols appear to exchange the information. All these changes provoke the new security problems against which the existing models or architectures of access control should make a stand.

The appearance of new business models for the organization and enterprise activities in the network and the appearance of new protocols for information exchange provoke that the information is more and more distributed and the traditional access models are insufficient to solve the problems of information control. Also, it is important to protect the information against the non-controlled utilization and on the other hand we would like to control the usage and the diffusion of this information. It gives the possibility to specify how it can be used and specify the utilization constraints. All these new problems are connected with the usage control. This new term in security domain necessities the mechanisms to apply the usage control that should be distributed in the information systems.

Development of information systems should answer more and more to the problems of federated data sources and the problems with heterogeneous distributed information systems [13]. The assurance of access data security realized in federated information systems with loose connection among local data sources is hard to achieve mainly for two reasons: local data sources are heterogeneous (i.e. data, models, access security models, semantics, etc.) and local autonomy of systems does not allow to create a global integrated security schema. To solve such problems we propose to use the intelligent agents that can assist in the process of real-time access by defined and undefined users to the data stored in different systems, subsystems or applications of federated information systems. Each of these systems or subsystems can be secured by a different security policy and the agents can help in the process of security policy integration on a global level.

This paper presents the proposal of cooperation between the information systems and multi-agent systems using the interactions between the system agents. It is necessary to assure the cooperation of local data resources and create the coherent structure for intelligent agents. It can be made by using the unified security model to exchange the data and to access them. The model based on the roles can assure the homogeneity of local security models and allows the description of local models. The dynamic process of conflict solving can be realized by using different techniques that come from the multi-agent systems.

The paper is structured as follows: section 2 presents the access control policies and models - traditional access control models, models based on role concept and models based on usage concept that particularly

*Institute of Information Technology, Technical University of Lodz, Poland (anetap@ics.p.lodz.p)

interesting for distributed information systems. Section 3 deals with the security problems of distributed information systems and presents the concept of agents and multi-agent systems. Section 4 presents the proposition of architecture based on multi-agent approach for secured cooperation in distributed information systems.

2. Access control policies and models. The security policies of a system generally express the basic choices taken by an institution for its own data security. They define the principles on which the access is granted or denied. Access control imposes the constraints on what a user can do directly, and what the programs executed on behalf of the user are allowed to do. In information system the access control is responsible for granting direct access to system objects in accordance with the modes and principles defined by the protection policies.

2.1. Access control models. It is possible to distinguish two categories of security policies of the information systems: discretionary security policy and mandatory (non-discretionary) security policy. We can find some access control models based on these policies [1, 2, 3]:

Discretionary security model - manages the users' access to the information according to the user identification and on the rules defined for every user (subject) and object in the system. For each subject and object in a system there are authorization rules that define the access modes of the subject on the object. The access modes: read, write and execute, are verified for each user and for each object. The access to the object in the specific mode is granted only to the subjects for whom an authorization rule exists and is verified. Otherwise it is denied. "Discretionary" means that users are allowed to grant and revoke access rights on particular objects. This implies decentralization of the administration control through ownership [1, 2].

Mandatory (non-discretionary) security model - manages the access to data according to classification of the subjects and objects in a system. Each user and each object of a system are assigned to specific security levels. The access to data is limited by the definition of security classes. Subjects and objects in a system are related to security classes, and the access of a subject to an object is granted if the relation between the classes of the subject and the object is verified [1, 2].

Role-Based Access Control model - RBAC model - regulates the access of users to the information on the basis of the activities that the users perform in a system. This model requires identification of roles in a system. The role can represent competency to do a specific task and it can embody the authority and responsibility. The roles are created for various job functions in an organization and the users are assigned to the roles based on their responsibilities and qualifications. The user playing a role is allowed to execute all access modes to which the role is authorized. RBAC model provides support for several important security principles (notably least privilege, privilege abstraction and separation of duties), but does not dictate how these should be put into practice [3].

Extended RBAC model - the complexity of organizations gave rise to the idea of extending the standard RBAC model to ensure a finer decomposition of the responsibilities in an enterprise. To this aim, the notion of a function has been introduced. Since a person employed in an enterprise may have many responsibilities in it, he may be attached to an entire set of roles. Each role defined in the extended RBAC model makes it possible to realize a specific task associated with the enterprise process. At the same time, every role can contain many functions that a user can take, and therefore it is possible to choose functions in the system that are necessary for a given role. Therefore, the classical RBAC model was extended by addition of some elements to express more complex elements of information system secured by security model (Fig. 2.1) [4, 5].

Usage Control (UCON) model - it is based on the three decision factors: authorizations,, obligations and conditions that have to be evaluated for the usage decision. It consists of eight main components: subjects, objects, subject attributes, object attributes, rights, authorizations, obligations, and conditions. Subjects, objects and rights can be divided into several detailed components with different perspectives. The UCON strategy is characterized by two features: mutability and continuity. Mutability means the mutability of subject and object attributes - with a mutability property the attributes can be either mutable or immutable. The mutable attributes can be modified by the actions of subjects and the immutable attributes can be modified only by the administrative actions. The continuity means that a decision can be made even after an access (Fig. 2.2) [10].

