Scalable Computing: Practice and Experience
Volume 12, Number 1, pp. 35–49. http://www.scpe.org
ISSN 1895-1767

MANAGEMENT OF ACCESS CONTROL IN INFORMATION SYSTEM BASED ON ROLE CONCEPT

ANETA PONISZEWSKA-MARANDA*

Abstract. Development of technology, progress and increase of information flow have the impact also on the development of enterprises and require rapid changes in their information systems. The growth and complexity of functionality that they currently should face cause that their design and realization become the difficult tasks and strategic for the enterprises at the same time. The informations systems store huge amount of data and allow to realize thousands of operations and business transactions on these data each day. In this case, it seems necessary to have the methods, techniques and tools that can make possibly the development of information system on level reflecting currently requirements.

The paper describes the aspects of access control management in information systems based on the concepts of roles. This concepts can be presented by the role-based access control model and its extensions defined during last years. The practical implementation of presented concepts was given in the form of platform for access control management that can be used by system developers and security administrators to support their job in assuring the security of data stored and processed in an information system and assuring the global coherence of access control rules in the whole system.

The proposed platform was based mainly on the approach connected with the access control model based on the role concept that reflects in the better way the company's organization on the access control level. The platform can be enrich with additional tool for access control administration with the use of other access control models.

Key words: security of information systems, access control management, access control models, role-based access control

AMS subject classifications. 68N02, 68U02

1. Introduction. Development of technology, progress and increase of information flow have the impact also on the development of enterprises and require rapid changes in their information systems. The growth and complexity of functionality that they currently should face cause that their design and realization become the difficult tasks and strategic for the enterprises at the same time. The informations systems store huge amount of data and allow to realize thousands of operations and business transactions on these data each day. In this case, it seems necessary to have the methods, techniques and tools that can make possibly the development of information system on level reflecting currently requirements.

It is important also for an enterprise to develop the security system that secure the information system against external threats. Very important stage of data protection building in information system is the creation of high level model, independent from the software, satisfying the needs of protection and security of a system. One of the basic concepts of protection models is access control. The purpose of access control to data in information system is a limitation of actions or operations that the system's users can execute. The access control based on role concept represents interesting alternative in relation to traditional systems of DAC (Discretionary Access Control) type or MAC (Mandatory Access Control) type. RBAC (Role-Based Access Control) model based on a role concept defines the user's access to information basing on activities that the user can perform in a system.

Security policies of information systems determine that it is necessary to define for each user a set of operations that it could be perform. Due to it the set of permissions should be defined for each system's user. It suffice to determine the permissions for execution of particular methods on each object accessible for that user. It is exists the need to create the tool, designated mainly for security administrator who could manage one of the security aspects of information systems, namely the control of users' access to data stored in a system.

To create the administration tool the access control model based on role concept in extended version (extended RABC) was chosen. It is necessary to deliver the tool allowing the definition of access control rules for any information system. It is exists the need to ensure the integrity of defined early access control rules in situation when we want to extend the existing information system by new components (i.e. applications). It is also necessary take the attention that two actors were distinguished in the design process of an information system and its associated security scheme: application/system developer and security administrator who cooperate with each other to define and apply the set of roles defined for particular system's users in according with security constraints assuring the global security strategy of an enterprise. The paper describes the platform created for access control management in field of information systems that can be used by these two actors

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

(i.e. application/system developer and security administrators) fulfilling their responsibilities in assuring the security of data stored and processed in information system and assuring the global coherence of access control on the global level.

The proposed platform was based mainly on the approach connected with the access control model based on the role concept that reflects in the better way the company's organization on the access control level. The created platform was based on the extended RBAC model [9] that provides the developers more flexibility and complex view of the organization security. However, the additional tool for administration of access control of information systems using the traditional access control models, as DAC model and classical RBAC model was added to this platform.

The paper is structured as follows. The first part presents the outline of access control approach and access control models, especially the extended RBAC model. The second part deals with the partition of responsibilities in creation process of access control rules realized by two main actors and describes in short the role engineering in creation of security schema, based on the extended RBAC model. The last part presents our proposition of a platform for management of access control in information systems and administration tool for access control in information systems with the use of DAC model and classical RBAC model.

2. Access control and access control model based on role concept. Access control represents one of the components of information system security, named logical security. Logical security contains three mutually supportive technologies that can be used to provide the system security: authentication, access control and audit. However, access control is the most important technique on logical security level and it is used frequently.

Access control allows to define the user's responsibilities and possibilities in a system. It can define what a user can do directly and also what programs executing on behalf of the user are allowed to do. Access control limits the activities of successfully authenticated users basing on the security constraints defined on the conception level and on the administration level. Access control approach consists of two components [1]:

- set of access policies and access principles that determine the possible access of system's users to data and information stored in a system using the access modes and
- set of control procedures (security mechanisms) that allow to verify the access requests sent by system's users in agreement with defined principles and rules; these access requests may be allowed, denied or modified.

The logical security system can contain two components that cooperate with each other to assure the global security of information systems:

- strategy of logical security that determines all the environments and specifications of the entire organization on the security level and
- access model with:
  - set of concepts to describe the system objects (data access) and system subjects (users),
  - definition of the users' rights to access the data,
  - access control policy that describes how users can manipulate data, defines data structure and manages the users' rights to access the data.

Three main access control policies were defined and used in practice during last years: Discretionary Access Control (DAC), Mandatory Access Control (MAC) and Role-Based Access Control (RBAC).

Discretionary Access Control (DAC) [1, 13, 14] manage the access of users to the information basing on user's identity and authorizations or rules that specify. The access to the object in the specific mode is granted only to subjects for which an authorization rule exists in ACL (Access Control List) and was verified. However, DAC has the weakness that information can be copied from one object to another and it is difficult for DAC to enforce the safety policy and protect the data against some security attacks.

