社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－DPS－135 (20)
2008／6／20

A Flexible Model of Access Control based on Social Relations

George Serban RADESCU†, Satoshi OGAWA†, Wenpeng WEI†,
Gen KITAGATA† , Atsushi TAKEDA†† , and Norio SHIRATORI†
† Research Institute of Electrical Communication/
Graduate School of Information Science, Tohoku University
2-1-1 Katahira, Aoba-ku, Sendai, 980-8577, JAPAN
†† Department of Computer Science, Tohoku Bunka Gakuen University
6-45-1 Kunimi, Aoba-ku, Sendai, 981-8551, JAPAN
E-mail: †{rgeos,satoshi,wei,minatsu,norio}@shiratori.riec.tohoku.ac.jp, ††atushi@ait.tbgu.ac.

Abstract—This paper introduces the notion of social relationship based access control and creation of trusted communities through negotiation and collaboration. Self interest agents interacting in the cyberspace, can chose to follow a cooperative behavior in order to gain access control over different resources. If the outcome of the interactions in a predetermined social system is more or less predictable, interactions between autonomous agents might lead to interesting results. Using concepts from our daily interactons, we try to show that authorization decisions in an agent based system, has a direct impact on the utility of the underlying system. Furthermore, we will try to maximize the utility of the system by building a negotiation and trust enhanced framework.

あらまし本稿では，社会的関係に基づくアクセス制御の概念と，交渉や協調による信頼のコミュニティを生成する手法を提案する．サイバースペースで活動するエージェントは、利己的な性質を持っている場合でも，リソースのアクセス権限を獲得するために，協力的な行動を取る場合がある．社会システムにおける相互作用の結果をある程度予測できれば，自律的なエージェント間の相互作用にって，興味深い結果が導かれる可能性がある．そこで本稿では、日常生活の交流で用いられている概念を活用し，エージェントシステムにおけるアクセス許可決定が、社会システムにおける効用に対して，直接的な影響があることを示す．また，交渉の仕組みにより，システム全体における効用を最大化する，信頼性の高いフレームワークの構築について述べる．

Index Terms—ACCESS, Access Control, Utility Maximization, Agent Computing, Social Systems, Cooperative Environments

1. Introduction

Solving complex problems like financial investment planning, foreign exchange, data mining over multiple servers and databases etc, requires many routines and subroutines, complex techniques (e.g. neural networks, fuzzy logic, genetic algorithms), and most of all it is directly related with security issues. Cooperative environments (peer to peer, e-commerce solutions, etc...) have been growing at a fast pace in recent years and along with it a new paradigm for distributed computing emerged. In this paradigm, agent based systems occupy important aspects of nowadays digital environments, recommender systems and mobile personal digital assistants. Agents are autonomous programs which accomplish tasks on behalf of their owners, and they are considered a very attractive option for building the infrastructure for e-commerce applications. The agents deployed in these applications often interact in an open environment with other agents or users (humans). These interactions involve cooperation, collaboration or competition for different resources in order to achieve the goals that they were build for.

Although agent computing has been attractive for different fields of activity, security issues became a challenge on thier own. Due to their nature (e.g. autonomous, mobile), agents pose several security related threats, as well as possibility of committing violations of social norms (netiquette). The identity of the requesting party can no longer be established with certainty; genuine agents, requesting several services, might encounter difficulties in protecting themselves while roaming over the plenitude of service providers.

Traditional access control systems cannot guarantee the behavior of authorized agents let alone prediction of malicious behavior.

2. Related Work

The concept of trust is not new to computer science. The term was introduced by **Stephen Paul Marsh** in his paper "*Formalizing Trust as a Computational Concept*".

Traditional access control policies, such as MAC and DAC have their own shortcomings.

Mandatory Access Control (MAC) ensures that the enforcement of organizational security policy does not rely on voluntary web application user compliance. MAC secures information by assigning sensitivity labels on information and comparing this to the level of sensitivity a user is operating at.

A system utilizing MAC features should at least guarantee that a user will not be permitted to change security attributes at will; all user utilities, programs and scripts must work within the constraints of the access rules provided by the selected security policy modules; and that total control of the MAC access rules are in the hands of the system administrator.

Discretionary access control, or DAC, is a form of access control to resources (directories and files) in which individual users can be assigned a unique type of access. "A means of restricting access to objects based on the identity of subjects and/or groups to which they belong. The controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject (unless restrained by mandatory access control)"[1].

