JOURNAL OF APPLIED
COMPUTER SCIENCE
Vol. 19 No. 2 (2011), pp. 89-109

Administration of Access Control in Information Systems Using URBAC Model

Aneta Poniszewska-Marańda

Institute of Information Technology
Technical University of Lodz, Poland
Wólczańska 215, 90-924 Łódź
Aneta.Poniszewska@p.lodz.pl

Abstract. *Since the value of information is constantly growing more and more businesses are in need for information system to aid them with information gathering and processing. The most important issue that arises here is how to ensure safety of this data that may be held on servers, personal computers or PDAs. This is where access control comes in. The main role of access control is to ensure that no unauthorized user will be able to gain access to resources and be able to copy or modify them.*

The paper deals with the process of access control administration in information systems with the use of usage role-based control approach. The presented process is based on the role engineering concept that includes the creation of security schema of access control divided between two actors - application/system developer and security administrator. They realize their tasks during two main phases that allow to define the complete access control schema for information systems of an organization and to ensure the access control coherence on the global level.

Keywords: *access control of information systems, usage role-based access control, role engineering, administration of access control.*

1. Introduction

Since the value of information is constantly growing more and more businesses are in need for information system to aid them with information gathering and processing. The most important issue that arises here is how to ensure safety of this data that may be held on servers, personal computers or PDAs. This is where access control comes in. The main role of access control is to ensure that no unauthorized user will be able to gain access to resources (such as files or directories that may hold information) and be able to copy or modify them. It ensures proper protection for information held within the information system so that no individuals or companies will be able to steal or use it. With the fast progress of technologies the security issue becomes the most important issue when any sensitive data is concerned.

There are currently many models of access control that suggest rules of guarding data from unwanted guests that we may choose from. Every company should analyze their needs and choose best suitable model for their needs to ensure safety and confidentiality of sensitive data. It is possible to use a model that ensures central administration of who can and who can't access system's files, or we may leave those decisions to the owners of files. It can be also possible to choose whether this access rules are to be modified statically or dynamically (e.g. during program execution). Every model has its advantages and disadvantages which is why every company wanting to implement them must firstly attentively research each model and study if it will fit their needs.

Development in information technology requires also the additional features for access control domain. Data protection against improper disclosure or modification in the information system is the important issue of each security policy realized in the institution. Access control policies, strategies or models should also be changed to manage the new security problems.

On the other side, distributed information systems or federation of information systems provide the access of many different users to huge amount of data, sometimes stored in different locations and secured by different strategies, security policies and models or inner enterprise rules. These users have different rights to the data according to their business or security profiles that depend on their organization positions, actual locations and many other conditions. The system data is transferred between the particular nodes of distributed system. It is important to protect the information against non-controlled utilization and control the usage and diffusion of the information that gives the possibility to specify how it can be

used and specify the utilization constraints. In order to meet these requirements in modern access control, a new access control approach, named Usage Role-based Access Control (URBAC) was proposed. Compared to the traditional models, URBAC approach assures the usage control in data accessing that is very important, especially in distributed information systems, and the organization of the access control strategies well-described in RBAC (Role-Based Access Control) model [1, 2] or its extensions.

The research of roles to set up in an organization is a complex task because very often the functions of actors in an organization are few or badly formalized. Moreover, the role concept is an abstract approach – it does not correspond to a particular physical reality and therefore it is very difficult to give definitions that comprise the whole world. The research of roles needs, like many other scientific domains, a real engineering approach that provides a guide to comprehend and maintain them.

The paper deals with the process of access control administration in information systems with the use of usage role-based control approach. The presented process is based on the role engineering concept that includes the creation of security schema of access control divided between two actors - application/system developer and security administrator. They realize their tasks during two main phases that allow to define the complete access control schema for information systems of an organization and to ensure the access control coherence on the global level.

The paper is structured as follows: section 2 presents the usage control concept and Usage Control (UCON) model, section 3 deals with usage role-based access control approach for dynamic, complex information systems while section 4 describes the role engineering process with the use of usage concept. Section5 presents the implementation example of usage role-based access control approach and section 6 describes the administration tool to manage the access control in information systems with the use of usage concept.

2. Usage control and UCON model

The traditional approach to usage control encompassed three main areas: access control (closed-environment authorization systems that base on identity and attributes of a known user), trust management (authorization for unknown users, based on their properties and capability) and digital rights management (control over the usage of resources that have already been disseminated). These three

components were studied separately and only Usage Control (UCON) approach encompassed all these three elements of logical security [3, 4].

