ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 4, Issue 7 , July 2017

Access Control Model and Policies for Collaborative Environments

KadirovMirhusanMirpulatovich

Assistant professor, Department of Information Technologies, Tashkent State Technical University, Tashkent, Uzbekistan

ABSTRACT: The article is devoted for managing and controlling users' permissions, defining users' rights, also developing new profitable role-based access control (RBAC) model in collaborative systems.
It was analyzed models and mechanisms of access control. It was shown context constraints and an algorithm of enforcing of team and task based RBAC model in collaborative system and created a program that relates to role-based access control.

KEYWORDS: Access control, role-based access control, collaborative systems, model of access control, attribute, software module.

I. INTRODUCTION

Access control generally suggests that there is an active user and/or application process, with a desire to read or modify a data object (file, database, etc). For simplicity, we will hereafter refer to an entity as a user and a data object as a file. Access control typically involves two steps: authentication and authorization. In order to authenticate an active user, the distributed system needs some way of determining that a user is in who he/she claims to be. A password is an example of a standard authentication method. On the other hand, authorization to access a file relies on a set of rules that are specified formally and are used to decide which users have the permissions required to access a file. Access control generally suggests that there is an active user and/or application process, with a desire to read or modify a data object (file, database, etc). For simplicity, we will hereafter refer to an entity as a user and a data object as a file. Access control typically involves two steps: authentication and authorization. In order to authenticate an active user, the distributed system needs some way of determining that a user is in who he/she claims to be. A password is an example of a standard authentication method. On the other hand, authorization to access a file relies on a set of rules that are specified formally and are used to decide which users have the permissions required to access a file.

II. ANALYSING OF TT-RBAC MODEL OF ACCESS CONTROL

Collaborative systems are becoming a popular means of providing efficient and scalable access to distributed computing capabilities. In collaborative systems a set of organizations share their computing resources, such as computer cycles, storage space, or online services, to establish virtual organizations aimed at achieving a particular task. Balancing the competing goals of collaboration and security is difficult because interaction in collaborative systems is targeted towards making people, information, and resources available to all who need it, whereas information security seeks to ensure the availability, confidentiality, and integrity of these elements while providing it only to those with proper authorization [1,5].

A variety of access control models have been developed over the years in response to system and security administration requirements. These access control models have been motivated by the need to reduce the security administration overhead commonly associated with low-level subject-object permissions. Models that incorporate additional contextual information and support higher-level policy abstractions can simplify policy administration by reducing the semantic gap between enterprise-level policies and policies that can be directly enforced within a system.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 4, Issue 7 , July 2017

Abstractions, such as “role”, “team” and “task” are developed to model contextual information associated with organizational roles, responsibilities and collaborative activities.

Currently there is still no access control model that rigorously defines the relations among team, task and RBAC entities. Motivated by this requirement, it was defined a Team and Task based RBAC (TT-RBAC) access control model that extends the NIST RBAC model through adding sets of two basic data elements called teams and tasks. The TT-RBAC model defines four model components through which entity relations under different situations are defined. The functional requirements for these model components are also specified.

III. AN OVERVIEW OF THE NITS RBAC REFERENCE MODEL

The NIST RBAC model [2] is defined in terms of four model components: Core RBAC, Hierarchical RBAC, Static Separation of Duty Relations, and Dynamic Separation of Duty Relations. Core RBAC defines a minimum collection of RBAC elements, element sets, and relations in order to completely achieve a role-based access control system. Hierarchical RBAC component adds relations for supporting role hierarchies. Static Separation of Duty Relations adds exclusivity relations among roles with respect to user assignments. Dynamic Separation of Duty Relations defines exclusivity relations with respect to roles that are activated as part of a user’s session. The NIST RBAC model is defined in Figure 1.

