Role approach in access control development with the usage control concept

Aneta Poniszewska-Maranda

Institute of Information Technology, Technical University of Lodz, Poland
anetap@ics.p.lodz.pl

Abstract. Development of information technology, progress and increase of information flow have the impact on the development of enterprises and require rapid changes in their information systems. Very important stage of data protection in information system is the creation of high level model, satisfying the needs of system protection and security. This paper presents the role engineering aspects of access control for dynamic information systems. The objective is to find the approach of access control to ensure in perspective the global coherence of security rules for all components of an system.

1 Introduction

Development of information technology, progress and increase of information flow have the impact on the development of enterprises and require rapid changes in their information systems. The growth and complexity of functionalities that they currently should face causes that their design and realization become the difficult tasks and strategic for the enterprises at the same time. Data protection against improper disclosure or modification in the information system is an important issue of each security policy realized in the institution.

Role engineering for role-based access control is a process consisting of determination and definition of roles, permissions, security constraints and their relations. The company's roles can be regarded in two aspects - functional (reflects the main business functions of the company) and organizational (reflects the organization hierarchy of the company). Before a concrete access control model can be implemented, the role engineering process has to take place. The aspect of role engineering is connected close to the aspects of analysis and design of information systems. During our previous studies, we examined the design methods of the information systems and the design methods of information system security [7]. However, it is difficult to find the global method that takes into account both the design of system and its associated security scheme.

The process of roles identification and setting up in an organization is a complex task because very often the responsibilities of actors in an organization and their functions are few or badly formalized. Moreover, the role concept is an abstract approach. It does not correspond to particular physical being and therefore it is very difficult to give definitions that comprise the whole world.

To conclude, the research of roles in security schema needs a real engineering approach that provides a guide identify, specify and maintain them.

Many different works concerning the problem of role engineering were presented in the literature [13–20] and others. First of all, most of the paper tread the static aspects of access control domains. They do not take into consideration the problems of dynamic changes affecting the access control rules in modern information systems, particularly distributed information systems. Moreover, whereas everyone agrees that safety must be taken into account as quickly as possible, the proposed role engineering approaches are often disconnected from the design and the evolution of information system for which they are provided.

Coyne in [16] proposes to collect different user activities to define the candidate roles. Fernandez et al. in [14] propose to use the use cases to determines the role rights of system users but do not describe the constraints aspect and role-hierarchies. Epstein et al. [17] propose to express RBAC elements with UML diagrams but do not define the role engineering process. Roeckle et al. [20] propose a process-oriented approach for role engineering but only on meta level without details about role hierarchy, derivation of permissions and relation between some elements. Epstein and Sandhu [18] describe role engineering of role-permission assignment but not go into detail about constrains and role hierarchies in the process. Neumann and Strembeck propose in [9, 10] very interesting scenario-driven role engineering process for RBAC model. However this proposition does not take into consideration the dynamic aspects of access control.

This paper presents the engineering aspects in access control of dynamic information systems. The paper is composed as follows: section 2 presents the role concepts and usage concept in aspect of access control, section 3 gives the outline of approach based on these concepts (Usage Role Based Access Control - URBAC). Section 4 deals with the cooperation of two actors in role creation process of information system security, while section 5 shows the representation of URBAC approach using the UML concepts. Section 6 describes the process of roles production based on URBAC approach presenting the stage of security profile creation for the system's users.

2 Access control based on role and usage concepts

Development in area of information systems, especially modern, distributed information systems causes that traditional access control models are not sufficient and adequate for such systems in many cases. Some imperfections were found in domain of these security models [7]:

- traditional access control models give only the possibility to define the authorizations but do not provide the mechanisms to specify the obligations or conditions of access control,
- developers or security administrators can only pre-define the access rights that will be next granted to the subjects,
- decision about granting or not an access can be only made before the required access but not during the access,

– mutable attributes can be defined for subjects and for objects of a system.

These disadvantages and the needs of present information systems caused the creation of unified model that can encompass the use of traditional access control models and allow to define the dynamic rules of access control. Two access control concepts are chosen in our studies to develop the new approach for dynamic information systems: role concept and usage concept. The first one allows to represent the whole system organization in the complete, precise way while the second one allows to describe the usage control with authorizations, obligations, conditions, continuity (ongoing control) and mutability attributes.