The second access control policy is Mandatory Access Control (MAC) that was defined to enforce the lattice-based policies [1, 15]. MAC strategy manges the accesses to the data basing on the classification of subjects and objects in the system. Mandatory policy can also be defined as a flow-control policy because it prevents information flow towards objects of a lower classification.

Access control based on the concept of role - Role-Based Access Control (RBAC) [2, 3, 4] is an interesting alternative to the traditional approaches of DAC type or MAC type [1]. RBAC model requires the identification of roles in a system. The role is viewed as a main semantic structure around which the access control policy is formulated. The role can represent the user responsibility, the competency to do a specific task and it can embody the authority of system users. The roles can be defined for different job functions in an organization

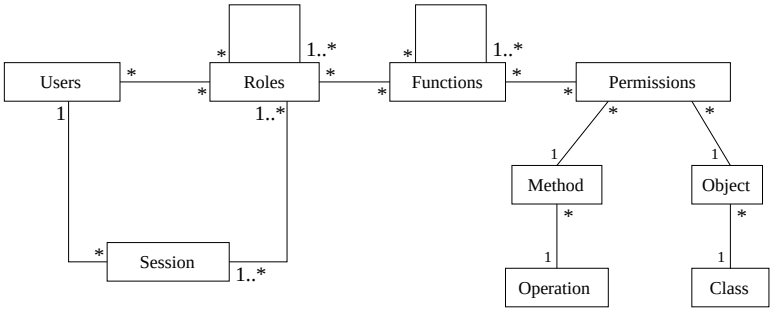

Fig. 2.1. *Extended RBAC model*

and the users can be assigned to the roles basing on their responsibilities and qualifications.

The RBAC model is very popular and stable in access control domain of information system security and for that reason it was extended during our studies. The obtained extension of RBAC model was next used as a base for creation of access control platform in field of access control management.

During our previous studies, we decided to extend the classical RBAC model. The reason for such extension was the need of better expressing the enterprise organization on the access control level, their relationships and dependences. In *extended RBAC (eRBAC)* model [9, 11, 22] each role realizes a specific task in the enterprise process and it contains many functions that the user can take and perform. It is possible to choose for each role the necessary system's functions. Thus, a role can be presented as a set of functions that this role can take and realize. Each function can be determined by one or more permissions that specify the possibilities of a function in a system. Therefore, a function can be defined as a set or sequence of permissions. The addition of function concept caused the necessity of new added relationships between the elements of model: R-F relations (i.e. relation between role an function), F-F relation (i.e. inheritance relation between functions) and F-P relations (i.e. assignment of permissions to a function).

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 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.

In extended RBAC model, the permission was 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.

Therefore, the extended RBAC mode is based on classical RBAC model and extended by addition of some elements, i.e. function, object, method, class, operation, to express more complex the elements of company's information system that are secured by these model (Fig. 2.1).

3. Two actors in role engineering of information system security. Two types of actors cooperate in the design and realization of security schema of information system [11]: 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 constraints that should be taken into consideration on the company level. We propose the partition of responsibilities between these two actors and their cooperation in order to fix the global access control (i.e. security schema) that fulfill the concepts of extended RBAC model. This partition of responsibilities is presented in figure 3.1 and they are divided into two stages: conception stage and exploitation stage.

3.1. Conception stage. The realization process of information system or simple application is provoked by a client's request or in general by 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 create the logical model of application/system, next define 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 user's needs. These elements can be also presented in a form adequate to the access control concepts - it will be the extended RBAC model in our case. Therefore, the developer generates the sets of following elements: roles, functions, permissions and constraints. These sets of elements should be presented

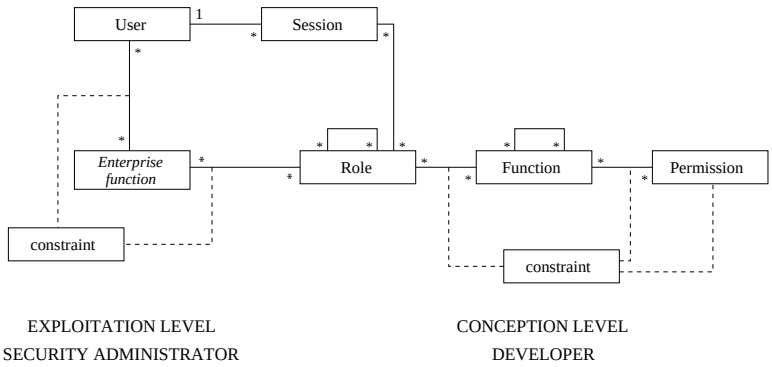

FIG. 3.1. *Two actors in creation of information system security - application developer and security administrator*

to the security administrator in a useful and legible form. The tasks of application developer basing on the extended RBAC model can be determined as follows:

- assignment of elements: permissions to functions and functions to roles,
- definition and setting up the security constraints associated to the elements of this application.

3.2. Exploitation stage. Next, 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 extended RBAC model: 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 (users) and functions realized in the enterprise (enterprise functions).

An *enterprise function* is a task or a set of tasks that can be realized in a system by the same person. If the tasks of an enterprise function require the cooperation between two or more persons this function is realized by these persons taking into consideration the constraints defined on such function that specify the additional conditions for its execution [22].

The main task of security administrator is to assign the users to roles based on the relations between the enterprise functions and the roles (Figure 3.1). This assignment is based on user's responsibilities in the organization. Security administrator is also responsible for the definition and assignment the security constraints on global level. These constraints concern first of all the following relations: user-enterpriseFunction, enterpriseFunction-enterpriseFunction and enterpriseFunction-role.