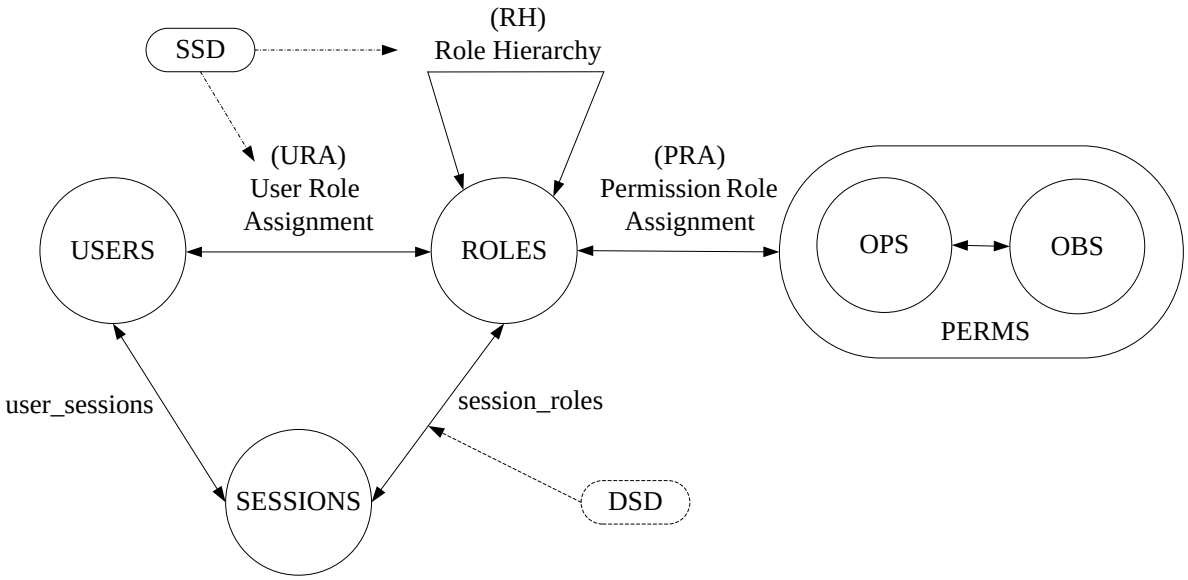

Figure 1: NIST RBAC

Core RBAC model element sets and relations are defined in Figure 1 (without RH, SSD, DSD). Core RBAC includes sets of five basic data elements called users (USERS), roles (ROLES), objects (OBS), operations (OPS) and permissions (PERMS). The RBAC model as a whole is fundamentally defined in terms of individual users being assigned to roles and permissions being assigned to roles. As such, a role is a means for naming many-to-many relationships among individual users and permissions. In addition, the Core RBAC model includes a set of sessions (SESSIONS) where each session is a mapping between a user and an activated subset of roles that are assigned to the user.

A user is defined as a human being or an autonomous agent. A role is a job function within the context of an organization with some associated semantics regarding the authority and responsibility conferred on the user assigned to the role. Permission is an approval to perform an operation on one or more objects. An operation is an executable image of a program, which upon invocation executes some function for the user. For example, within a file system, operations might include read, write, and execute. An object is an entity that contains or receives information. The objects can represent information containers, such as files and directories in an operating system or contents within a database management system. Objects can also represent exhaustible system resource, such as printer, disk space, and CPU cycle. The set of objects covered by RBAC includes all of the objects listed in the permissions that are assigned to

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 4, Issue 7 , July 2017

roles. The types of operations and objects that RBAC controls are dependent on the type of system in which they will be implemented.

IV. CONTEXT CONSTRAINTS OF TT-RBAC

Context is an elusive concept, which has many different meanings to different people and communities. In the area of ubiquitous and pervasive computing context can be defined as: "any information that can be used to characterize the situations of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves". Context may consist of almost all information describing a specific situation. The context information may be static like a person's nationality or dynamic like time[3].

Context constraint is an abstract concept on the modeling level. The context constraint specifies that certain context attributes must meet certain conditions to permit a specific operation. For example, a context role can be activated only when all its associated context constraints evaluate to true [4].

Figure 2 shows that RBAC roles are associated with context constraints. A context constraint is defined through the terms context attribute, context function, and context condition. They are described as follows:

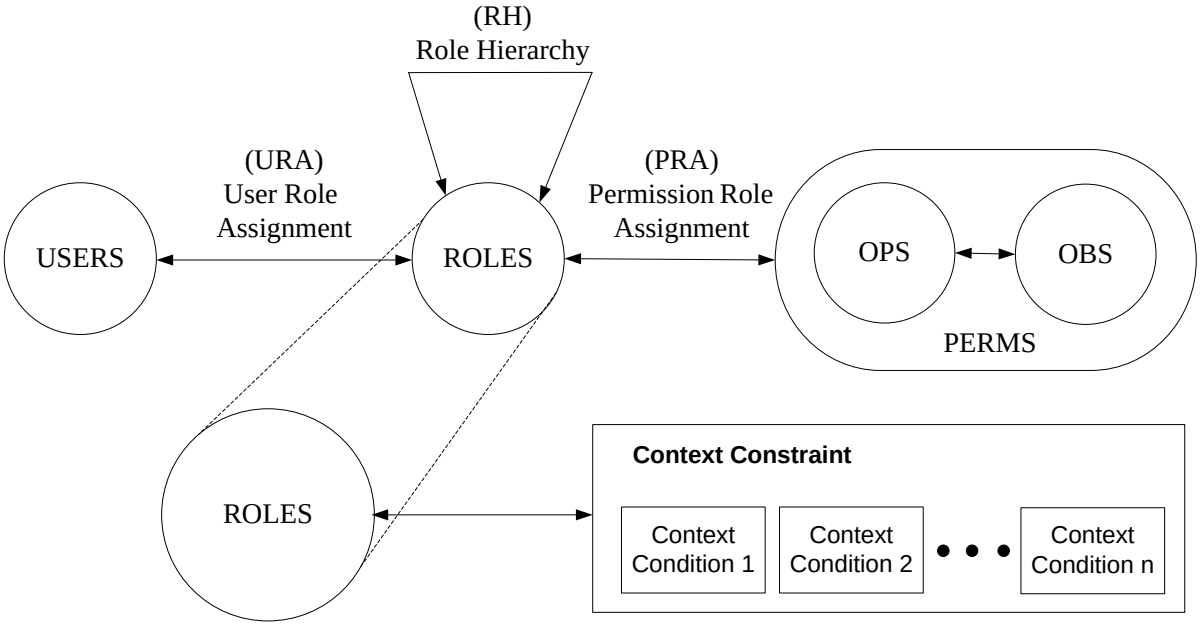

Figure 2: RBAC roles with context constraint

Context attribute represents a certain property of the context whose actual value may change dynamically (e.g. time, date or session-data) or which varies for different instances of the same abstract entity (e.g. location, birthday, or nationality). Thus, context attributes are a means to make context information explicit.

Context function is a mechanism to obtain the current value of a specific context attribute. For example, the function getDate() returns the current date. One or more context functions are encapsulated into a context observer that corresponds to a library or a package in the implementation. For example, the functions getDate(), getTime() and getIP() may be organized into LocalHostObserver.

Context condition is a predicate that consists of an operator and two or more operands. At least one operand represents a certain context attribute, while the other operands may be either context attributes or constant values. Context attributes are gotten by using corresponding context functions. The operator is either a prefix operator that accepts two or more input parameters or a binary infix operator that compares two values.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 4, Issue 7 , July 2017

V. DEVELOPING SOFTWARE MODULE OF TT-RBAC

The program consists of two parts. The first one is defining user rights by choosing users and the second one based on current user. Defining user rights by choosing users is below:

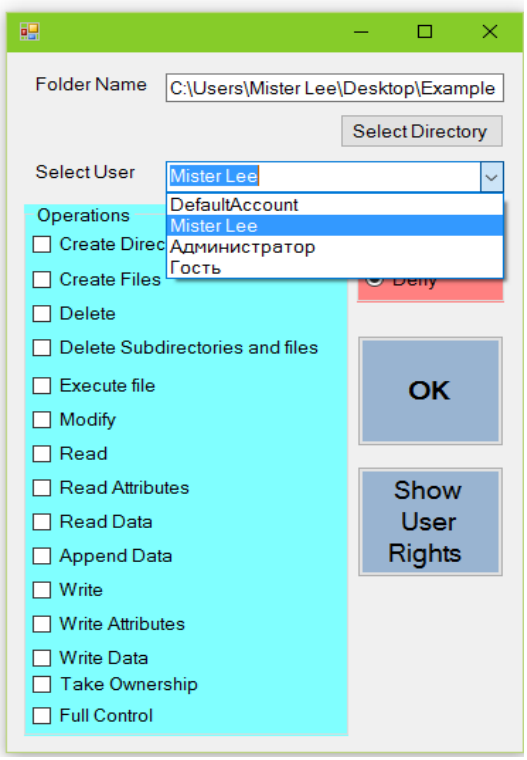

Figure 3: Defining user rights by choosing users

In this part of program defining users rights has been given by choosing users through file which has been shown. Also It can be checked user rights that had been defined through the current file. In the below, the process of defining user rights through file which has been shown:

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 4, Issue 7 , July 2017

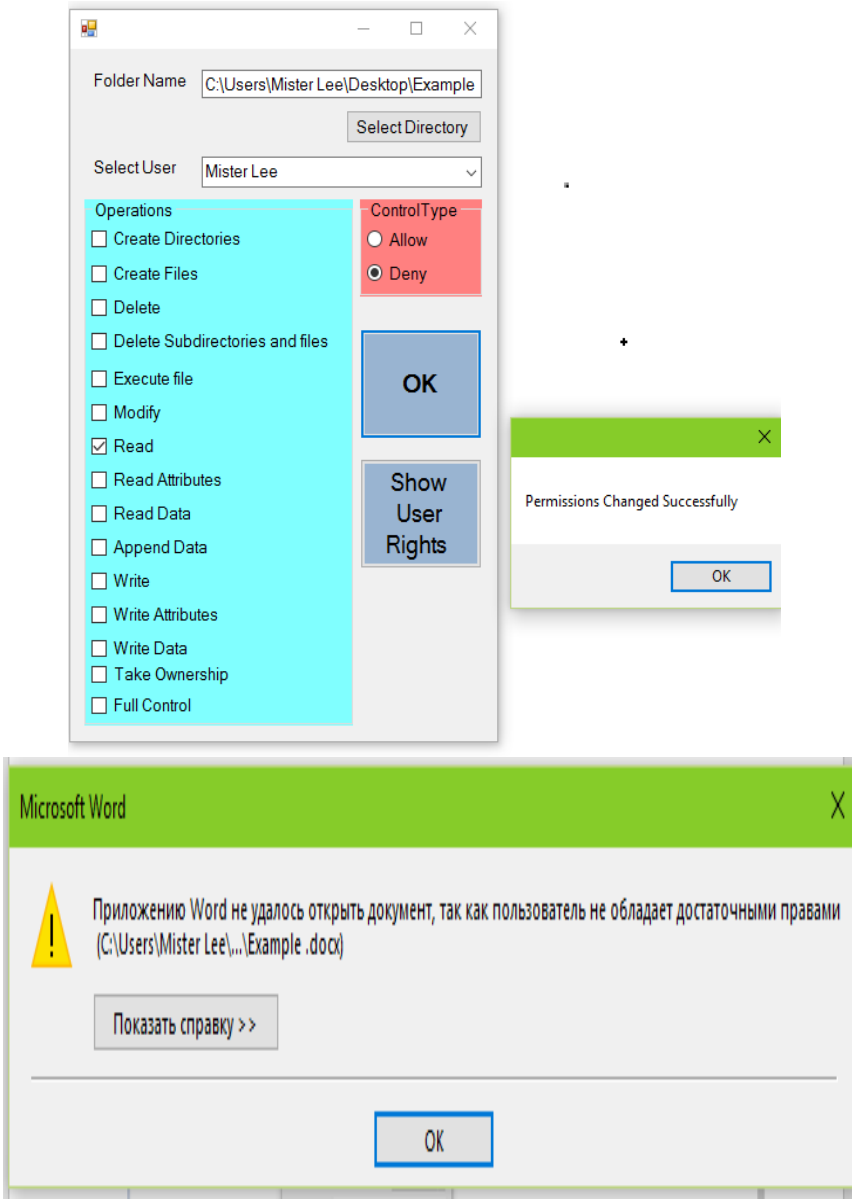

Figure 4: The process of defining user rights through file

In the below, the checking process of defining user rights that had been remarked through the current file:

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 4, Issue 7 , July 2017

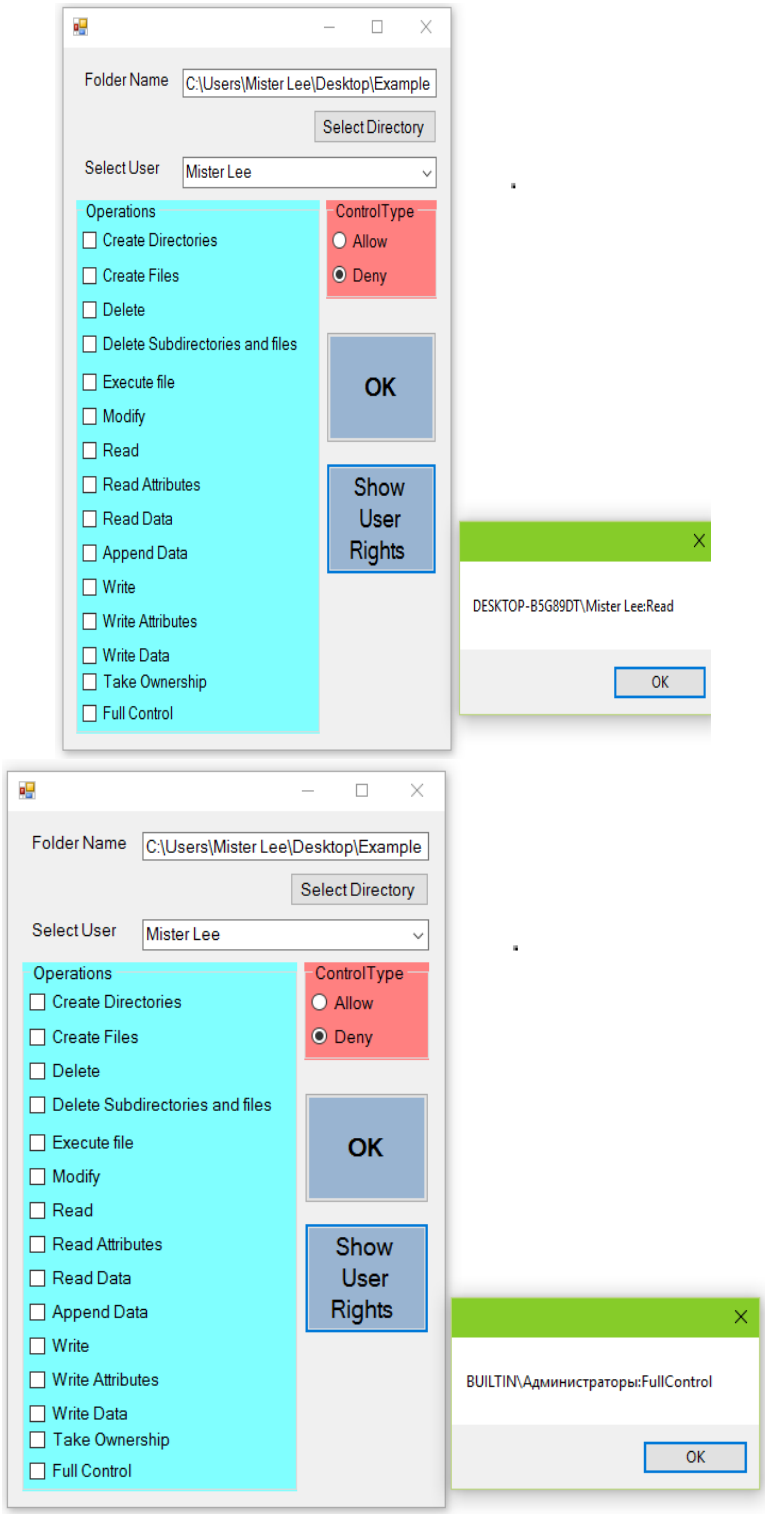

Figure 5: The checking process of defining user rights

The second part of the program consists of the two windows. In the first window, defining rights for file was located. In the second window, defining rights for folder was located.

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 4, Issue 7 , July 2017

VI. CONCLUSION

In this article, it was introduced a software structure for TT-RBAC implementation. The major contribution of this work is using object-oriented technology to implement finegrained context-aware TT-RBAC systems. It was introduced what objects should be defined in a TT-RBAC system, how the functionalities defined in TT-RBAC are arranged into these objects, and how these objects work together to make access control decisions. It was introduce a mechanism for adding context constraints to any TT-RBAC entities. It was defined the core classes and their relations in TT-RBAC implementation. It was introduced how the context constraints are added to TT-RBAC entities. The TT-RBAC authorization evaluation process was investigated.

REFERENCES

[1] W.Tolone, G.Ahn, T.Pai, and S.Hong. Access control in collaborative systems. ACM Computing Surveys, 37(1):29–41, March 2009.

[2] V.C.Hu, D.F.Ferraiolo, D.R.Kuhn. Assessment of Access Control Systems. National Institute of Standards and Technology (NIST) Interagency Report 7316, September 2010.

[3] W.Zhou, C.Meinel. Function-Based Authorization Constraints Specification and Enforcement. In Proceedings of the Third International Symposium on Information Assurance and Security (IAS 2007), pp. 119-114, Manchester, United Kingdom, August 2011.

[4] W.Zhou, C.Meinel. Team and Task Based RBAC Access Control Model. In Proceedings of the 5th Latin American Network Operations and Management Symposium (LANOMS 2010), pp. 84-94, Petrópolis, Brazil, September 2010.

[5] Rajaboevich, GulomovSherzod, KadirovMirhusanMirpulatovich, and TulyaganovZoxidjonYakubdjanovich. "The Methodology of the Ways for Increasing the Efficiency of Intrusion Detection Systems." International Journal of Engineering Innovations and Research 5.5 (2016): 296.

KadirovMirhusanMirpulatovich Assistant professor.Has more than 77 published scientific works in the form of articles, journals, theses and tutorials. Currently works at the department of “Information technologies” in Tashkent State Technical University.