Creating Trust Based Access Policies to Control User Actions on Documents

Danilescu Marcel, Besliu Victor

Technical University of Moldova
Ştefan cel Mare Av., 168, Chisinau, MD2004, Republic of Moldova
Tel.: (37322) 509905, e-mail: marceldanilescu@hotmail.com, vbesliu@yahoo.com

ABSTRACT

The privacy of data and data security is one of the current requirements in organizations. In this paper we present the implementation method, using trust policies. This article complements previous studies concerning the possibility of document security implementation, controlling the information access rights in virtual environments based on Web technologies, and also for the SME.

Keywords: privacy, security, access control, trust, xml

1. INTRODUCTION

In any type of organization (real or virtual), the tasks that must be solved are generally attributed to people, who are grouped according to various criteria, but more often on competence, ability, benevolence, etc. and in each group, the tasks are also allocated on criteria similar to those that led to the creation of groups. For example, in the virtual environment OpenOffice.org, LibreOffice, or any community of sourceforge.net, we find the working groups for: help-desk, design, coding, testing, help systems, user support, documentation, localization and translation, creating sample content, developing tutorials, developing template documents for applications, and many other types of work.

This structure based on working groups, which have a management group, assumes a hierarchical organization, both at the organizational level and at group level. To access to an organization, any member should receive the member's trust, so it can carry out the assigned tasks and has the power, goodwill and other qualities necessary for the completing the task. In the last 20 years, there has been researches on trust which can be accorded to the various groups and to the members of the group [3], [9].

In the literature, the maximum given trust is "Blind Trust" with value 1, and the minimum given trust is "No Trust" with value 0 [10]. Based on those mentioned above, we consider that assigning full trust to a person or group, they enjoy the same trust as the person or group who gave it, and the lowest value of the confidence means that there is no trust.

Between those two values, it can be created a trust hierarchy, based on trust levels. This way of quantifying the trust granted to a user or user group, help us to determine the access and the rights of the actions on the files (called objects henceforth) in a virtual environment.

In practice, not all objects have the same importance for a user or group of users, because each one covers different topics, more or less important to them. Therefore, we can say that, for a category of users is more important one object while for others it is less important, which makes an object to be necessary for a certain user and unnecessary for other. Also there may be objects that need to be provided with a much higher or lower trust degree towards a user or group of users.

Virtual environment based on web technologies, allows an impersonal interaction between various users, knowing one another or not, being part of a real or virtual organization.

Over time, there have been various researches about the privacy and security of data in system, such as

- Bell – LaPadulla system [1];
- "Lattice" based system, designed by Dorothy E. Dennis [2];
- Doctoral thesis "Formalising trust as a computational concept" [3];
- Role based access model [4];
- EPAL (Enterprise Privacy Authorization Language [5];
- XACML developed by OASIS [6];
- "Control Access To Information By Applying Policies Based On Trust Hierarchies" [7];
- „Assurance model behaviour in social networks based on trust” [8].

2. CONCEPTS AND TERMS

Generally, the trust, which is granted to a person [3], allow that person to perform an action within a group and is based on various criteria such as:
Reputation; competence; loyalty; experience; goodwill; courage.

Depending on requirements, criteria necessary for the application of a trust policy are adaptable.

Further, we will describe a theoretical model of applicability and enforcement of the access policies based on trust. To create access control policies for users of virtual storage, we must define the following: Objects; Object Group; Life cycle or lifetime of an object; Users; Users Groups; Domains; Trust level corresponding to an action; Requirements for establishing the trust level; Trust level granted to a user for a specific domain, or to one or more objects of the domain; Trust level granted to a user group within a group of a domain.

The **object** is a homogeneous and unitary entity of information on electronic support, on which the actions is carried out to achieve the purpose for which it was created.

Object group represents a collection of objects that belong to a domain.
Generally, it is difficult to identify and determine that an object belongs strictly to a group or another.