3.3. Role engineering in creation of security schema. The most important stage in role engineering process of security schema creation is the complex and proper identification and definition of set of roles in the application or in the whole information system form the point of view of RBAC/eRBAC model. The role engineering process of the security scheme in information system using the RBAC/eRBAC model is proposed as follows (Fig. 3.2) [22]:

- Application/system developer creates the system application or a set of system applications (i.e. logical model, project) using object-oriented methodology (e.g. Unified Modeling Language - UML) for analyses and design of information systems. These methodology is used to define the application Model containing all elements that express the needs and requirements of users.
- Application developer initiates the process of user profile creation (e.g. role engineering) [11] based on the security rules concerning this application.
- The application Model created by application/system developer is translated into the concepts of access control models (i.e. RBAC or eRBAC) based on the connections of UML concepts with the concepts of RBAC/eRBAC model [10, 11]. Also the process of user profile creation is finished on the developer level.

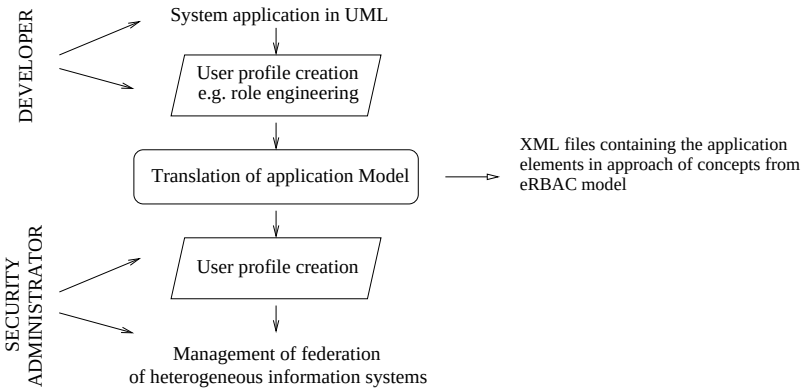

FIG. 3.2. *Role production in creation of security scheme*

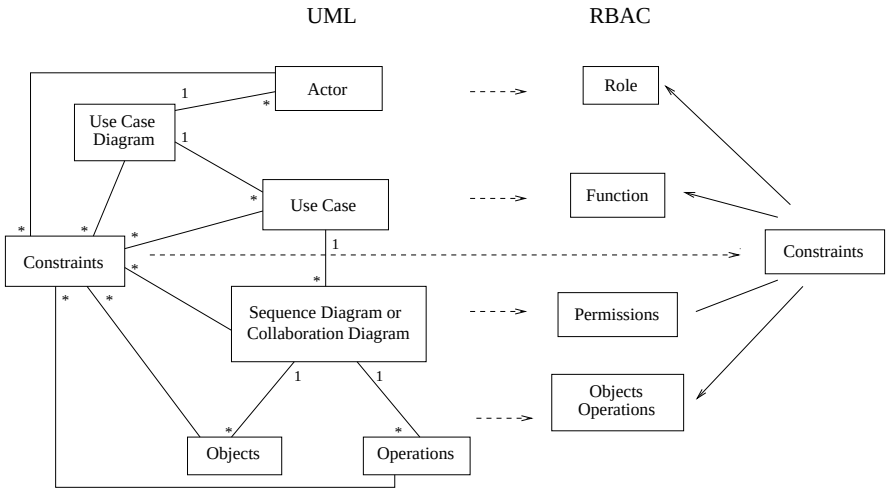

FIG. 3.3. *Concepts of extended RBAC model and their relationships with UML concepts*

- Security administrator receives the Model containing the lists of access control elements that are presented in a special form, e.g. in XML files. The administrator finishes the process of user profile creation using the rules of the global security policy.

3.4. Association of concepts of extended RBAC model with UML concepts. The UML (Unified Modeling Language) has been chosen for the analysis and design of information system. It is also used to design the security schema on access control level. Nowadays, UML is a standard modeling tool, properly reflecting the description of the information system and its requirements. UML proposes the set of design diagrams to present the modeling system. Two types of the UML diagrams have been chosen to present and implement the elements of extended RBAC model: use case diagram and sequence diagram. The concepts of extended RBAC model were joined with some chosen concepts of UML (Fig. 3.3). Description of connections between concepts of extended RBAC model and UML concepts was given widely in [9]:

- *RBAC role* is joined with UML actor,
- *RBAC function* joined with UML use case,
- *RBAC methods* and *objects* joined with methods and objects of UML,
- *RBAC permissions* can be found in the sequence diagrams,
- *RBAC constraints* are joined with constraint concept existing in UML [11],
- relations of different types that occur between the elements of extended RBAC model can be found in the use case diagrams and in the sequence diagrams.

3.5. Creation of user profiles basing on extended RBAC model. Each security policy demands in general the definition for each user a set of operations that the user will be allowed to execute. In consequence,

using extended RBAC model a set of permissions should be defined for each user. It suffices to specify the permissions for the execution of certain methods on each object accessible for that user. According to the connections between extended RBAC model and UML, a definition of user profiles on the developer level and security administrator level was proposed [22].

The implementation of extended RBAC model using UML concepts can be realized with the use of sequence diagrams, where the permissions are assigned to the rights of execution of methods realized in each use case [10, 22]. It is possible to identify and define the elements of extended RBAC model using the UML meta-model: the roles, the functions used by these roles and the permissions needed to realize the functions. The list of actors co-operating with the information system (i.e. roles) and the list of use cases (i.e. functions) can be received from the use case diagrams. Next, the analysis of these use case diagrams allows to find the relations between the particular elements: R-R (role-role) relation (from the generalization relation between the actors), R-F (role-function) relation (from the association relation between actors and the use cases) and F-F (function-function) relation (generalization relation between the use cases).

The description of each use case in the form of interaction diagrams (i.e. sequence diagram or communication diagram) gives the information about the actions and activities that allow to realize the functionality of the system's functions. Each activity represents the definition of a method on an object. Therefore, it is possible to specify the permission *execute* for each method attached to a message sending in a sequence diagram or communication diagram and next add to the set of permissions of a certain function. Therefore the F-P relations can be also automatically managed.

3.5.1. Creation of roles. Each user of an information system should have an assigned user profile which is defined by the set of roles played by him. A user profile is defined by a pair *(u, listRoles(u))*: *u* is a user, *listRoles(u)* is a set of roles assigned to this user [22].