Role-based access control approach [1] or its extensions [7, 8] requires the identification of roles in a system. The role is properly viewed as a semantic structure around which the access control policy is formulated. Each role realizes a specific task in the enterprise process and it contains many functions that the user can take. For each role it is possible to choose the necessary system functions. Thus, a role can be presented as a set of functions that this role can take and realize. Each function can have one or more permissions, and a function can be defined as a set or sequence of permissions. If an access to an object is required, then the necessary permissions can be assigned to the function to complete the desired job. Specific access rights are necessary to realize a role or a particular function of this role.

The usage control approach [4–6] is based on three sets of decision factors: authorizations, obligations and conditions that have to be evaluated for the usage decision. The obligations and conditions are added in the approach to resolve certain shortcomings characteristic for the traditional access control strategies. The most important properties of usage control that distinguish it from traditional access control concepts and trust management systems are the continuity of usage decisions and the mutability of attributes. Continuity means that usage control decision can be determined and enforced before an access (as in traditional access control models) and also during the ongoing period of the access. Mutability represents the mutability of subject and object attributes that 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.

3 Approach of role based access control with usage control

The complexity of actual organizations (i.e. matrix based organizational structure) has guided our proposition to extend the standard RBAC model by role management facilities that allow a finer distribution of responsibilities in an enterprise. On the other hand, the dynamism of actual information systems was the reason to use the concepts of Usage Control to develop the new implementation of access control models to support the management of information system security.

The proposed access control approach was based on two access control concepts: role concepts and usage concepts. It was named Usage Role-based Access Control (URBAC) [21, 22]. 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 detailed view of usage role-based access control approach is presented on figure 1.

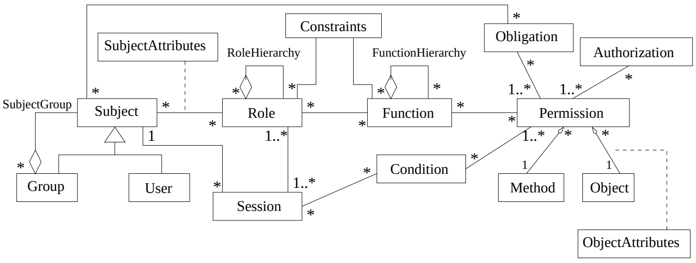

Fig. 1. Meta-model of URBAC model

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.

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 role can represent a 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.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 the roles and subjects is described by the association class **SubjectAttributes** that represents the additional subject attributes (i.e. subject properties) as in Usage Control. *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 a specific task associated with an enterprise process. A role can contain many functions **Function** that a user can apply. Consequently, 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, represented by the aggregation relation named *FunctionHierarchy*, which provides the hierarchical order of system functions.

Each function can perform one or more operations, a function needs to be associated with a set of related permissions **Permission**. 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. In order to access the data, stored in an object, a message has to be sent to this object. This message causes an execution of particular method **Method** on this object **Object**. Very often the constraints have to be defined in assignment process of permissions to the objects. Such constraints are represented by the authorizations and also by the obligations and/or conditions.

Authorization (A) is a logical predicate attached to a permission that determines the permission validity depending on the access rules, object attributes and subject attributes. **Obligation (B)** is a functional predicate that verifies the mandatory requirements, i.e. a function that a user has to perform before or during an access. **Conditions (C)** evaluate the current environmental or system status for the usage decision concerning the permission constraint.

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, Cst)* where*o* is an object, *m* is a method which can be executed on this object and *Cst* is a set of constraints which determine this permission. Taking into consideration a concept of authorization, obligation and condition, the set of constraints can take the following form $Cst = \{A,\ B,\ C\}$ and the permission can be presented as a function $p(o,\ m,\ \{A,\ B,\ C\})$. According to this, the permission is given to all instances of the object class except the contrary specification.

The **objects** are the entities that can be accessed or used by the users. The objects can be either privacy sensitive or privacy non-sensitive. 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 can not be specified in the object's class and they can be used for usage decision process. The examples of object attributes are security labels, ownerships or security classes. They can be also mutable or immutable as subject attributes do. The **constraints** can be defined for each main element of the model presented above (i.e user, group, subject, session, role, function, permission, object and method), and also for the relationships among the elements. The concept of constraints was described widely in the literature [3, 8, 11, 13]. It is possible to distinguish different types of constraints, static and dynamic, that can be attached to different model elements.