2.2. Access control approach for dynamic information systems. Modern information systems are very often dynamic, distributed and heterogeneous in different aspects. They can contain many different components, applications, located in different places in a city, in a country or on the globe. Each of such components can store the information, can make this information available to other components or to different users. The

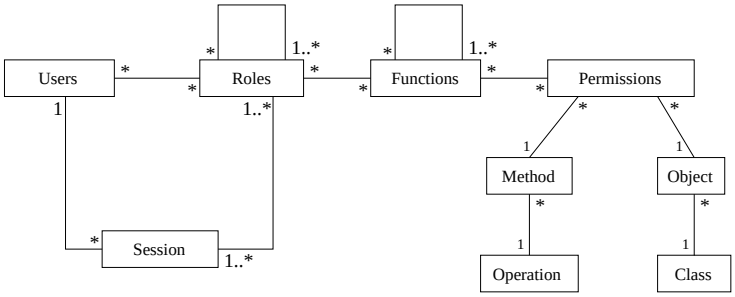

Fig. 2.1: Elements of extended RBAC model

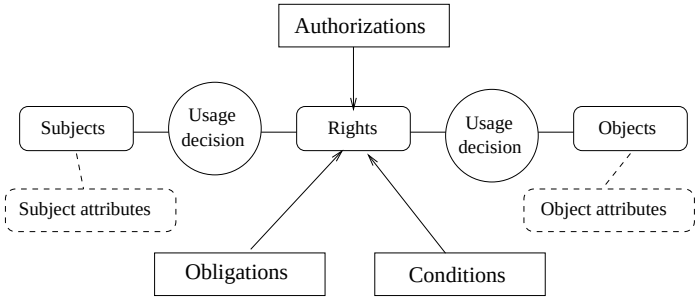

Fig. 2.2: Elements of Usage Control model

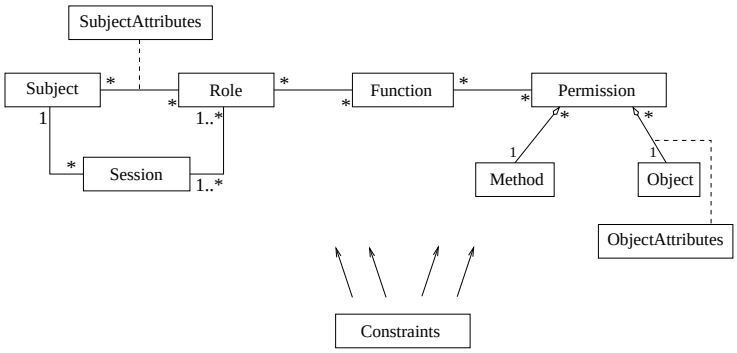

Fig. 2.3: Usage Role-based Access Control approach

authorized users accessing the information can change this information, its status, role or other attributes at any time. These changes can cause the necessity of modifications in security properties of accessed data on access control level. Such modifications are dynamic and often should be realized ad hoc because other users from other locations can request the access to the information almost at the same time.

The new access control approach was based on two access control models: extended RBAC model [2, 3] and UCON model [10]. It was named Usage Role-based Access Control (URBAC) (Fig. 2.3) [17]. The term usage means usage of rights on information system objects. The "rights" include the rights to use particular objects and also to delegate the rights to other subjects.

The core part of URBAC model essentially represents the extended RBAC model. Subjects can be regarded as individual human beings. They hold and execute indirectly certain rights on the objects. Subject permits to formalize the assignment of users or groups of users to the roles. Subject can be viewed as the base type of all users and groups of users in a system. The aggregation relation SubjectGroup that represents an ordering relation in the set of all system subjects can assign subjects to the groups.

A Role is a job function or a job title within the organization with some associated semantics regarding the authority and responsibility conferred on a member of the role. The roles are created for various job functions in an organization. The direct relation is established between roles and subjects that represent the users or groups of users. It is also possible to define the hierarchy of roles, represented by aggregation relation RoleHierarchy, which represents the inheritance relations between the roles.

The association relation between roles and subjects is described by the association class SubjectAttributes that represents the additional subject attributes (i.e. subject properties). Subject attributes provide additional properties, describing the subjects that can be used for the usage decision process, for example an identity, enterprise role, credit, membership, security level.

Each role allows the realization of specific task associated with an enterprise process. A role can contain many functions that a user can apply. A role can be viewed as a set of functions that this role can take to realize a specific job. It is also possible to define the hierarchy of functions, presented by aggregation relation named FunctionHierarchy, which represents the inheritance relations between the functions.