It is possibly to give the rules of role creation and the assignments of these roles to users.

User profile should be defined for each user who interact with the information system

$$u_i \vdash userProfile_i$$

User profile is defined as a set of roles that the user can take and realize

$$userProfile_i \vdash setRoles_i$$

The process of role production in the security schema of information system with the use of UML concepts (the concepts of use case diagrams and interaction diagrams, i.e. sequence diagram or communication diagram) contains two stages [12, 22]:

- assignment of *set of privileges* (i.e. permissions) to a *use case* in order to define a function,
- assignment of *set of use cases* (i.e. functions) to an *actor* in order to define a role.

The security administrator is responsible for the creation of user profiles and assignment of users to the roles according to the elements and security constraints defined by the application/system developer and according to the security constraints defined on the global level.

4. Management platform of access control. The presented platform was created first of all for the security administrator to manage the logical security of information system (i.e. security on access control) on the global level. This platform was based on the approach presented in short in the previous sections. The second purpose of the platform was to make possible the cooperation between application/system developer and security administrator to realize and integrate 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 (Fig. 4.1).

4.1. Main functionality of the platform. The important stage of platform functionality is the process of role engineering. This stage is based on the concepts of extended RBAC model 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. Figure 4.2 presents the stages of the role engineering.

The first stage of the role engineering process is the creation of roles and definition of set of roles. This stage is realized basing on the associations between concepts of extended RBAC model and UML concepts. The algorithm of role construction of an information system was specified and implemented using the UML meta-model [22]. The stages of this algorithm are as follows (Algorithm 4.1.1):

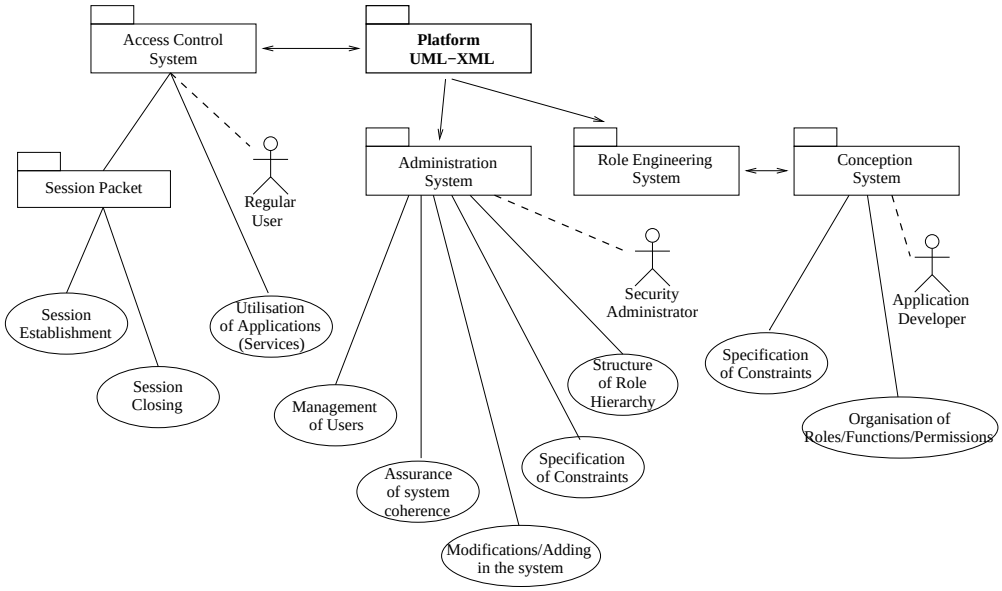

FIG. 4.1. *Functionality of platform for management of access control*

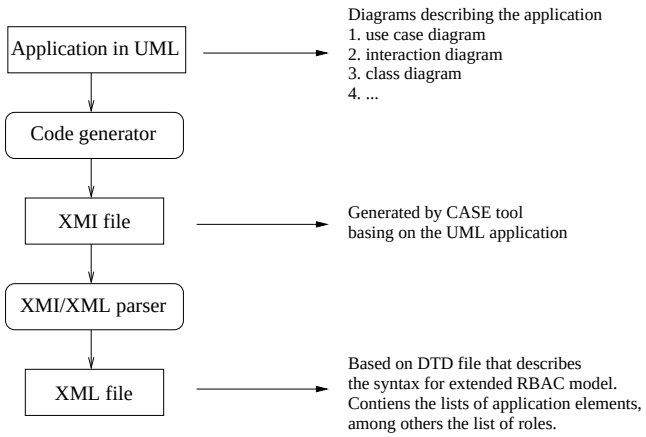

FIG. 4.2. *Role production of extended RBAC model*

- assignment of set of permissions to a function (use case of conceptual model) - Algorithm 4.1.2 and
- assignment of set of functions to a role (actor of conceptual model) - Algorithm 4.1.3.

Algorithm 4.1.1 Algorithm for construction of roles

ConstructionRoles(Model)
Begin
AssignPermissionsToFunctions(Model)
AssignFunctionsToRoles(Model)
End

The first phase of role construction algorithm, i.e. assignment of set of permissions to a function (an use case of conceptual model), contains two main stages:

- searching of set of functions (use cases) in use case diagram and allocating it to *setUC* - Algorithm 4.1.4 and
- attaching of set of permissions to each function (use case) identified in the model that are necessary for

Algorithm 4.1.2 First stage of role creation algorithm

AssignPermissionsToFunctions(Model : Model)
Begin
$setUC = SearchUC\,(Model)$
for *each* $cu_i \in setUC$ **do**
$setPermissions = CreationSetPermissions\,(uc_i)$
done
End