Usage decision is not only based on existence of subject attributes and object attributes. It is based on fulfillment of required actions by the users during accessing the data or simply during connecting with a system. It is also based on certain environmental or system status that can be static and constant during some period of time or can change dynamically. Such decision factors were named obligations and conditions and represent the dynamic aspects of access control in information systems. They represent the type of constraints that determine the possibility to allow an access to data basing on the factors that can change dynamically - they can be evaluated before or during the access request. Moreover, the usage of an object can demand some modifications in subject attributes or object attributes before an access, during an access or even after the access.

The UCON concept can be divided taking into consideration its scope. The distinction can be done based on control domain: the system might include a Server-Side Reference Monitor (SRM), where a central entity manages usage control (this approach addresses primarily the issues of access control and trust management), a Client-Side Reference Monitor (CRM), when a client-side application controls access to resources (this approach addresses primarily the issue of DRM), or both [5, 6].

Definition of rights in UCON model is fairly similar to that arising from the traditional approaches: rights are privileges a subject (e.g. user) holds on a certain object (e.g. files the user wants to access), represented by a usage function enabling access to objects [3, 4]. The core idea of UCON model is the possibility of detecting certain rights in dynamic way. Certain usage decision functions are applied to determine the existence of a right whenever a subject wishes to access the object. The result of these functions depends on subject's and object's attributes. Also, these attributes can be altered as a result of executing a right, which then can have an influence on future usage decisions (e.g. the user may have the possibility to access a file five times only, or access only one file of a given type). This ensures a fully dynamic approach to usage control [3].

The UCON model [5, 6, 7] consists of three core components, i.e. subjects, objects, and rights, and three additional components that are mainly involved in authorization process, i.e. authorization, conditions, and obligations. Subjects and objects can have in addition the attributes (i.e. subject attributes and object attributes) that define the additional characteristic features of subjects or objects,

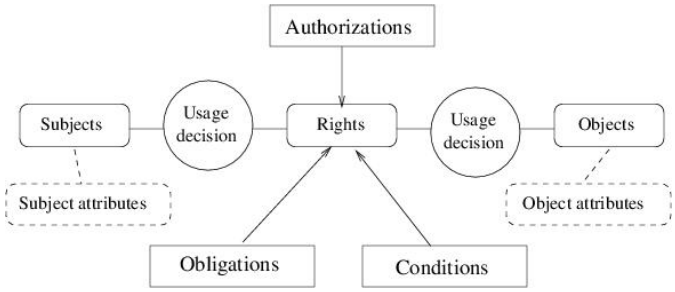

Figure 1. UCON model

which can be used in decision process of usage control. An attribute is regarded as a variable with a value assigned to it in each system state (Fig. 1).

3. Usage role-based access control approach for dynamic, complex information systems

Distributed information systems 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 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 proposed access control approach was based on two access control models: extended RBAC model [8] and UCON model [5, 6]. It was named Usage Role-based Access Control (URBAC) (Fig.2).

The core part of URBAC [9] 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. It can be presented as an abstract

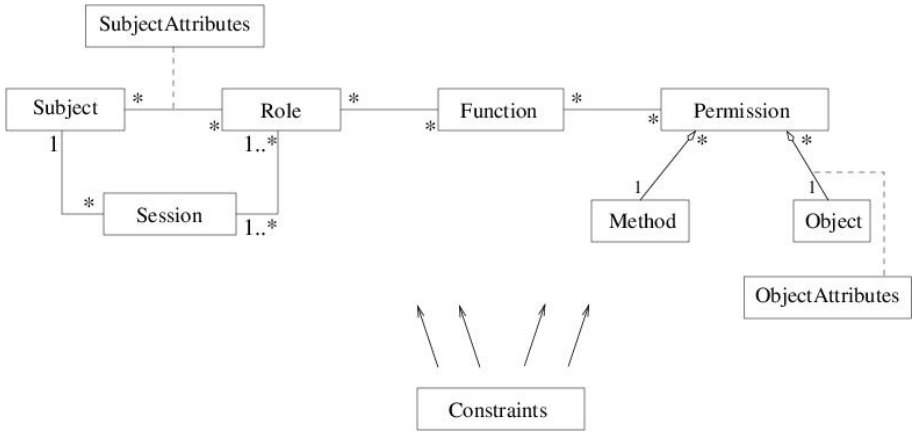

Figure 2. General view of Usage Role-based Access Control model

type, so it cannot have direct instances - each subject is either a user or a group of users. The aggregation relation *SubjectGroup* that represents an ordering relation in the set of all system subjects can assign subjects to the groups.