May be situations where an object may belong to the several fields. For an easy distribution of the objects in groups, we consider that the object belongs to the domain that has the most interaction with it, and eventually end the lifecycle of the object. Groups of objects may have inside a hierarchical organization; some objects arising from end of life on another object.

Object's life cycle (duration of existence of an object) represents all the stages of an object, from creation to archiving or deletion.

The user, is the person who interacts with the objects during the existence period of object's and performs different actions.

The **groups of users',** are users who interact with a set of objects in a domain.
Domain of activity is part of the activities performed, grouped by common characteristics, such as technical knowledge, economic or scientific common interest, scope, etc...

Definition: We call a trust value granted to an action, a value between 0.00 and 1.00 corresponding to actions taken on an object, according to the necessary competences for enforcement action.

Requirements needed for determination of value of the trust are an arbitrary set of conditions on which one user must meet to be granted with a certain trust value in order to execute actions.

Trust level is a permission granted to a user or group of users to interact with an object or several objects from certain area of activity and to perform specific actions corresponding to the **trust value**.
To create logical mechanism to control access to objects, we formalized the principles outlined above.

For this, we make the following considerations about the elements with which we work.
We define a hierarchy as a finite set of values ($H_1 \leq H_2$) ascending ordered.
We define a sub-hierarchy ($I_1 \leq I_2$) as a sub-set of a hierarchy ($H_1 \leq H_2$) if ($I_1 \leq I_2$) $\subseteq$ ($H_1 \leq H_2$).

Between objects and user interaction is possible, that a user can perform certain operations on an object: Reading; Creation; Writing (update); Addition (append); Copy; Rename; Deletion; Archiving; Approval.

Interaction between object and user we call action and we will note with a_i. All actions will create the set of actions A.

We define a relationError! Reference source not found. as a connection that exists between two elements x and y belonging to disjoint sets and that can be expressed as (r, x, y).

A trust relationship is a relationship that can be quantified by values between 0.00 and 1.00 corresponding to "no trust" to "blind trust".

When **r** = 0, there is no trust relationship between **x** and **y**, and when **r** = 1, trust is complete. Between these two values representing the extreme value for relations , can be defined various actions that can be applied on elements, depending on the trust value relationship , applied to a user or group of users, for an item or category of items.

Lemma: An action "a" of "x" over "y" can only occur if the value of the relationship between "x" and "y" is equal to or greater than the minimum necessary to enforce the action.

Thus: if $\boldsymbol{r}=0 \vee \boldsymbol{r}<\boldsymbol{v}$ ($\boldsymbol{v}$ = minimum value for which $\exists\ \boldsymbol{a}$) $\Rightarrow \neg\boldsymbol{a}$, otherwise $\boldsymbol{r}>\boldsymbol{v}\Rightarrow\boldsymbol{a}$.

Therefore, the control of "***a***" actions can be realized according to the value attributed to "r".
If "***r***" has "***v***" value, greater than the minimum required to execute an action, then "***r***" corresponds to all actions whose value is less than or equal to "***v***". If no value is set for "***r***", then „***r=0***”.

We propose the correspondence between actions and trust levels values.

Table 1. The example of the according of the incredibility values associated to the actions

The level of the încredibility	**Action**
0. 01	a00
0.02	a02
…………….	…………
0.1	a1
0.2	a2
0.3	a31,a32,a33
0.7	a41,a42
0.9	a5
…………….	…………
1	a100

These actions applied to objects can be represented as a tree as shown at fig.1 [1].