Each function can perform one or more operations that this function needs to be defined as a set of permissions. To perform an operation one has the access to required object, so necessary permissions should be assigned to corresponding function. The permission determines the execution right for a particular method on the particular object. Very often the constraints have to be defined in assignment process of permissions to the object. Such constraints are represented first of all by the authorizations. Authorization is a logical predicate attached to a permission that determines the permission validity depending on the access rules, object attributes and subject attributes. A constraint determines that some permission is valid only for a part of the object instances. Therefore, the permission can be presented as a function $p(o, m, c)$, where o is an object, m is a method which can be executed on this object and c is a set of constraints which determine this permission. Taking into consideration a concept of authorization, the permission can be presented as a function $p(o, m, A)$, where A is a set of the authorizations determining this permission. The constraints defined on permission can be also determined by the obligations and conditions.

The objects are the entities that can be indirectly accessed or used by the users. The relation between objects and their permissions are additionally described by association class ObjectAttributes that represents the additional object attributes (i.e. object properties) that cannot be specified in the object's class and they can be used for usage decision process. They can be also mutable or immutable as subject attributes do.

The security constraints can be defined for each main element of the model presented above and also for the relationships among the elements. The concept of constraints is described widely in the literature [3, 4, 11, 12]. It is possible to distinguish different types of constraints, static and dynamic that can be attached to different model elements. The URBAC approach distinguishes the following general types of constraints:

- Authorizations - constraints defined for the permissions, basing on access rules defined by enterprise security policy but also basing on objects' attributes and subjects' attributes.
- Obligations - the subject can be associated with the obligations which represents different access control predicates that describe the mandatory requirements performed by a subject before (pre) or during (ongoing) the access.
- Conditions - session is connected with the set of conditions that represent the features of a system or application. They can describe the current environmental or system status and states during the user session that are used for the usage decision.
- Constraints on roles and on functions. The most popular type in this group of constraints is Separation of Duty (SoD) constraints [3, 4, 11, 12].
- Constraints on relationships between the model elements [3, 4, 11, 12].

3. Access control for dynamic distributed information systems. The components of the information systems, i.e. applications, databases are typically distributed and heterogeneous. The topology of these systems is dynamic and their content is changing sometimes very quickly or rapidly and it is difficult for a user of an application to obtain the correct information or for the enterprise to maintain the consistent information. The information systems are large and complex in several meanings [13]:

- they can have many components, i.e. applications, databases,
- they can have huge content of the number of concepts and of the amount of the data about each concept,
- they can be geographically distributed,
- they can have a broad scope, i.e. coverage of a major portion of a significant domain.

In distributed information system each local component can be secured by another access control model. The objectives of the security policy in cooperative information systems are to respect the local security model of each system (each model specifies the security principles of a local system) and to control the indirect security connected with the global cooperation level: a member of a local system may in another local system access only the equivalent information according to his local profile. Each system can have other security policy for describing the access control rules to access its data. This situation can involve some difficulties and heterogeneities in definition of the global security model.

In order to better allow defining of the access control of distributed information systems it is necessary to have more expressive access control model. This model should allow the greatest structure of security policy to make possible the decomposition of this policy and make easy its definition. It should be possible to express more that the simple authorizations but also the interdictions or obligations that should be fulfilled in order to obtain the access to the information system. Also the model should allow expressing of the rules assigned to the conditions on the system state or on the access context of a system [15, 16].

Four major techniques exist for handling the huge size and complexity of such enterprise information systems: modularity, distribution, abstraction and intelligence. A very reasonable solution is to use the intelligent, distributed modules which are the components of the entire information system. Using this concept, the intelligent agents or computational agents can be distributed and embedded throughout the enterprise. The agents could act as intelligent programs working for the applications, as active information resources, "actors" that surround and buffer conventional components, or as the on-line network services. These agents should have the great knowledge about information system resources that are local to them and they should cooperate with other agents to provide the global access to the information in the data flow from and to the information system. Multi-agent systems are the best way to characterize and design the distributed information systems because of the large size of systems, their dynamism and the needs of formulation and implementation of the global principles and solutions.

Some definitions of an agent or a multi-agent system can be found in literature [7, 8]:

"An *agent* is a computer system or application that is situated in some environment and that is capable of autonomous actions in 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."

Agents operate and exist in some environment that 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 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 messages). Agents communicate in order to achieve better goals for themselves or for 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 the problems that are beyond the abilities of any individual member [7, 9].

A multi-agent system consists of a number of agents that interact with one-another. In the most general case, the agents will be acting on behalf of the 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 provides an infrastructure specifying the communication and interaction protocols for the agents. It is typically open and contains the agents that are autonomous and distributed and may be self-interested or cooperative.

3.1. Related works. The concept of agents and multi-agent systems used for security of information systems, in particular for access control can be found in some works presented in the literature.