The *Session* element represents the period of time during which the user is logged in a system and can execute its access rights. Session is assigned directly to the Subject. On the other hand a session is connected with the roles and this association represents the roles that can be activated during one session.

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 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. The Usage Control introduced two types of attributes: immutable attribute that cannot be

changed by the user activity (only by administrative actions) and mutable attribute that can be modified as a side effect of subjects' access to the objects.

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. 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 on this object. Very often the constraints have to be defined in assignment process of permissions to the object.

Such constraints are represented by the authorizations and also in some cases 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 a usage. *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 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 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 [10, 8]. It is possible to distinguish different types of constraints, static and dynamic that can be attached to different model elements. The URBAC model 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 [11].
- 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 [10, 8].
- Constraints on relationships between the model elements [8, 12].

The detailed model of usage role-based access control approach is presented on figure 3.

4. Role engineering with the use of usage concept

The aspect of role engineering in access control domain is connected close to the aspects of analysis and design of information systems. However, it is difficult to find the global method that takes into account both the design of information system and its associated security scheme [13, 14].

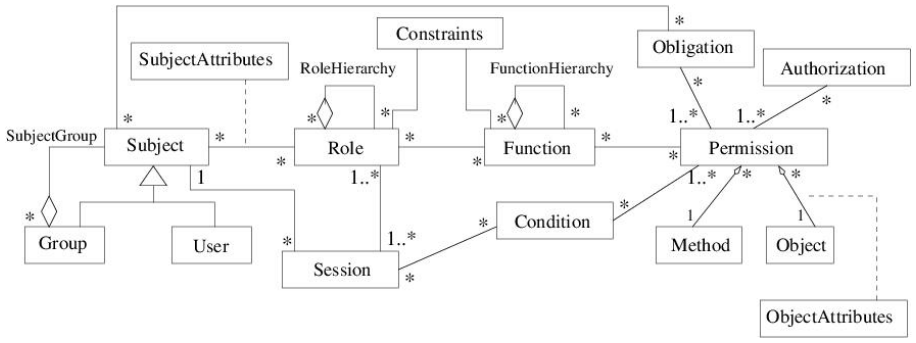

Figure 3. Complete structure of URBAC approach

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 and it does not correspond to a particular physical being. The identification and determination of roles in security schema needs real engineering approach that provides the appropriate management of roles in security schema of information system [14, 15].

Two types of actors cooperate in the design and realization of security schema of an information system: 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 security constraints that should be taken into consideration on the global company level.

We propose the partition of 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 model. Their responsibilities were divided into two stages: conception stage and exploitation stage (Fig.4).

The application developer realizes the conception stage on the local level of information system. He defines the elements of the application and its constraints corresponding to the client's specifications. The developer generates the sets of following elements: roles, functions, permissions and security constraints. These sets should be presented to the security administrator in a useful and legible form. The duties of application developer basing on URBAC model can be determined as follows:

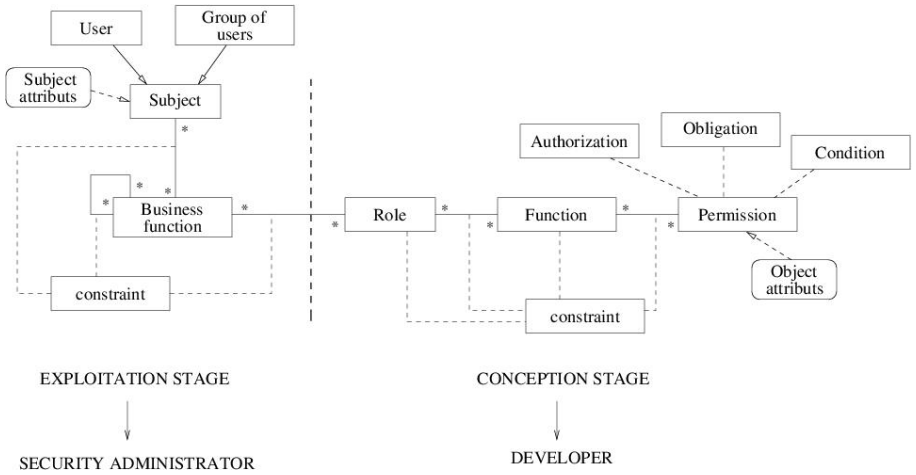

Figure 4. Partition of responsibilities in role engineering process

- 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.