4 Two actors in role creation process of information system security

Two types of actors cooperate in the design and realization of security schema of an information system [8]: on the one hand it is application/system developer who knows its specification that should be realized and on the other hand it is security administrator who knows the general security rules and constraints that should be taken into consideration on the whole company level. We propose to partition the responsibilities between these two actors in the process of definition and implementation of security schema on access control level and to determine their cooperation in order to establish the global access control schema that fulfill the concepts of URBAC. This partition of responsibilities is presented in figure 2 and the responsibilities were divided into two stages: conception stage and exploitation stage.

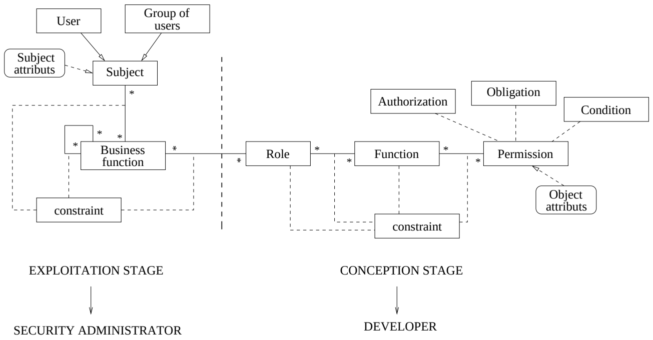

Fig. 2. Two actors in creation of information system security schema

Conception stage The realization process of information system or simple application is provoked by a client's request or in general by the client's needs to create a new information system or new application (i.e. to add a new component to the existing information system). Basing on the client's needs and requirements the application/system developer creates the logical model of application/system and next designs the project of this system that will be the base for its implementation. This model and next the project contain all the elements expressing the client's needs (i.e. needs of future users).

The application developer defines the elements of this application and its constraints corresponding to the client's specifications. These elements can be presented in a form adequate to the access control concepts - it will be the URBAC approach in our case. Therefore, the developer generates the sets of following elements: roles, functions, permissions and security constraints. These sets of elements should be presented to the security administrator in a useful and legible form. The duties of application developer basing on URBAC are:

- definition of permissions - identification of methods and objects on which these methods can be executed,
- definition of object attributes associated to certain objects according with access control rules,
- assignment of elements: permissions to functions and functions to roles,
- definition of security constraints associated to the elements of the application, i.e. authorizations, obligations and conditions on the permissions and standard constraints on roles, functions and their relationships.

Exploitation stage The exploitation stage is realized by the security administrator on the global level of information system. The security administrator defines the administration rules and company constraints according to the global security policy and application/system rules received from the developer. He should also check if these new security constraints remain in agreement with the security constraints defined for the elements of existing information system in order to guarantee the global coherence of the new information system.

The security administrator received from the developer the sets of elements in the form adequate to URBAC: set of of roles, set of functions, set of permissions and set of security constraints of the application. He uses these sets to mange the global security of the information system. First of all he defines the users' rights to use the particular applications. Two sets on company level are important to define the users' rights: persons working for the enterprise (i.e. users) and functions realized in the enterprise (i.e. business functions).

Security administrator is also responsible for the definition of security constraints for these assignments on global level with respect of defined security rules. These constraints concern first of all the following relations: user-businessFunction, businessFunction-businessFunction and businessFunction-role.

Therefore, the duties of security administrator are as follows:

- definition of users' rights basing on their responsibilities and their business functions in an organization - assignment of users to the roles of information system,
- organize the users in groups and definition the access control rights for the groups of users that realize for example the same business functions - assignment of groups to the roles of information system,
- definition of subject (i.e. user or group of users) attributes associated to certain users or groups of users that allows to determine the dynamic aspects of security constraints,
- definition of security constraints for the relationships between users and roles or group of users and roles.

The second important task of security administrator is the management of set of applications assuring the global coherence of the whole system on access control level. This assurance is exceptionally important in case of addition of new application in an information system when new access control elements appear both on local level and on global level.

5 Representation of URBAC using the UML concepts

Unified Modeling Language (UML) is now a standard language for analysis and design of information systems [2]. It is used almost all over the world in software engineering field for object-oriented analysis and design. It has a set of different models and diagrams that allow to present the system project from different points of view showing the whole system together with its components. Some elements and concepts of UML can be used to implement the URBAC approach, especially during the design stage of information system and its associated security schema based on URBAC.