Kagal in [21] proposes solution based on trust management with the use of agent concept that can be applicable to distributed informations system. This solution involves developing a security policy, assigning credentials to entities, verifying that the credentials fulfill the policy, delegating trust to third parties, and reasoning about users access rights. However, this proposition does not take into consideration the dynamic aspects of security and access control.

In [22] Varadharajan, Kumar and Mu propose a security agent based approach for the authorization aspects in distributed environment. They define the security agents used to capture the privileges and a part of security

policy in distributed authorization. The introduce some concepts - the principles make use of the agents to carry out their requests on the remote hosts the targets verify the authenticity of security agent and its privileges and use them together with their local security policy to grant or deny the requests. The operations of authorization system is described using these security agents and the use of agents to support dynamic decision making is also presented. This proposition does not concern the dynamic and heterogeneous access control.

Antonopoulos et all. in [18] propose distributed access control architecture that is based on the concept of distributed, active authorization entities (lock cells). The combinations of these entities can be referenced by an agent to provide input and/or output access control. The authors presents how these authorization entities can be used to implement security domains and how they can be combined to create composite lock cells. However, this architecture does not concern the dynamic aspects of information system security.

Seleznyov in [19, 20] presents conceptual architecture for an autonomic middle-ware component designed to provide the application-independent access control, named ADAM. This component can be used in large-scale dynamic computing environments. Such environments do not allow to determine the centralized access control policy because of the complexity of trust relationships. The architecture is based on multi-agent system. The agents dynamically organize themselves into cooperating distributed communities that mediate between users and devices (collectively known as trustees) and network resources (principals). Authorization decisions in ADAM are based on negotiations between two agents: user agents and authorization agents. The user-agent contains information about its legal owner, including secret keys, and certificates required for the user authentication. Authorization agents protect network resources by ensuring that only valid users can obtain access to them. They are enforce the security policies and procedures of the resources that they manage. This solution allows to manage the dynamic information systems on the access control level but do not concern the problems of heterogeneous informations system. Furthermore, it functions on total absence of explicitly stated organizational policy.

In [23] Weippl et all. presents the project SemanticLIFE that stores an individual's entire digital life and makes it available to co-workers. The presented security scheme is based on implementing role-based access controls through the usage of database systems. Security policies offer a more flexible and much robust way for security administration than in the project ADAM.

Belsis et all. present in [24] the proposition of system architecture for knowledge management from the point of view of security and access control. This architecture is based on Role-base Access Control model and use the concepts of intelligent agents. This solution proposes a distributed, resilient knowledge management architecture to handle knowledge assets exchange between autonomous domains in distributed information systems. This proposition represents the domain of static access control and does not take up the aspect of heterogeneous information systems.

4. Access control architecture for dynamic distributed information systems. The agent approach in the information system security can be considered as follows: the agents can be used to assist the security policies already defined in a federation of distributed information systems, to preserve the data security on the higher level or to solve the security problems attached to the real-time access desired by some users. The agents can use the security rules already defined in a system to decide whether or not to give the users the access to the desired data or to the part of it.

The distributed information systems can also base on the communication between the agents and on the communication of external users with the agents. The agents exchange the data between themselves or with users. The systems communicate with each other by means of agents, which exchange the information, search, explore the data from one system to another. We can distinguish some types of agents in this situation: search agents (explore agents), exchange agents, communication agents (Fig. 4.1).

The incorporation of the local data sources on the global level in federation of cooperative information systems can be divided into two phases:

- definition and representation of local data exported to the global level using the corresponding descriptive elements to obtain the *described local schema*,
- allocation of these local schema on the global level and their assignment to the particular security agents.

In general, there are: *security objects* (passive data entities) and *security subjects* (active entities like users). These elements can be described by the *Security Entities (SE)* that are initiated from SE classes (i.e. Data, User, Application, System). The system SE classes describe the general security strategy of the local system.

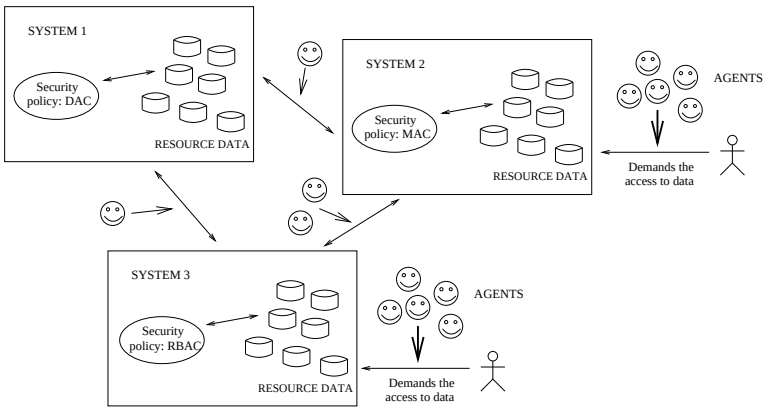

Fig. 4.1: Cooperative information system with the agents

The local security authorization units like groups (DAC models), MAC "containers", roles (RBAC models, URBAC model) or subjects (UCON model) can be described by *Access Model Entities (AME)*. Additionally, one more structure can be defined, *Information Entities (IE)* that represents all these elements on the global level and assures the homogeneous representation of each local information entity.