The security administrator realizes the exploitation stage on the global level of information system. He defines the administration rules and company constraints according to 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 whole information system.

The duties of the 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,

- organization of users in groups and definition of access control rights for the groups of users that realize for example the same business functions – assignment of groups to the roles,
- definition of subject (i.e. user or group of users) attributes associated to certain users or groups of users that allow to determine the dynamic aspects of security constraints,
- definition of security constraints for the relationships between users and roles or groups of users and roles.

UML (Unified Modeling Language) [16, 17] can be used in role engineering process to implement and realize the URBAC model. To accomplish this, the concepts of UML and URBAC model should firstly be joined. Two types of UML diagrams have been chosen to provide the URBAC model: use case diagram and interaction diagram. The relationships between UML concepts and concepts of usage role-based access control model are as follows (Fig. 5):

- role (R) from access control model can be presented as an UML actor,
- function (F) from URBAC model 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 model,
- 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 model 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 role and function, i.e. R-F relation,
  - generalization relation between actors, representing the inheritance relation between roles (R-R relation),

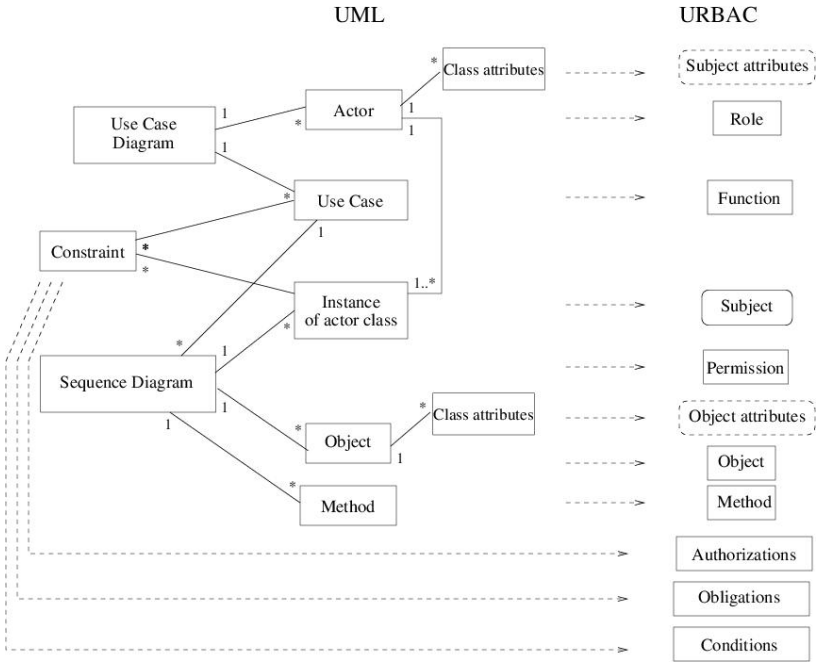

Figure 5. Elements of URBAC model and corresponding UML concepts

  - two types of relations between use cases represent the inheritance relations between functions of URBAC model, i.e. F-F relations

- subject attributes (e.g. user attributes) from URBAC model can be represented by the set of attributes defined for an instance of actor class of UML,
- concept of object attributes from URBAC model can be attached to set of attributes defined for the objects in its class specification.

The concept of constraints of the URBAC model corresponds directly to the constraint concept existing in UML. The security constraints of URBAC model 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 process of role production can be automatic or partially automatic and is based on the connections between UML and URBAC model. This process is

realized with the use of use case diagrams, where system roles and functions are defined and with the use of sequence diagrams, where permissions are assigned to the rights of execution of methods realized in each use case. The diagrams of these two types should be examined to identify the roles of URBAC model, the functions that are used by these roles to interact with the information system and the permissions needed to realize these functions.

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

- determination of a set of privileges (i.e. permissions) for a use case in order to define a function and
- determination of a set of use cases (i.e. functions) for an actor in order to define a role.

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

5. Exemplary implementation of URBAC components

The URBAC model is an abstract model, as well as RBAC model or UCON model, and it represents the union of role based control approach and usage control approach in security of information systems. It contains the set of logical security concepts that can be used in practice to define the security rules for particular information systems of enterprises. We decided to study how the URBAC model can be useful in creation of security schema of an application of information system. The main advantage of this model is the possibility to manage all complex and dynamic aspects of access control in current applications or information systems.