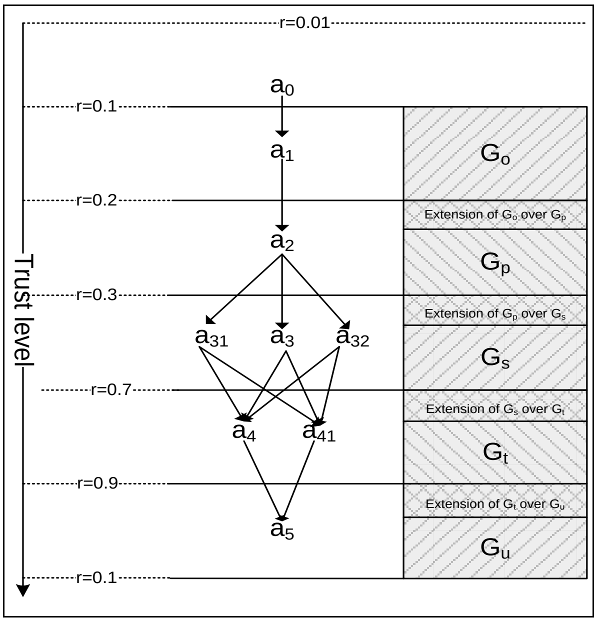

Figure 1. The actions applied to the objects

An object or group of objects belong generally to a domain. Depending on the relationship of trust between a group of users (or one user) that belong (belong) to a domain and the group of objects (object) are set the actions they (this) may apply to object. From the above results in the following:

1. Each object is attached to a group of reliable values corresponding to a hierarchy of actions, which is the order of actions which will cover the subject.
2. Each user has a level of confidence in relation to object, depending on which enjoys the confidence to perform actions on the object or group of objects.
3. Right of execution of an action on an object is determined by the value of trust.

This means that it can create a first set of tuples representing the relationship between object groups, user groups and actions based on the level of confidence (GB, D, G, R) on that we call general policy of trust.

Where:
• GO = group objects
• D = domain
• G = Group of users
• R = the confidence level of the group.
Given that an object Oi, which belongs to a group of objects GO_J, in a domain activity D_l ,for a user group G_m has a trust level value R_u , that can only less than or equal to the confidence of the group R_g.

Which is transcribed as:

$$R_u (O_i,D_l,U_n)\leq R_g(GO_j,D_l,G_m)$$

If the in above relationship for users and objects is replaced R_u the confidence level with the corresponding action, we obtain the following tuple: (O_i, D_l., U_n, A_x). In other words, the action A_x on the object O_i is allowed for the user U_n of the domain with the confidence level R_u equal to one level of trust that allows the execution of the action [2].

To simplify allowed actions for an user to an object, we can use only tuple U_n,Au, allowed actions are those that correspond to the appropriate confidence levels. This leads to the attachment of a group of tuples (U, A) to an object [2].

Steps of an object should be recorded as hierarchical sets of tuples consisting of shares and an integer value that can express Vs state of the object (value status). $Vs\in(0,1,2)$ where :

- 0 = unexecuted
- 1 = in work
- 2 = performed

Expression of trust policies applied to a user to a particular object belonging to a particular area, in simplified form, is of the form (O_i,U_n,A_x), and as complete is (O_i,D_l,U_n,A_x).

May be situations where the rights of user groups may not involve the existence of appropriate actions assigned to users in the group. Then you have to establish some restrictions [2].

Restrictions: We call restriction, limiting the action of an user for an object or category of objects, though he had the necessary confidence level for enforcement action.

To designate a restriction on an action, we note with“ -A” a detailed restriction and“-Ru”a set of restrictive policies. Thus we have a set of elements (Oi, A,-Ax) or (Oi, A,-Ru) for the domain D_l.

In general, a restriction must be accompanied by a delegation to another user.

The delegation is reliable transfer made from one user to another in order to carry out actions on objects.

Basic principles applied in trust policy are:

- generalization - allows reliable policy of an object or class of documents applied to a user to apply to all members of the group who have the same level of confidence. We say that relation (O_i, U_n, R_u) in a domain $D_{l,}$ can be transformed in (GO_J, G_m, R_g) or (O_i, G_m, R_g).
- The inheritance allows that trust policy of a group to be applied by default to one member of the group, unless otherwise is specified. In this case, the policy defined as (O_i, GO_J, R_g) for the D_l can be applied to a user like (O_j, U_n, A_x).

3. WORKFLOW MODELING - SUPPORT FOR A IMPLEMENTATION OF TRUST BASED POLICY

Creating workflow is very important in order to facilitate the implementation of policies based on trust , by revealing of the processes $\mathscr{P}$, of the actions flow and the level of confidence granted to various users.