Taking into consideration the representation of security elements on two levels in the security aspects of cooperative information systems given above, we can enrich the process of incorporation of the local data sources on the global level as follows (Fig. 4.2):

- definition of the local data exported to the global level using the corresponding descriptive elements to obtain the *described local schema*,
- description of each data element from the exported local data schema using the security structures of semantic model given above,
- representation of these local schema using the semantics of unified security model (for example based on roles),
- allocation of the local security schema on the global level and their assignment to the particular security agents.

Therefore, the security architecture for the federation of cooperative information systems can be defined on four levels. The first level, representing the definition of local data exported to the global level using the corresponding descriptive elements, contains the exported data schema joined with the local data.

The main stages of this process (i.e. the creation of system application Model and creation of user profiles based on the access control models) for three types of access control models (MAC, DAC and eRBAC) are presented in [14].

This creation is possible with regards to the features of access control concepts [1] and the concepts of access control models. It can be realized by the automatic transformation of XML files, containing the application elements in approach of concepts of access control models (DAC/MAC/eRBAC/UCON/URBAC), to the XML file(s) containing these elements using the common concepts. First of all, it is necessary to create the DTD (Data Type Definition) files for these XML files to define their structures. The root elements of DTD files for each access control model are given as follows:

- DTD file for DAC model:
  $< !ELEMENT\ DAC(user+, object+, operation+, userIdentification*) >$
- DTD file for MAC model:
  $< !ELEMENT\ MAC(user+, object+, operation+, authorizationLevel*,$
  $classificationLevel*, category*) >$
- DTD file for eRBAC model:
  $< !ELEMENT\ eRBAC(user+, role+, function+, permission+, method+,$
  $object+, operation+, class+, constraint*) >$
- DTD file for UCON model:

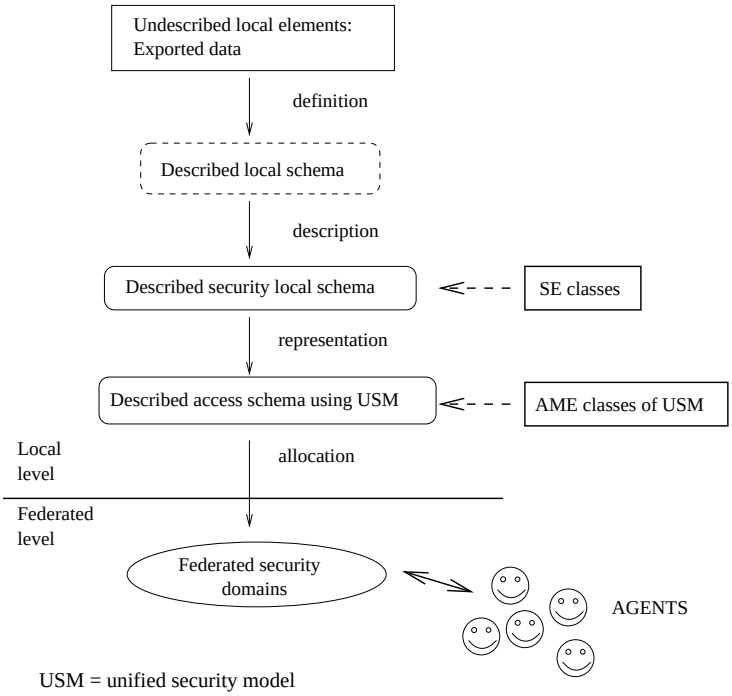

Fig. 4.2: Security incorporation process of data on the federation level

$< !ELEMENT\ UCON(subject+, subjectAttr+, object+, objectAttr+,$
$right+, authorization*, obligation*, condition*) >$

- DTD file for URBAC model:
  in general:
  $< !ELEMENT\ URBAC(subject+, subjectAttr*, role+, function+,$
  $permission+, method+, object+, objectAttr*, constraint*) >$
  in details:
  $< !ELEMENT\ URBAC(user+, userGroup*, subjectAttr*, role+,$
  $function+, permission+, method+, object+, objectAttr*, authorization*,$
  $obligation*, condition*) >$

The second level of presented above process of incorporation of local data sources on the global level is composed of the set of semantic descriptive elements and the security structures.

In an information system the access control is responsible for granting direct access to the system objects in accordance with the modes and principles defined by the protection policies. An access control system defines: the *subjects* (active entities of a system) that access the *information* (passive entities) executing different *actions*, which respect the access rules. The subjects can describe the *users* or the processes that have access to the data stored in a system. The information, i.e. the data, determines the system *objects* on which the actions represented by the most popular *operations*, i.e. read, write, delete, execute, can be performed. Therefore, it is possible to distinguish three main sets of elements describing the access control rules: **subjects**, **objects** and **operations** and two additional sets: **subject attributes** and **object attributes**.