The exemplary application was created to check how the elements and principles of URBAC model could be put into practice. This application is a web service offering an on-line music store. It will allow customers to buy, download and upload music files to the system, putting a price on them, and availing the users with a possibility to buy the credits. The service also allows browsing through the files

on offer to find those the users wish to have. The users can have the access to different levels of the application's functionality based on membership type. For this reason, we can distinguish the following types of roles of URBAC model:

- *Guests* – new visitors with no account, only allowed to view the content of the web page restricted to the home page, About page, and the list of music files available,
- *Registered users* - users whose accounts have been activated, potentially eligible to perform certain transactions, can have different privileges depending on the type of membership:
  - *Regular users* are only able to upload, buy and download files, and have to watch advertisements upon downloading,
  - *Premium users* do not have this limitation and additionally have access to the transaction of trading,
- *Administrator* - the person that has access to all application data via a separate administration area.

The functions of URBAC approach are represented by the functionality of application. The basic functions contains (Fig. 6):

- registration of a new user,
- buying of first credits in order to start trading with other site members,
- adding of user's own file to the site,
- buying of selling the tracks,
- editing or deleting of files by a user but only these files of which he/she is the owner,
- viewing of user's personal site with information about number of actual credits, list of his uploaded files, list of downloaded files, accepted transfers and transfers sent,
- buying of additional rights which allows user to omit the advertisements and waiting for downloading files as well as access the trade operation.

Figure 6. User personal site of the on-line music store

The created application includes a big amount of authorizations and condition rules as well as a number of obligations examined whenever users wish to access a particular file. These components work in both a static and dynamic way.

The first URBAC set of elements is subjects that are represented by users whose properties are collected in the User model. This is an identifiee subject type, as it represents a certain set of credentials. However, it can also represent a provider subject if the User owns a file uploaded onto the server. The concept of consumer rights separated from identifiee rights is not utilized, since any cooperation of the application with a custom player for the files did not been provided, nor any DRM policies did not been introduced to be imposed on the downloaded files.

The model has certain attributes, which can change as a result of executing a certain right. For example the number of credits will be decreased when the user buys a file. Also, the User's group can change once the decision to upgrade to a premium user has been made. The User also contains a field indicating whether the most recent terms and conditions introduced have already been agreed on. This is an example of checking global obligations before any transaction is committed. Some attributes, however, are static, and can only be changed by means of an administrative action (e.g. user name and the password).

The Group can also be considered as a subject, and it has its own attributes. Those are boolean values, representing the accessibility of certain transaction, as

well as a short value, representing the fee users have to pay to become members of a certain group. The default groups are: guests, regular users and premium users. However, when authorization is being checked, we do not rely on the type of the group, but on the group's attributes. These approach is more flexible - if need be, the administrator may define new groups with different transactions accessible and different membership fees. This ensures a finer-grain control over usage policies.

The MusicFile should be perceived as an object as far as the URBAC model is concerned. It has some static attributes, such as the owner, which can be modified only by the system administrator. In contrast, the attributes as the price, name, artist and the number of the number of available downloads after it is bought can be considered dynamic, since they can be modified by the owner by means of executing modification option, which is also a transaction.

The authorization rules are defined as follows: the primary rule is to check whether the user is not a guest. Unregistered user only has access to viewing the list of files, without the possibility of buying or downloading them. Next, the application checks if the User's group can execute a certain transaction on any file. Then, it is examined whether the User has enough credits (in case of buy and upgrade operations) or any downloads left available on the file (in case of download transaction), or whether the User is the owner of the file (in case of edit and remove transactions). As we can see, the authorization process relies heavily on subject's attributes, and we utilize the pre-Authorizations URBAC variant.

Even before a transaction is started, conditions are tested. The application enables the system administrator to apply a very flexible approach to defining conditions. This will be better understood if we analyze the *ConditionRule* model. The first field of the model defines the type of transaction that this rule will be applicable to. It is possible to set the value of this attribute to ALL, indicating that the rule will be applied to any transaction. The construction of the *ConditionRule* class results in an ability to define rules of the form 'the value of variable a must be in the range between b and c' or 'variable a must contain c', connected into longer predicates using logical operators.