Algorithm 4.1.3 Second stage of role creation algorithm

AssignFunctionsToRoles(Model : Model)
Begin
$setA = SearchA\,(Model)$
for *each* $a_i \in setA$ **do**
$setFA_{a_i} = CreationSetFunctions\,(a_i)$
for *each* $cu_j \in setFA_{a_i}$ **do**
$setRelationUC_{uc_j} = SearchRelationUC\,(uc_j)$
for *each* $uc_k \in setRelationUC_{uc_j}$ **do**
if $(relation\,(uc_j, uc_k)\ of\ type\ \ll include \gg)$
then
$setPermissions_{uc_j} = setPermissions_{uc_j} \cup$
$setPermissions_{uc_k}$
endIfi
if $(relation\,(uc_j, uc_k)\ of\ type\ \ll extend \gg)$
then
$setFA_{a_i} = setFA_{a_i} \cup uc_k$
endIf
done
done
$heritageSetA_{a_i} = SearchRelationA\,(a_i)$
for *each* $a_j \in heritageSetA_{a_i}$ **do**
$setFA_{a_i} = setFA_{a_i} \cup setFA_{a_j}$
done
done
End

function execution - Algorithm 4.1.5.

The second phase of role construction algorithm, i.e. assignment of set of functions to a role (an actor of conceptual model), contains the following stages:

- searching of set of roles (actors) in use case diagram and allocating it to *setA* - Algorithm 4.1.6,
- creation of set of functions for each role (actor) of the model (from the set *setA*) - Algorithm 4.1.7,
- completing of set of functions of each role by the set of functions being in relations with the particular use case and
- completing of set of roles for each user by the generalization relations that exist between the actors.

The second stage of role engineering process is the creation of XMI/XML file. This file can be generated automatically from the logical model of an application created in UML, using the CASE tool. Next, the created parser of XML file is used to analyze the obtained XMI files that contain the description of all elements of each UML diagram of the particular application using the XML syntax.

The functionality of XMI/XML parser was described by the following stages [22]:

- passage of XMI file and identification of elements existing in two types of diagrams (i.e. use case diagram and interaction diagram): actors, use cases, their relations, classes, objects and messages sent

Algorithm 4.1.4 Searching of set of functions (use cases)

SearchUC(Model : Model)
Begin
for *each* $model_i \in Model$ **do**
if $(model_i\ of\ type\ UseCaseView)$ **then**
for *each* $modelElement_j \in model_i$ **do**
if $(modelElement_j\ of\ type\ UseCase)$ **then**
$set\_CU = set\_CU \cup modelElement_j$
endIf
if $(modelElement_j\ of\ type\ Package)$ **then**
$set\_CU = set\_CU \cup SearchUC\,(modelElement_j)$
endIf
done
endIf
done
return *set_CU*
End

Algorithm 4.1.5 Creation of set of permission for each function (use case)

CreationSetPermissions(uc : useCase)
Begin
for *each collaboration* $\subset uc$ **do**
for *each interaction* **do**
for *each message* **do**
$permission = SearchPermissionElements\,(message)$
if $(permission \notin setpermissions)$ **then**
$setPermissions_{uc} = setPermissions_{uc} \cup permission$
endIf
done
done
done
return $setPermissions_{uc}$
End

Algorithm 4.1.6 Search of set of roles (actors)

SearchA(Model : Model)
Begin
for *each* $model_i \in Model$ **do**
if $(model_i\ of\ type\ UseCaseView)$ **then**
for *each* $modelElement_j \in model_i$ **do**
if $(modelElement_j\ of\ type\ Actor)$ **then**
$set\_A = set\_A \cup modelElement_j$
endIf
if $(modelElement_j\ of\ type\ Package)$ **then**
$set\_A = set\_A \cup SearchA\,(modelElement_j)$
endIf
done
endIf
done
return *set_A*
End

Algorithm 4.1.7 Creation of set of functions for each roles (actors)

CreationSetFunctions(a : Actor)
Begin
for *each* $associationEnd_i \subset a$ **do**
$association = Search\,(associationEnd_i)$
for *each* $associationEnd_j \subset association$ **do**
$classifier = associationEnd_j.type$
if $(classifier\ of\ type\ UseCase)$ **then**
$setFunctionsA = setFunctionsA \cup classifier$
endIf
done
done
return $setFunctionsA$
End

between the objects,
- definition of sets of identified elements,
- identification of these elements as the elements characteristic for the concepts of extended RBAC model,
- determination of sets of elements of particular application as the sets of elements characteristic for extended RBAC model,
- generation of XML file that contains the sets of identified and defined elements.

The parser returns the XML documents containing all the elements and their proprieties came from the logical model of the particular application. These elements are used next for the integration of extended RBAC model. The received XML document contains the elements of extended RBAC model such as: roles, functions, permission, objects and methods. The formal description of this file was given in file *eRBAC.dtd* which root element has the following form:

$<!ELEMENT\ \ RBAC(role+, function*, permission*, method*, object*,$
$operation*, class*) >$

The document described above is the result of developer job on local level of access control of an information system. It contains the elements of application logical model created by the developer and translated into concepts of extended RBAC model. Next, this document is given the security administrator who manages the security of the whole system. He should integrate the new elements, e.g. the new application, in the system on the logical security level. The XML document allows to realize such integration independent of the environment. It describes the elements of the new application on the RBAC level used by the administrator of access control. The security administrator uses the XML document to define the rights of the users in the system, so to define the user profiles [22].

The platform should assure the verification and validation of access control rules in the case of integration of new application in a system. It it realized taking into consideration the job of application/system developer and the job of security administrator. The validation should guarantee the global coherence of security schema defined for the organization.

4.2. Additional tool for access control administration using traditional models. Additional tool was created to manage the access control in a field of DAC model and classical RBAC model.

DAC model is a simple model containing first of all subject and objects - the aspects of DAC policy. For each object there is an access control list (ACL) defining privileges for defined user, application or process (Figure 4.3).