We propose to represent the elements of access control models, i.e. DAC, MAC, eRBAC, UCON and URBAC, with the use of these sets of concepts and additionally the concept of constraints (Fig. 4.3, Fig. 4.4 and Fig. 4.5).

The root element of DTD file containing the common concepts for describing the security elements of each access control model is as follows:

$< !ELEMENT\ commonModel(subject+, subjectAttr*, object+, objectAttr*,$
$operation+, constraint*) >$

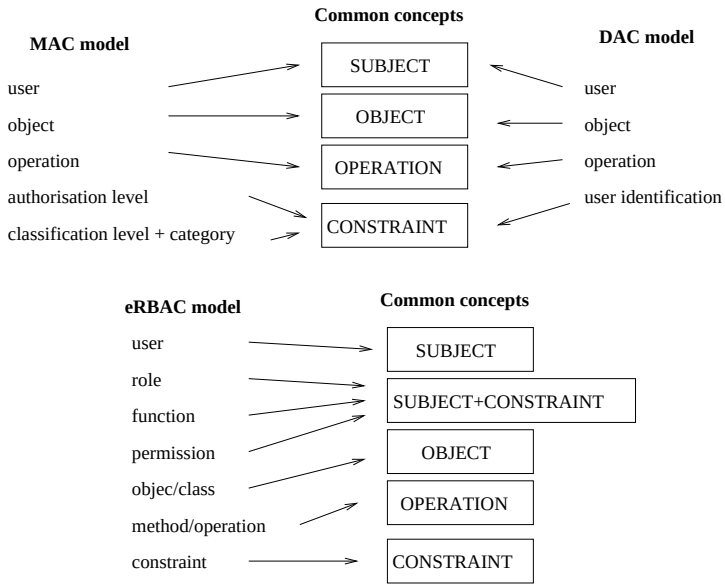

Fig. 4.3: Common concepts for access control models (DAC, MAC and eRBAC)

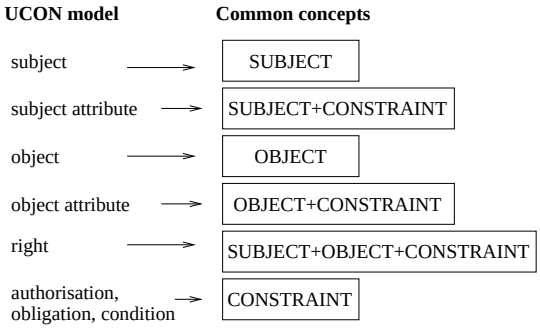

Fig. 4.4: Common concepts for UCON model

It describes the common concepts of heterogeneous security systems. The XML files based on such DTD file can be intended for the security administrator(s) to manage the federation of heterogeneous security systems.

The third level of the presented process contains the security common elements based on the semantics of unified security model. This unified security model can be based on the extended RBAC model or on the UCON model. The question is: which model or which strategy is good or quite enough to manage the access control in distributed information system, that can change sometimes very quickly and the topology of these systems is dynamic? Traditional access control models, trust management, DRM (digital right management) or usage control? Maybe the chosen security strategy should be extended by adding of the new elements to support the administration of information system security?

Actually, it seems that the most proper model to support the security of dynamic information systems is the UCON model. In this model the security policy is dynamic because it can change during the information access. The dynamic change of security policy can be translated by the change of the values of subject attributes or object attributes there are mutable attributes. The modification of an attribute can be realized before the information access, during the information access or at the end of the access.

However, the unified security model should be determined to have one common notion to express the access control elements from different models and levels.

The last level of the presented process contains the security agents and their connections with the security