Finally, the obligations the user has before exercising the rights on a file include displaying an advertisement before the download starts. Checking whether the obligation should be applied depends both on subject's and object's attributes: whether to display an advertisement depends on whether an advertisement has been assigned to the file. Similarly, some user groups may have the privilege to omit the advertisement.

The created application utilizes a wide variety of elements derived from the concept of URBAC, implemented in both a static and dynamic way. The presented solution provides a general idea on how the abstract model can be employed in a real application.

6. Administration tool for access control in information systems using usage concept

The previous section presents the exemplary application from the point of view of URBAC elements. That application is manage with the use of URABC approach. But, of course, we need the administration tool that will helps us to manage the logical security, i.e. access control, usage control, of such application, set of application or whole information system.

To obtain this goal, the administration tool to manage the access control of information systems from the point of view of security administrator was created. This tool was based on the URBAC approach - it allows to manage the access control of an application of an information systems with the use of URBAC concepts. The security administrator who is responsible for the information system security obtained a tool that facilitate the management of one of security aspects, namely the management of access control of users to data stored in a system. He has the knowledge of general security rules, privileges and constraints that should be respected on the enterprise level. However, the application/system developer who knows the specifications of information system that need to be realized can also use the tool to define the access control rules for the application's elements on the conception level.

The presented tool is a part of the platform for access control management in distributed information system based on URBAC approach. The important stage of platform functionality is the process of role engineering. This stage is based on the concepts of URABC approach and uses the UML to design the logical model of an application or a system and uses the XML language to exchange the information came from different UML diagrams, assuring the independent formalism. The first stage of the role engineering process is the creation of roles and definition of set of roles. This stage, presented in section 4, is realized basing on the concepts of URBAC approach and the UML concepts.

The second purpose of the platform is to make possible the cooperation between application/system developer and security administrator to realize and inte-

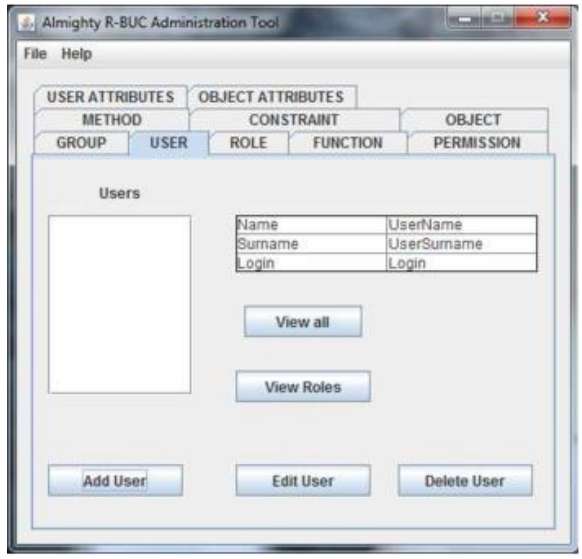

Figure 7. Administration tool - management of users

grate the presented concepts on global level of access control. The result of such cooperation should be the validation of activities realized by the developer and by the administrator to assure the global coherence on access control level in the information system.

Exemplary views of the administration tool from the security administrator point of view are presented on the figure 7 and 8. They show the possibility to manage the access control basing on the URBAC approach. The administrator can manage the system users, defining the user profiles. He can add new users to the system, change the names, passwords and define the roles of existing users. While adding new user, administrator can assign him multiple roles (Fig. 7). There is also the possibility of deleting some of them from the list.

In another window the security administrator gets the possibility of managing the roles defined in a system that he obtained in a form of list. This list of roles can be modified by adding new roles, modifying already existing ones or removing particular roles (Fig. 8). He can assign the functions to each role to define the role specification.

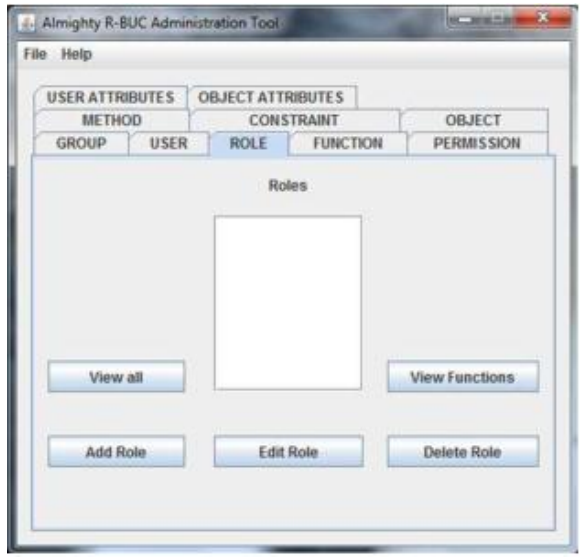

Figure 8. Administration tool - management of roles

7. Conclusion

The traditional access control models are insufficient in distributed, complex and dynamic information systems, especially to express the policy of usage control. We need to have the security model to specify the general permissions, interdictions and obligations of an information system and to define the security rules dependent on application context.