RBAC model is much more complicated model. Apart from basic entities: users, roles, permissions, there are some types of constraints (Figure 4.4):
- *Role-Object Constraints* - they limit some permissions for certain objects for defined roles,
- *User-Object Constraints* - they limit some permissions for certain objects for defined users,
- *Number of Members Constraints* - they limit the number of users that can be assigned to the role,
- *Prerequisite Roles* - they define which role is required to be assigned to the user beforehand,

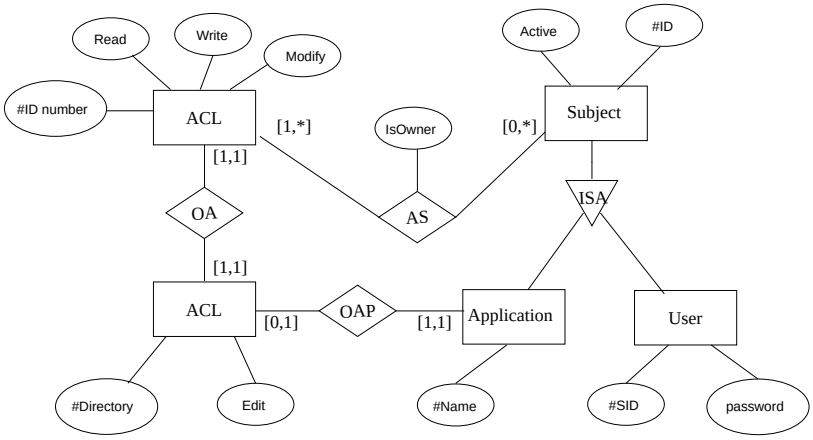

FIG. 4.3. *DAC entity relationship diagram*

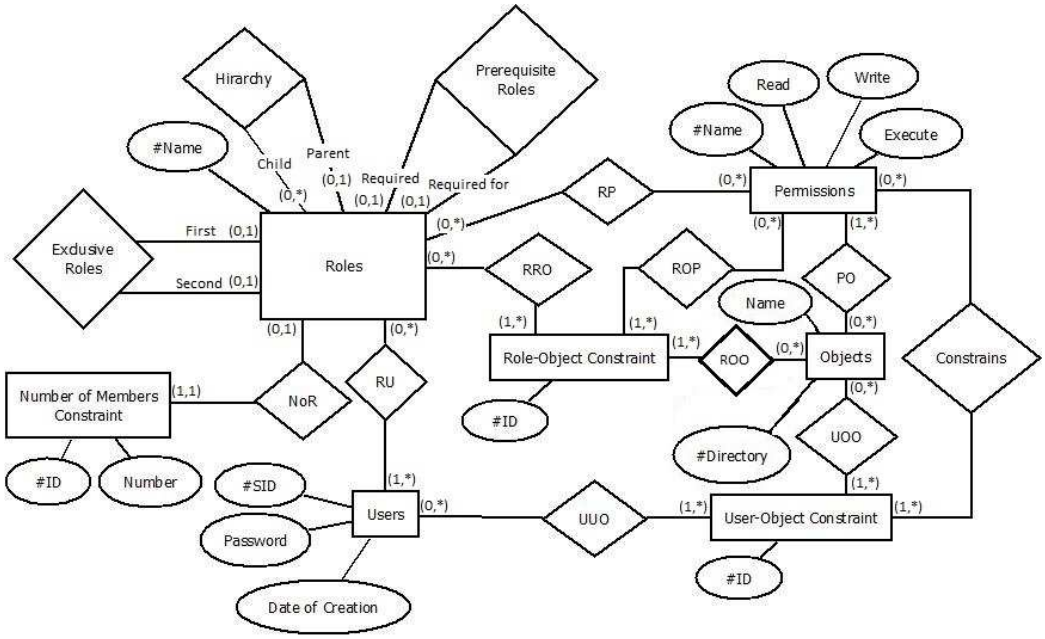

FIG. 4.4. *RBAC entity relationship diagram*

- *Exclusive Roles* - they define which roles cannot be assigned to the user at the same time.

The functionality of the tool for administration of DAC model is presented in form of UML use case diagram on Figure 4.5.

After the administrator chooses DAC model to manage, he is provided with a window consisting of user list, object list, their access control lists and the processes (Figure 4.6). He is able to add, modify and delete users and objects. When creating a new or modifying an object, the administrator has to define its directory and choose its owner from the list. After selecting a user and an object, their common access control list is presented; then the administrator gains ability to modify it as well. He can change which subject is connected with which object. Modifications include also read, write and execute permissions. In application/process panel, the administrator can choose either the process or the application which is allowed the access to chosen objects.

The functionality of the tool for administration of RBAC model is presented in form of UML use case diagram on Figure 4.7.

When the RBAC model is chosen for the administration, there is another window to manage this access control model (Figure 4.8). The security administrator gets the possibility of managing the roles defined in the