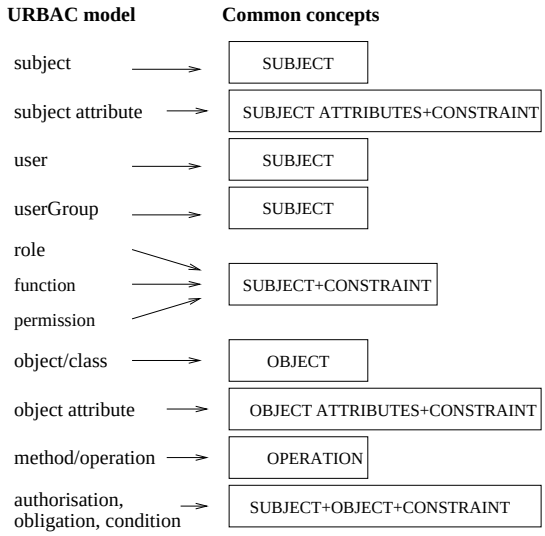

Fig. 4.5: Common concepts for URBAC model

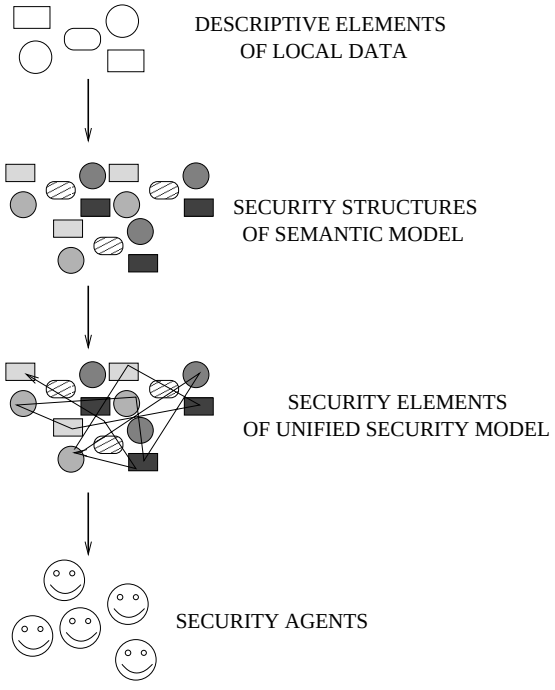

Fig. 4.6: Security architecture for federation of cooperative information systems

domains containing the elements that came from the local levels (Fig. 4.6). These agents are specialized in different tasks - it is possible to distinguish different types of agents, e.g. management agents, security agents, semantic agents or organization agents.

The agents cooperating with the federation security domains on the global levels manage the different systems functions, particularly on the access control level (Fig. 4.7):

- *security agents* are responsible for the management of global security domains,
- *semantic agents* manage the local semantic domains and the global semantic domains composed of the Information Entities and their relationships,

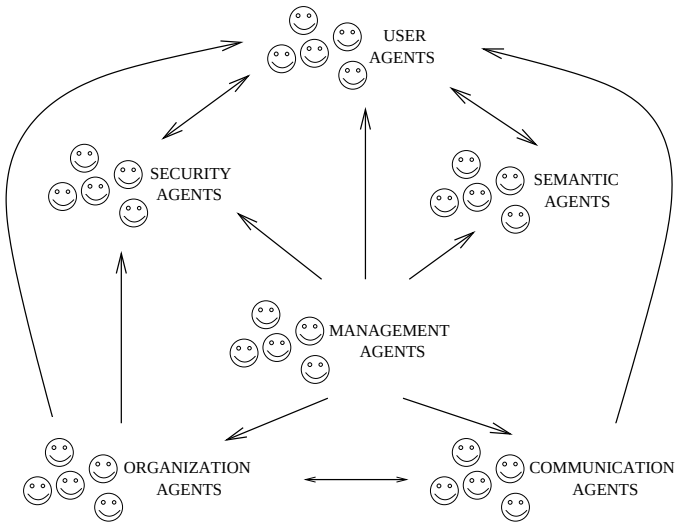

Fig. 4.7: Security agents and their relations in the federation

- *user agents* manage the system users and their rights on the global level of the federation,
- *organization agents* are responsible for the relations among the elements in the federation and the knowledge database of the security domains,
- *communication agents* are responsible for the communications on the level of local systems and on the global level and manage the security alerts generated during the occurrence of the security problems and access control problems
- and the *management agents* are responsible for the proper functioning of other types of the agents.

These agents have specific competences and they communicate with each other exchanging the information concerning different aspect of the security domains in the federation and the users of different cooperative systems in the federation.

4.1. Implementation of presented approach. A multi-agent system for access control of distributed information systems was developed to implement the presented approach. The proposed system is used to support the methodology of information project management. It manages and assigns dynamically an access to code repositories basing on current developer's status and basing on progress of works in particular phase of software development.

The main purpose of the system is to manage and monitor an access to code repositories of project groups that use a specific software creation methodology. The use of chosen methodology results in the following situations:

- project team assign dynamically the tasks to the developers according to their preferences and qualifications,
- members of project team should be focused only on one project,
- client can actively take part in process of product development and in consequence he has an access to particular resources of development company,
- reduction of costs in programming companies provokes that use of cloud computing and data stored in clouds becomes 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.

The exemplary situations presented above cause the application of intelligent system monitoring an access to resources of development company operating in distributed environment. The monitoring of access rights to resources should be realized with the use of intelligent agents from multi0agent systems assigning appropriate access rights to company resources.

5. Conclusions. 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. The logical security (i.e. access control) concerns the access control management based on the identification, authentication and authorization, counteracting the data modification or theft and wrong access to the data and programs. The traditional access control models are insufficient in distributed information systems, especially to express the policy of usage control. We need to have the unified security model to specify the general permissions, interdictions and obligations of an information system and to define the security rules dependent on an application context.