The concepts of presented access control model were used to define the process of role engineering for creation of security profiles for users of information system. The paper presented the representation of URBAC model using the UML concepts and the process of roles production based on URBAC model.

It seems that the URBAC approach can be used to support the security of dynamic information systems. The dynamic change of security policy can be translated by the change of the values of subject attributes or object attributes. The modification of an attribute can be realized before the information access, during the information access or at the end of the access.

The discussed approach requires collaboration of the system developer and the global security administrator since both of them define security elements and

constraints for the system, though each one on a different level. The issues of their activities have to be confronted and verified. The confrontation of two viewpoints consists of the verification that the developer's work does not cause incoherences in the global security of the information system through identification of the problems that may appear between these two levels.

References

[1] Sandhu, R. S. and Samarati, P., *Access Control: Principles and Practice*, IEEE Communication, Vol. 32, No. 9, 1994, pp. 40–48.

[2] 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, 2001.

[3] Sandhu, R. and Bhamidipati, V., *The ASCAA Principles for Next-Generation Role-Based Access Control*, In: 3rd International Conference on Availability, Reliability and Security (ARES), Spain,, 2008.

[4] Pretschner, A., Hilty, M., and Basin, D., *Distributed usage control*, Communications of the ACM, Vol. 49, No. 9, 2006, pp. 39—-44.

[5] Park, J., Zhang, X., and Sandhu, R., *Attribute Mutability in Usage Control*, In: 18th Annual IFIP WG 11.3 Working Conference on Data and Applications Security, 2004.

[6] Park, J. and Sandhu, R., *The UCON ABC Usage Control Model*, ACM Transactions on Information and System Security, , No. 7, 2004.

[7] X. Zhang, F. Parisi-Presicce, R. S. and Park, J., *Formal Model and Policy Specification of Usage Control*, ACM Transactions on Information and System Security, Vol. 8, No. 4, 2005, pp. 351–387.

[8] Goncalves, G. and Poniszewska-Marańda, A., *Role engineering: from design to evaluation of security schemas*, Journal of Systems and Software, Elsevier, Vol. 81, No. 8, 2008, pp. 1306–1326.

[9] Poniszewska-Maranda, A., *Implementation of Access Control Model for Distributed Information Systems using Usage Contro*, SIIS 2011, LNCS, Vol. 7053, 2011, pp. 54–67.

[10] Ahn, G.-J. and Sandhu, R. S., *Role-based Authorization Constraints Specification*, ACM Transactions on Information and Systems Security, 2000.

[11] M. Ben Ghorbel, F. Cuppens, N. C.-B. and Bouhoula, A., *An extended role-based access control model for delegating obligations*, LNCS, Vol. 5695, No. 2, 2009, pp. 127–137.

[12] Poniszewska-Maranda, A., *Conception Approach of Access Control in Heterogeneous Information Systems using UML*, Journal of Telecommunication Systems, Springer-Verlag Heidelberg, Vol. 45, No. 2-3, 2010, pp. 177–190.

[13] Neumann, G. and Strembeck, M., *A Scenario-driven Role Engineering Process for Functional RBAC Roles*, In: 7th ACM Symposium on Access Control Models and Technologies (SACMAT), 2002.

[14] Coyne, E. and Davis, J., *Role Engineering for Enterprise Security Management*, Artech House, 2008, Artech House.

[15] Basin, D., Doser, J., and Lodderstedt, T., *Model driven security: From UML models to access control infrastructures*, ACM Transactions on Software Engineering Methodology, Vol. 15, 2006, pp. 39—-91.

[16] G. Booch, J. R. and Jacobson, I., *The Unified Modeling Language User Guide*, Addison Wesley, 2004.

[17] Group, O. M., *OMG Unified Modeling Language (OMG UML): Superstructure*, Tech. Rep. Version 2.2, 2009.