From this reason, UML was chosen to be used in role engineering process to implement and realize the URBAC approach. To accomplish this, the concepts of UML and URBAC should firstly be joined. Two types of UML diagrams have been chosen to provide the URBAC: use case diagram and interaction diagram. The use case diagram presents the system's functions from the user point of view. It define the system's behavior without functioning details. The interaction diagram describes the behavior of one use case [2]. It represents the objects and messages exchanged during the use case processing.

The relationships between UML concepts and concepts of usage role-based access control are as follows (Fig. 3):

- role (R) from access control model can be presented as an UML actor,
- function (F) from URBAC can be represented by an UML use case,
- each actor from use case diagram can be in interaction with a set of use cases and these relations specify the relations of R-F type (between roles and functions),
- methods executed in sequence diagrams and also in other UML diagrams can represent the methods of URBAC,
- objects that occur in UML diagrams, e.g. sequence diagram, communication diagram, can be attached to the object concept of access control model,
- permissions (P) of URBAC can be found examining the sequence diagram(s) describing the particular use case,
- use case diagram offers four types of relations between its elements:
    - communication relation between an actor and a use case that represents the relation between a role and a function, i.e. R-F relation,
    - generalization relation between actors, representing the inheritance relation between roles (R-R relation),
    - two types of relations between use cases represent the inheritance relations between functions of URBAC, i.e. F-F relations
    - subject attributes (e.g. user attributes) from URBAC can be represented by the set of attributes defined for an instance of actor class of UML,
    - concept of object attributes from URBAC can be attached to set of attributes defined for the objects in its class specification.

The concept of constraints of URBAC approach corresponds directly to the constraint concept existing in UML. The security constraints of URBAC can be defined for different elements and for relations between these elements. These constraints can be presented on UML diagrams corresponding to types and locations of elements for which these constraints will be defined.

The authorization is a constraint attached to a permission that determines the permission validity depending on defined access rules. It can be represented by the UML constraint defined for the method's execution in sequence diagram.

The obligation is a constraint defined on a permission but it concerns also the subject (e.g. a user) - subject should fulfill the obligation executing a function before or during an access. This type of constraints can be presented as UML

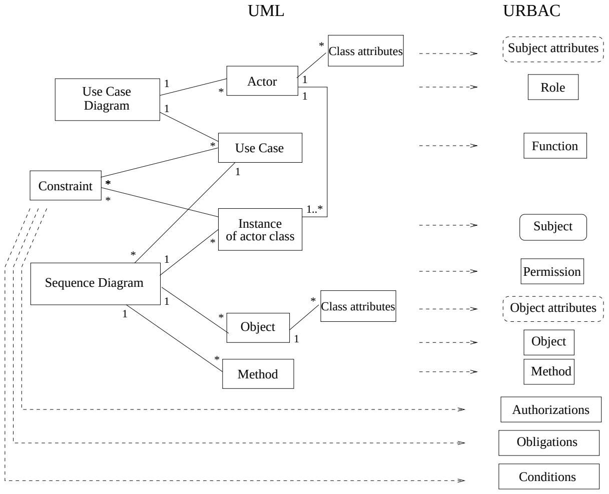

Fig. 3. Relationships between concepts of URBAC and UML concepts

constraint in sequence diagram (as pre-condition or an invariant), especially from version 2.0 of UML that provide the combinator fragments in sequence diagrams (e.g. ”alt” or ”opt”) allowing the definition of constraint conditions.

The condition is a constraint also defined on a permission but it concerns the session element. It defines current environmental or system status and states during the user session that can be used for the usage decision. The conditions can be also represented by UML constraints defined in sequence diagrams (mainly as an invariants).

Remaining types of constraints represent the constraints defined for the roles, functions and their relations. Such constraints are represented by the UML constraints defined for actors, use cases and their relations on use case diagrams or sometimes on sequence diagrams.

6 Production of roles based on URBAC approach

The process of role production is based on the connections between UML and URBAC, described in section 5. It can be automatic or partially automatic. Two types of UML diagrams were used to realize this process: use case diagrams, where roles and functions of a system are defined and sequence diagrams, where permissions are assigned to the rights of method executions realized in framework of each use case. These two types of diagrams should be examined to identify the roles of URBAC, the functions that are used by these roles to interact with the information system, the permissions needed to realize these functions and the constraints that determine the possible rights.