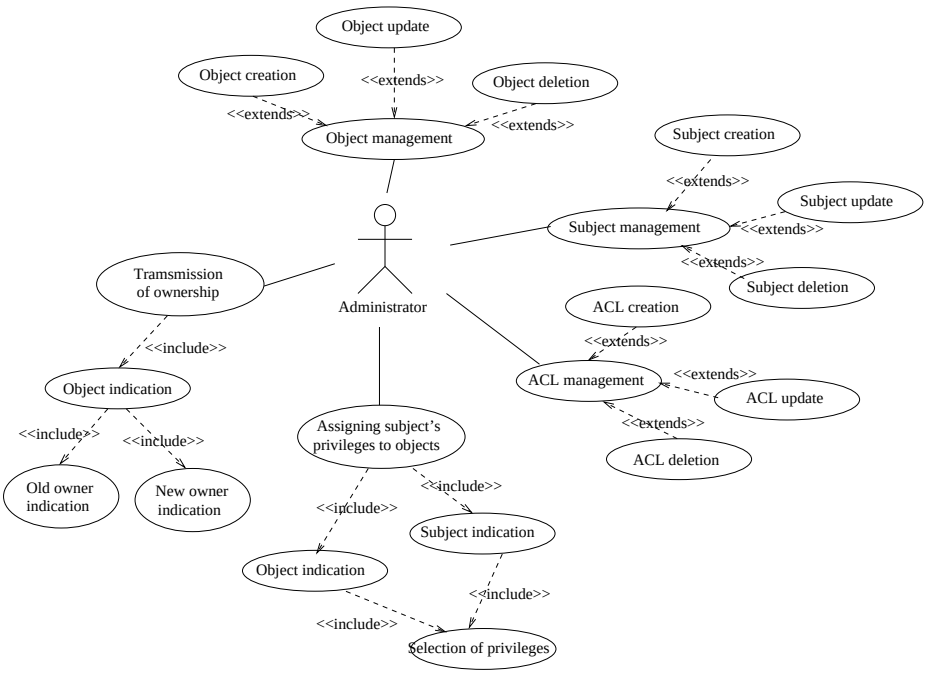

FIG. 4.5. *Functionality of administration tool for DAC model*

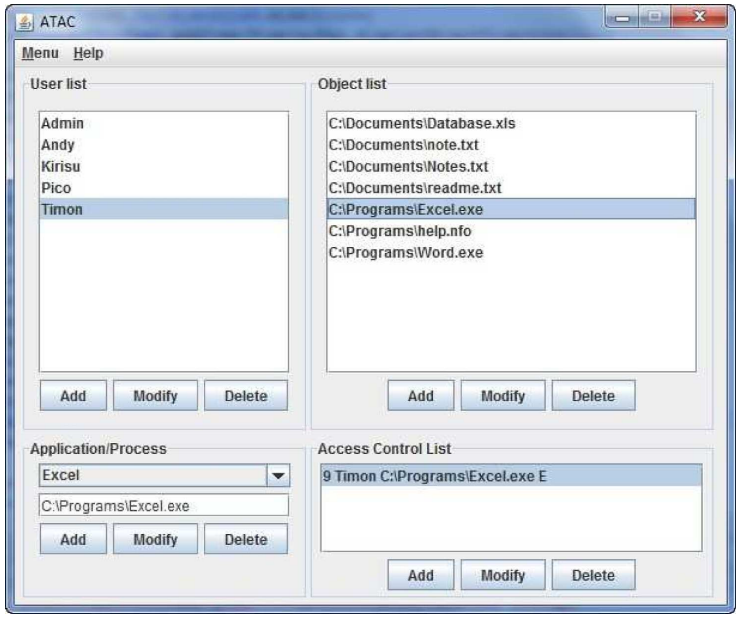

FIG. 4.6. *Tool - DAC model main window*

system obtained from the list. The list can be modified by adding new roles, modifying already existing ones or removing particular roles. The name of the role can be changed, permissions may be set and the role can be grouped in hierarchy by setting the parent role.

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. There is also the possibility of deleting some of them from the list (Figure 4.9).

Another list in the main RBAC window consists of set of permissions. Modifications of the list include forming the new roles, eliminating old roles or changing the names or permissions of existing roles and references

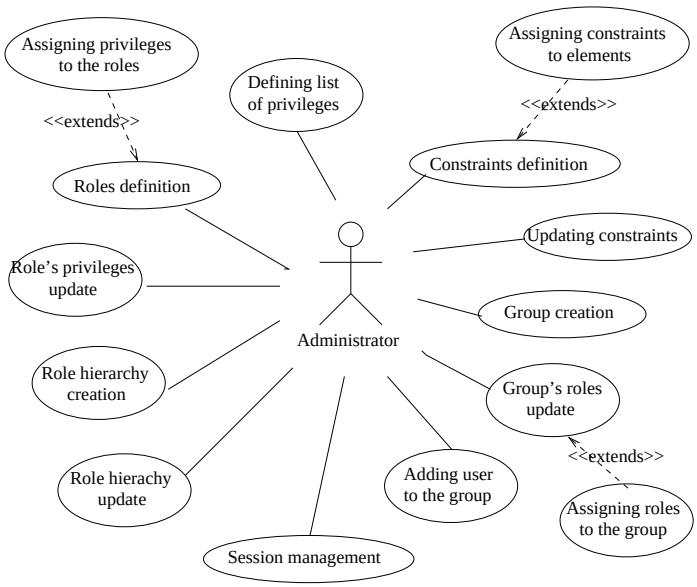

FIG. 4.7. *Functionality of administration tool for RBAC model*

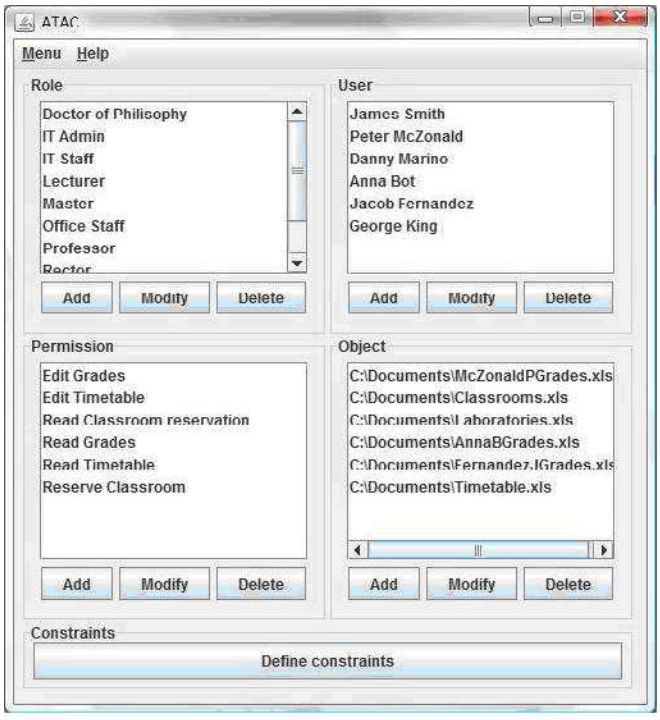

FIG. 4.8. *Platform - RBAC model main window*

to particular objects. The last one of the lists includes objects defined by their directories. Alterations of it embrace changing the directory of existing objects, removing undesired ones or creating new. Furthermore, our application enables the administrator to set various constraints that are the fifth part of the window. The constraints were divided into five groups, namely: role-object constraints, user-object constraints, number of members, exclusive roles and prerequisite roles.