Thus, an object has been suffering through its life a series of processes ordered according to the schedule you created before . To each process Pi corresponds actions (A), events (E), sequences of the flow (F) that determines its semantics . They are executed or are intended for the users.

Each process has a well established position in the workflow of the object, allowing us the opportunity to make a hierarchy of processes, which in turn contain hierarchies of actions (A_k), hierarchies of events (E_k) and sequences of flow(F_k).

In establishing processes flow , are determined the restrictions of the processes, delegation and levels of trust required for user groups in different areas to access and interact with objects.

Design and implementation of policies based on trust involves the determining which actions, events and sequences of flows which contributes each to the process stream (Pi) and their assignment to different groups of users based on their level of trust and their restrictions which need to be applied.

Therefore, we can define the conditions needed to apply a policy of trust.

Let be $O_i \in GO \wedge P_i \in \mathscr{P}$ where $P_i=(p_1,p_2,\ldots p_k \ldots p_n)$, and $p_k=H_k(A_k) \cup H_k(E_k) \cup \Sigma F_k$

For $\forall A_k , \exists U_k \in G_m \Leftarrow \exists R_u , R_u(U_k)=R_a(A_k) \wedge R_u(U_k)=\leq R_g$

Where:
O_i=Object i
GO=Group of objects
P_i=The process applied to O_i
$p_1..p_n$=numbers of subprocess ale P_i
$H_k(A_k)$ = the corresponding action hierarchy to the p_k subprocess
$H_k(Ek)$ = the corresponding hierarchy of events to the pk subprocess
F_k = flow sequences
U_k = user designated to perform the action Ak
R_u = confidence level of the U user , that is needed for the Oi object
R_g = confidence level for the GM group
$R_a(A_k)$ = level of confidence necessary to the enforcement of the Ak action
G_m = the GM group

In the pictures below are summarized these operations.

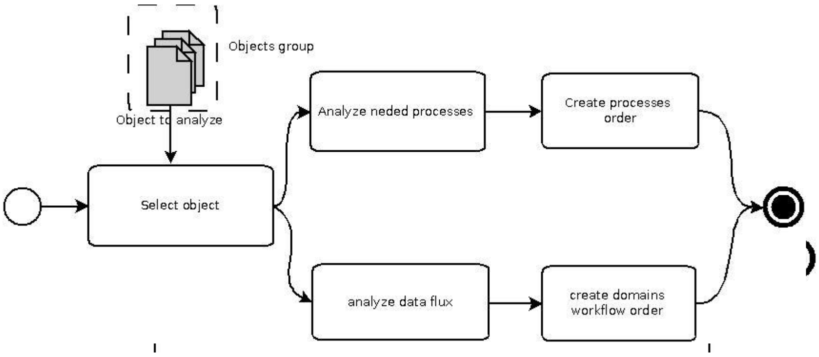

Figure2. Objects analyses

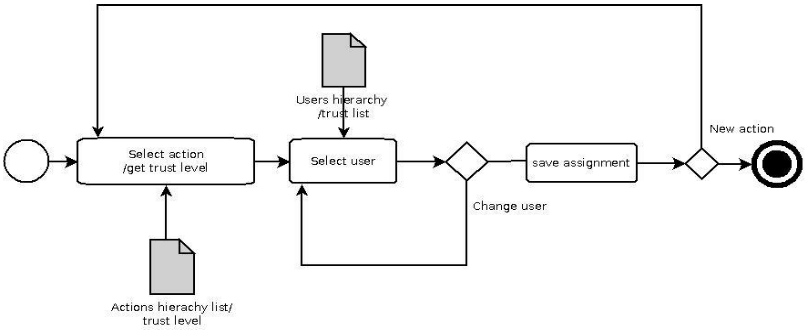
Users hierarchy /trust list
New action
Select action /get trust level
Select user
save assignment
Change user
Actions hierachy list/ trust level

Figure 3. Process analyses

4. APPLYING THE RESTRICTIONS TO THE ONE POLICY OF TRUST

Many times it may happen that in the design process, the user should be assigned a more actions that must be executed at a time. (Ex. the object filling, printing, transmitting information, and so on).