To obtain these elements of URBAC, first the rules for creation of set of roles has to be defined.

Each subject (i.e. user or group of users) in an information system is assigned to a security profile (i.e. user profile) which is defined by the set of roles that can be played by him. A security profile is defined by a pair *(s, listRoles(s))*: *s* is a subject, *listRoles(s)* is a set of roles assigned to this subject. Taking into consideration the concept of user, such profile can be defined as follows: *(u, listRoles(u))*, where *u* is a user, *listRoles(u)*.

The process of creation of user profiles, i.e. production of set of roles, in information system with the use of UML diagrams contains two stages [7]:

Determination of a function with assigned permissions

As it was shown in section 5, a use case of UML meta-model corresponds to a function of URBAC model. Use cases define the system functionality or in other words the interactions and needs of system's users that cooperate with the system. Each use case should be defined by its scenario that defines the specification of the use case interaction in form of sequence of actions performed on the system's objects. It allows the definition of set of privileges for execution of different actions on the objects. Therefore, in order to identify the permissions assigned to a function it is necessary to start from the sequence diagram corresponding to the function.

Each message $msg\,(o_1, o_2, m)$ in the interaction sent by object o_1 to object o_2 to execute method m on object o_2 should have a suitable permission assigned. This permission corresponds to the execution of method m on object o_2. On the addition of security constraints, the definition of the message is as follows: $msg\,(o_1, o_2, m, cst)$ where cst represents a set of constraints - authorizations, obligations and conditions: $cst(p) = A\,(p) \cup B\,(p) \cup C\,(p)$

Consequently, the set of permissions for interaction i is defined as follows:

$$P\,(i) = \{p \mid \varphi\,(msg\,(o_1, o_2, m, cst)) = p\,(m, o_2) \wedge cst\,(p) = true\}$$

where φ is a function that assigns a permission to message msg, o_1 is an actor or class instance that can execute method m on object o_2 and cst is a set of constraints

$$cst(p) = A\,(p_{m,o_2}) \cup B\,(p_{m,o_2}) \cup C\,(p_{m,o_2})$$

.

Sequence diagram in UML meta-model is defined by the set of interactions. Therefore, the set of permissions determined for sequence diagram d_S and described by the set of interactions D_S is as follows:

$$P\,(d_S) = \bigcup_{i\,\in\,D_S} P\,(i)$$

The use case μ described by set M of interaction diagrams d_i has the following set of permissions assigned to it:

$$P\,(\mu) = \bigcup_{d_i\,\in\,M} P\,(d_i)$$

The set $P\left(\mu\right)$ represents the set of permissions assigned to the function specified by the use case μ.

Determination of a role with assigned functions

The use case diagram presents the system's functionality from the point of view of the actors. It is possible to find the set of use cases (i.e. URBAC functions) for each actor (i.e. URBAC role) examining this type of UML diagrams. Therefore, the determination of a role with the set of functions assigned to it will be realized by examining the relationships between the actors and use cases on the use case diagram. The use case diagram ucd_i contains the use cases (i.e. functions) attached to chosen actors (i.e. roles). The set of functions assigned to role r_j, described by one use case diagram, is defined by the functions that are in direct or indirect relations with this role (i.e. by the inheritance relations between the functions) on this diagram:

$$F\left(r_{ucd_i}\right) = \{f \mid f = uc\ ,\ uc \in ucd_i \wedge \left(r_{ucd_i}, f\right)\ \in R - F\}$$

$$\cup \{f' \mid f' = uc\ ,\ uc \in ucd_i \wedge\ \left(\left(f, f'\right) \in F - F \wedge \left(r_{ucd_i}, f\right) \in R - F\right)\}$$

The set of functions of role r_j is defined by the union of use cases assigned to this role in all use case diagrams describing the whole system application D_{uc}:

$$F\left(r_j\right) = \bigcup_{ucd_i\ \in\ D_{uc}} F\left(r_{ucd_i}\right)$$

In order to define the security profiles for system users or groups of users, the set of roles should be assigned to subject profiles (i.e. user profiles). This task is realized by security administrator during the exploitation stage who has to take into consideration the security constraints defined on the global level and the subject attributes defined for the subjects that determine the access control rights of particular system users.