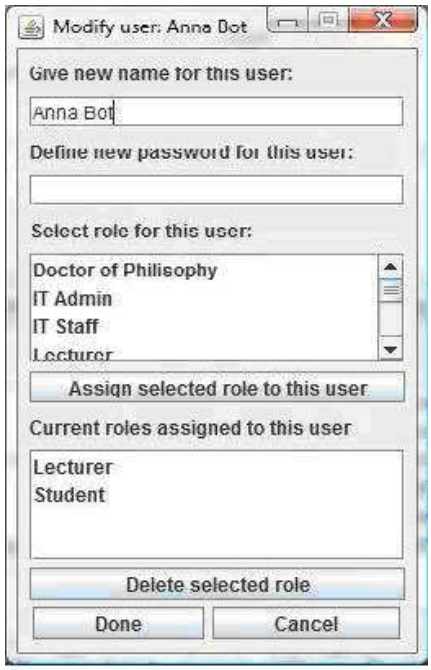

FIG. 4.9. *Platform - User modification window*

5. Conclusions. The presented paper describe the possibility to manage the one of aspects of logical security in the information systems. This management was based on the extended role-based access control model. The concepts of this model were used to define the process of role engineering for creation of user profiles in information systems. 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/system developer and security administrator who cooperate with each other to guarantee the global coherence on access control level.

The tool presented in the paper allows manage the logical security of company information system from the point of view of application/system developer and from the point of view of security administrator. Particularly, the security administrator, who is responsible for the information system security, can facilitate the management of access control of users to data stored in a system.

Currently, not all platform functionalities, shown on figure 4.1, were realized but the already implemented functions allow to manage the security of the information systems.

REFERENCES

[1] S. CASTARO, M. FUGINI, G. MARTELLA AND P. SAMARATI, *Database Security*, Addison-Wesley, 1994.
[2] R. S. SANDHU, E. J. COYNE, H. L. FEINSTEIN AND C. E. YOUMAN, *Role-Based Access Control Models*, IEEE Computer, Volume 29, No 2, s. 38-47, 1996.
[3] R. S. SANDHU AND P. SAMARATI, *Access Control: Principles and Practice*, IEEE Communication, Volume 32, No 9, s. 40-48, 1994.
[4] 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, 2001.
[5] G. BOOCH, J. RUMBAUGH AND I. JACOBSON, *The Unified Modeling Language User Guide*, Addison-Wesley, 1998.
[6] G.-J. AHN, *The RCL 2000 Language for Specifying Role-Based Authorization Constraints*, ACM Transactions on Information and Systems Security, 1999.
[7] E. DISSON AND D. BOULANGER AND G. DUBOIS, *A Role-Based Model for Access Control in Database Federations*, Information and Communications Security, Proc. of 3th ICICS, China, 2001.
[8] G.-J. AHN AND R. S. SANDHU, *Role-based Authorization Constraints Specification*, ACM Transactions on Information and Systems Security, 2000.
[9] A. PONISZEWSKA-MARANDA, G. GONCALVES AND F. HEMERY, *Representation of extended RBAC model using UML language*, Proc. of SOFSEM 2005, LNCS 3381, Springer-Verlag, 2005.
[10] A. PONISZEWSKA-MARANDA, *Access Control Coherence of Information Systems Based on Security Constraints*, Proc. of 25th International Conference on Computer Safety, Security and Reliability, LNCS, Springer-Verlag, 2006.

[11] G. Goncalves and A. Poniszewska-Maranda, *Role engineering: from design to evaluation of security schemas*, Journal of Systems and Software, Elsevier, Vol 81, 2008.
[12] A. Poniszewska-Maranda, *Conception Approach of Access Control in Heterogeneous Information Systems using UML*, Journal of Telecommunication Systems Springer-Verlag, No 45, 2010.
[13] B.W. Lampson, *Database Security*, Addison-Wesley, 1974.
[14] D. Dows and J. Rub and K. Kung and C. Jordan, *Issues in discretionary access control*, Proc. of IEEE Syposium on Research in Security and Privacyv, 1985.
[15] D. Bell and L. LaPadulla, *Secure computer systems: Unified exposition and multics interpretation*, Mitre Corporation, 1975.
[16] E. Bertino and C. Bettini and P. Samarati, *A Temporal Access Control Mechanism for Database Systems*, IEEE Transitions on Knowledge and Data Engineering, 1996.
[17] E. Bertino and P. Bonatti and E. Ferrari, *A Temporal Role-based Access Control Model*, ACM Transaction on Information and System Security, 2001.
[18] A. Gal and V. Atluri, *An Authorization Model for Temporal Data*, ACM Transaction on Information and System Security, 2002.
[19] B. James and E. Joshi and U. Bertino and A. Latif and A. Ghafoo, *A Generalized Temporal Role-Based Access Control Model*, IEEE Transitions on Knowledge and Data Engineering, 2005.
[20] J. Park and R. Sandhu, *The UCON ABC Usage Control Model*, ACM Transactions on Information and System Security, 2004.
[21] J. Park and X. Zhang and R. Sandhu, *Attribute Mutability in Usage Control*, 18th Annual IFIP WG 11.3 Working Conference on Data and Applications Security, 2004.
[22] A. Poniszewska-Marada, *Platform for access control management in information system based on extended RBAC model*, IEEE Computer Press, Proceedings of the 12th IEEE International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, Timisoara, Romania, September 2010.

Edited by: Dana Petcu and Alex Galis
Received: March 1, 2011
Accepted: March 31, 2011