The presented paper focuses on the access control security in cooperative information systems. The proposed approach has to treat the cooperation of open and evaluative information systems and has to guarantee the respect of various local security policies on the global level. 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 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] S. Castano, M. Fugini, G. Martella and P. Samarati, *Database Security*, ACM Press, Addison-Wesley, 1994.
[2] R. S. Sandhu and Q. Munawer, *How to do Discretionary Access Control Using Roles*, Proceeding of 3rd ACM Workshop on Role-Based Access Control, 1998.
[3] D. Ferraiolo, R. S. Sandhu, S. Gavrila, D. R. Kuhn and R. Chandramouli, *Proposed NIST Role-Based Access Control*, ACM, Transactions on Information and Systems Security (TISSEC), Vol 4, No 3, 2001.
[4] G. Goncalves and A. Poniszewska-Maranda, *Role engineering: from design to evaluation of security schemas*, Journal of Systems and Software, Elsevier, Vol. 81, 2008.
[5] A. Poniszewska-Maranda, G. Goncalves and F. Hemery, *Representation of extended RBAC model using UML language*, LNCS, Proceedings of SOFSEM 2005: Theory and Practice of Computer Science, Springer-Verlag, 2005.
[6] B. Lampson, M. Abadi, M. Burrows and E. Wobber, *Authentication in Distributed Systems: Theory and Practice*, ACM Transactions on Computer Systems, 1992.
[7] M. Wooldridge, *An Introduction to MultiAgent Systems*, John Wiley & Sons, 2002.
[8] G. Weiss, *Multi-Agent Systems*, The MIT Press, 1999.
[9] M. Singh and M. Huhns, *Readings in Agents*, Morgan-Kaufmann Pub., 1997.
[10] J. Park and R. Sandhu, *The UCON ABC Usage Control Model*, ACM Transactions on Information and System Security, Vol 7, No 1, February 2004.
[11] G.-J. Ahn, *The RCL 2000 Language for Specifying Role-Based Authorization Constraints*, ACM Transactions on Information and Systems Security, 1999.
[12] G.-J. Ahn and R. Sandhu, *Role-based Authorization Constraints Specification*, ACM Transactions on Information and Systems Security, 2000.
[13] A. Ouksel and C. Naiman, *Coordinating Context Building in Heterogeneous Information Systems*, Journal of Intelligent Information Systems, No 3, Kluwer, Academic Publishers, 1994.
[14] A. Poniszewska-Maranda, *Conception Approach of Access Control in Heterogeneous Information Systems using UML*, Journal of Telecommunication Systems, Springer-Verlag, Vol 44, 2010.
[15] D. Basin, J. Doser and T. Lodderstedt, *Model Driven Security: from UML Models to Access Control Infrastructures*, ACM Transactions on Software Engineering and Methodology, Vol. 15, 2006.
[16] D. Basin, J. Doser and T. Lodderstedt, *Model Driven Security*, Engineering Theories of Software Intensive Systems, Springer, 2005.
[17] A. Poniszewska-Maranda, *Implementation of Access Control Model for Distributed Information Systems using Usage Control*, P. Bouvry et al. (Eds.): SIIS 2011, LNCS 7053, pages 54-67, Publisher: Springer, Heidelberg (2011).
[18] N. Antonopoulos, K. Koukoumpetsos and A. Shafarenko, *Access control for agent-based computing: a distributed approach*, Internet Research, Vol. 11 Issue: 1, pp.55 - 64 (2001).
[19] A. Seleznyov and S. Hailes, *Distributed Knowledge Management for Autonomous Access Control in Computer Networks*, International Conference on Information Technology: Information Assurance and Security, USA (2004).
[20] A. Seleznyov, M.O. Ahmed and S. Hailes, *ADAM: An Agent-based Middleware Architecture for Distributed Access Control*, The Twenty-Second International Multi-Conference on Applied Informatics: Artificial Intelligence and Applications, pages 200 - 205, Innsbruck, Austria (2004).
[21] L. Kagal, T. Finin and A. Joshi, *Trust-Based Security in Pervasive Computing Environments*, Computer (2001).
[22] V. Varadharajan, N. Kumar and Y. Mu, *Security Agent Based Distributed Authorization: An Approach*, Proceedings of the 21st National Information Systems Security Conference (NISSC), USA, pp. 315-328 (1998).
[23] E. Weippl, A. Schatten, S. Karim and A. Tjoa, *SemanticLIFE Collaboration: Security Requirements and solutions security aspects of semantic knowledge management*, Proceedings of Practical Aspects of Knowledge Management (PAKM), Austria (2004).
[24] P. Belsis, S. Gritzalis and Ch. Skourlas, *Security Enhanced Distributed Knowledge Management Architecture*, Proceedings

of I-KNOW, Austria (2005).

Edited by: Dana Petcu and Jose Luis Vazquez-Poletti
Received: November 1, 2011
Accepted: November 30, 2011