If, at a time designated user is unavailable, his actions it will be delegated to the another user, and the original user will have restrictions for the delegated actions.

5. ACCESS DECISION MODEL

A user issues a request to access an action to be performed on an object. To check a user's access to an object, the process will issue a request to the **access controller**. It will consult **evaluator access** which will issue a request for the **policy evaluator**.

Policy evaluator consults the **access list of users**, **list of user’s delegation** and **list of restriction of the users** and seeks information about the user.

Evaluator decisions consults the **policy evaluator** results and returns a response to the **evaluator access**, which forwards it to the **access controller**.

Depending on the response, the user has access or not to a particular action on an object.

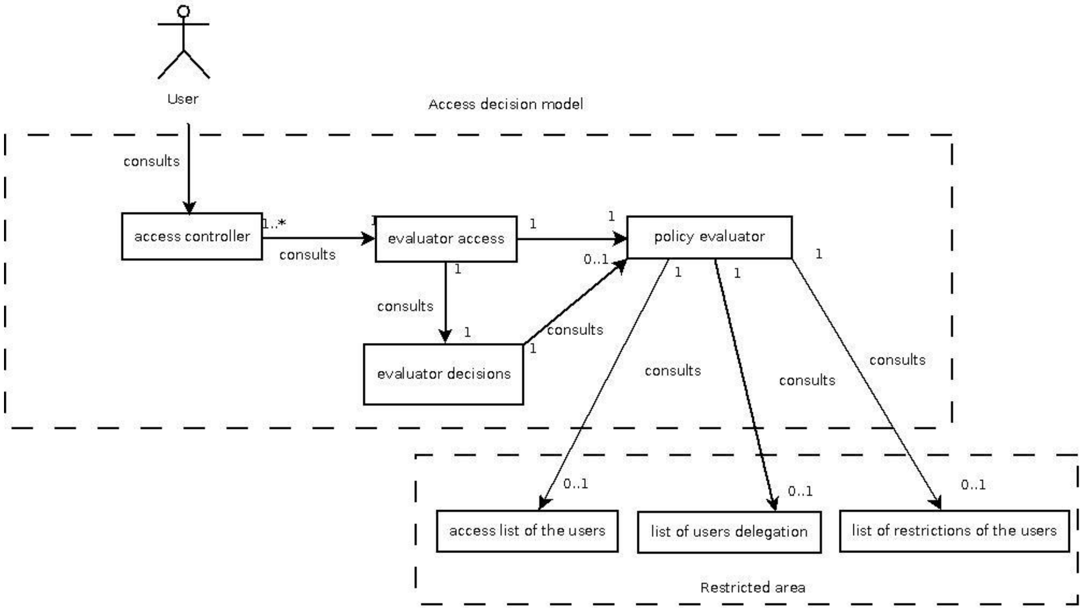

Figure 4. Access Decision Model

6. ADMINISTRATIVE MODEL

This model essentially describes how to use administrative model for implementing access control.

In the picture below we present the image that represents the elements that they manage security administrator.

First list is the list of objects to be created and followed by the list of domains, for the each domain its own groups and users of the groups with their trusted levels.

Then the administrator will create the processes flow and the sub processes with the hierarchies of actions, the hierarchies of events and stack flow.

Then for each object of the hierarchy of actions, the data will be completed by assigning an user per action, and recorded.

If a delegation of an action is required, shall be filled in the list of delegations and a new restriction will be implemented in list of user restrictions.