7 Conclusion

The concepts of access control approach presented in the paper were used to define the process of role engineering for creation of security profiles for users of information system. The paper presents the representation of URBAC using the UML concepts, the process of roles production based on URBAC, the stages of creation of user profiles. The process of role production is a very important stage in definition of logical security policy of an information system. It can be realized by two actors: application developer and security administrator who cooperate with each other to guarantee the global coherence on access control level.

The aspects of presented approach are implemented on software platform that provides with the software tool to manage the logical security of company information system from the point of view of application developer and from the point of view of security administrator. Our next research is concentrated on aspects of security constraints for URBAC approach and on the algorithm to maintain the coherence of URABC scheme during the addition of new application.

References

1. D. Ferraiolo, R. S. Sandhu, S. Gavrila, D. R. Kuhn and R. Chandramouli, Proposed NIST Role-Based Access control, ACM TISSEC, 2001
2. OMG Unified Modeling Language (OMG UML): Superstructure. February 2009. Version 2.2, The Object Management Group
3. J. Park, X. Zhang and R. Sandhu, Attribute Mutability in Usage Control, 18th IFIP WG 11.3 Working Conference on Data and Applications Security, 2004
4. A. Lazouski, F. Martinelli and P. Mori, Usage control in computer security: A survey, Computer Science Review Vol. 4, Issue 2, May 2010, pp 81-99
5. A. Pretschner , M. Hilty and D. Basin, Distributed usage control, Communications of the ACM, Vol. 49, No 9, September 2006
6. X. Zhang, F. Parisi-Presicce, R. Sandhu and J. Park, Formal Model and Policy Specification of Usage Control, ACM TISSEC, 8(4): 351-387, 2005
7. G. Goncalves and A. Poniszewska-Maranda, Role engineering: from design to evaluation of security schemas, Journal of Systems and Software, Elsevier, No 8, Vol81
8. A. Poniszewska-Maranda, Conception Approach of Access Control in Heterogeneous Information Systems using UML, Journal of Telecommunication Systems, Springer-Verlag Heidelberg, Vol. 45, No 2-3, pages 177-190, October 2010
9. G. Neumann and M. Strembeck, A Scenario-driven Role Engineering Process for Functional RBAC Roles, In: Proc. of 7th ACM SACMAT, USA, June 2002
10. M. Strembeck, Scenario-Driven Role Engineering, In: IEEE Security & Privacy, Vol. 8, No. 1, January/February 2010
11. M. Strembeck and G. Neumann, An Integrated Approach to Engineer and Enforce Context Constraints in RBAC Environments, ACM TISSEC vol. 7, no 3, 2004
12. E.J. Coyne and J.M. Davis, Role Engineering for Enterprise Security Management, Artech House, 2008
13. E. Bertino, E. Ferrari and V. Atluri, The Specification and Enforcement of Authorization Constraints in Workflow Management Systems. ACM TISSEC, 2(1)
14. E. B. Fernandez and J. C. Hawkins, Determining Role Rights from Use Cases. In Proc. of 2nd ACM Workshop on Role-Based Access Control (RBAC), USA, 1997
15. D. Basin, J. Doser, and T. Lodderstedt, Model driven security: From UML models to access control infrastructures. ACM Transactions on Software Engineering Methodology, 15:3991, 2006
16. E.J. Coyne, Role engineering, In Proc. of the ACM Workshop on role-Based Access Control, 1996
17. P. Epstein and R. Sandhu, Towards a UML Based Approach to Role Engineering, In Proc. of the ACM Workshop on role-Based Access Control, 1999
18. P. Epstein and R. Sandhu, Engineering of Role-Permission Assignment to Role Engineering, In Proc. of 17th ACSAC, 2001
19. E.J. Coyne and J.M. Davis, Role Engineering for Enterprise Security Management, Artech House, 2008
20. H. Roeckle, G. Schimpf and R. Weidinger, Process-oriented approach for role-finding to implement role-based security administration in a large industrial organization, In Proc. of ACM Workshop on role-Based Access Control, 2000
21. A. Poniszewska-Maranda, Implementation of Access Control Model for Distributed Information Systems using Usage Control, SIIS 2011, LNCS 7053, (2011)
22. A. Poniszewska-Maranda, Administration of access control in information systems using URBAC model, Journal of Applied Computer Science, Vol. 19, No 2, 2011