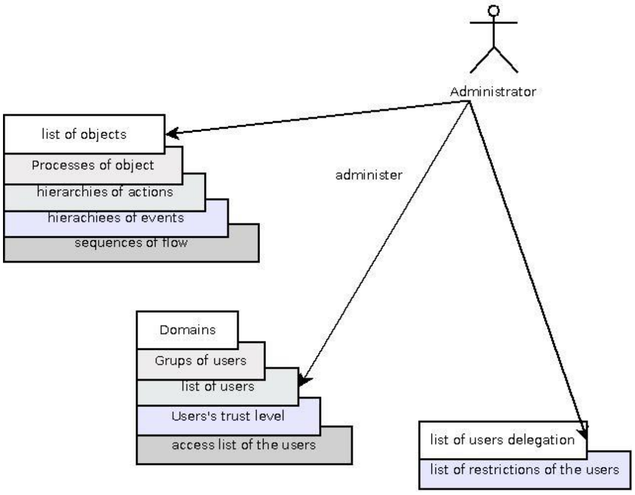

Figure 5. Administrator lists

7. CONCLUSIONS

Control access and users actions within virtual organizations or SME, it is difficult for modeling, since these organizations are dynamic in structure, with less staff and less stable functions. In order to model the control access and users actions, was made this study, this study completes a void in this regard. This paper presents a innovative path, easy to implement, allowing a security model that allows to the user to interact with the object by determining allowed actions.

For the future, we plan to refine this study and to develop a language for expressing access policies and a system modeling based on graphics, which will be then translated into XML, for an easier interpretation of the elements required by policy.

BIBLIOGRAPHY

1. Assurance model behavior in social networks based on trust. Adomnicai C., Danilescu M. [ed.] IACSIT. Chengdu, China : IACSIT, 2011. 2011 3rd International Conference on Computer technology and Development. ISBN: 978-0-7918-5991-9.

2. Control access to information by applying policies based on trust hierarchies. Marcel Danilescu, Laura Danilescu. Bucureşti : Universitatea Titu Maiorescu, 2010. Conferinţa internaţională"Educaţie şi Creativitate pentru o societate bazată pe cunoaştere" 2010. pg. 49-54. ISBN 978-606-8002-47-7.

3. Marsh Stephen Paul. Formalising trust as a computational concept. Stirling : Dept. of Computing Science and Mathematics, University of Stirling, 1995.

4. Roy J. Lewicki, Daniel J. McAllister and Robert J. Bies. Trust and Distrust: New Relationships and Realities. The Academy of Management Review - Stable URL: http://www.jstor.org/stable/259288. 23 July 1998, pg. 438-458.

5. Control Access To Information By Applying Policies Based On Trust Hierarchies. Laura Danilescu, Marcel Danilescu. Manila , Philippine : Institute of Electrical and Electronics Engineers, Inc, 2010. International Conference on Computer and Software Modeling, ICCSM 2010. pg. 285-290. IEEE Catalog Number: CFP1093L-PRT ISBN: 978-1-4244-9095-0, IEEE Catalog Number: CFP1093L-ART ISBN: 978-1-4244-9097-4.

6. D. Elliot Bell, Leonard J. LaPadula. Secure Computer Systems: Mathematical Foundations. s.l. : MITRE, 1973. 2547.

7. Denning Dorothy E. A Lattice Model of Secure Information Flow. [ed.] Chicago, IL Robert L. Ashenhurst Univ. of Chicago. Communications of the ACM. May 1976, Vol. 19, pg. 236 - 243 .
8. Marsh, Stephen Paul. Formalising Trust as a Computational Concept. Stirling : University of Stirling, 1994.

9. Ravi S. Sandhu, Edward J. Coynek, Hal L. Feinsteink and Charles E. Youmank. Role-Based Access Control Models [ed.] IEEE press. IEEE Computer. October, February 1995, Vol. 29, pg. 38-47.
10. OASIS. Extensible Access Control Markup Language (XACML), Version 2.0. www.oasis-open.org/committees. [Interactiv] 2005. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml..

11. IBM. Enterprise Privacy Authorization Language (EPAL). w3.org. [Interactiv] 2003; the version submitted to the W3C. [Citat: 20 martie 2009.] http://www.w3.org/Submission/2003/SUBM-EPAL-20031110/Version 1.2.