Mandatory Access Controls are considerably safer than discretionary controls, but they are harder to implement and often require consideration tweaking to ensure all applications function correctly.

Prior to the development of RBAC (Role Based Access Control), the previous two models where the only ones. If a model was not MAC was considered DAC and vice-versa.

RBAC is based on roles created within a certain organization for various job functions. Dif-

ferent roles have different access controls which simplifies the task of assigning permissions to users by affiliating them to a specific group.

For large systems, with hundreds of roles, thousands of users and millions of permissions, managing roles, users, permissions and their interrelationships is a formidable task that cannot realistically be centralized in a small team of security administrators.

Here is where our proposal comes to fill in the shortcomings of the previously mentioned systems.

3. Access Control in Cooperative Environments utilizing Social Systems

3.1 Agent negotiation

In our daily activities we create social networks with individuals as nodes. Unconsciously, we grade these nodes based on personal rules and values and in this way we create an Access Control (AC) system in which we allow nodes – friends – with high rating or trust to use some of our resources. At the same time we disallow low rated nodes to access our resources.

Not intending to reflect an exact real life situation, let's take the example of Alice, Bob, Claire and Dave (A, B, C and D):

- ✔ social relationship: classmates;
- ✔ common goal: school physics project;

In achieving the common goal, the team needs to share several resources: someone's house, a study room, 2 PCs (PC_0 & PC_1) and a whiteboard(WB).

Alice's parents (P_0 & P_1) don't know Bob and Claire so normally they wouldn't be allowed to enter the house. However, if Alice is at home, due to the social relationship between classmates, Bob and Claire are allowed inside.

PC_0 has important data for the project and at the beginning only Alice will be allowed to use it. Since Bob is good at taking notes and making logic schemes, Alice will grant access to him to resources PC_1 and WB. But Claire is only good at making logic schemes, so she will be allowed to use only the WB resource. Like this we have a cooperative environment in which the drive for the common goal is the main rule of the AC system. If we consider Dave which is also a classmate of the three (Alice, Bob and Claire), but he is in a rival group, then the highest access level he will get to, is the resource house. He is not going to be a cooperative user in achieving the common goal, therefore he won't get access to resources designated for users given a higher level of trust.

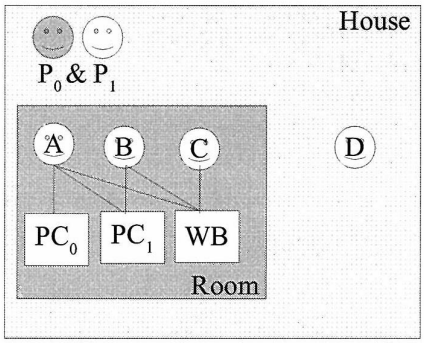


fig. 01 – resource allocation

Alice has to be present in person at home in order for the authentication and authorization process to take place. If Alice is not present, then there is no social relationship which can be used for authentication and authorization, therefore we will have the situation in fig. 02.

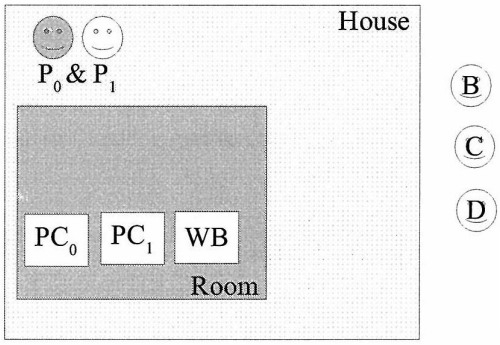


fig. 02 – failed authentication & authorization

With this hypothetical example we tried to exemplify the fact that collaboration can bring higher utility for the individual as well as for the group.

Furthermore, the Internet is known to have a long culture of lies. It is possible that one of the agents will use as strategy to lie in order to maximize its own utility without thinking at the group's utility. Trust implies some degree of uncertainty, hopefulness or optimism, regarding the outcome. This trust problem can be solved in a trust negotiation scheme.

3.2 Trust Negotiation

This concept is divided in two parts – one that is focused on the interaction between the user and the system (function F_0) and the other one on the interaction between the users (function F_1).

In the case of F_0, Alice can do anything she wants within the rules imposed by the system (e.g. no loud music, no work after 10 PM etc ...). If her behavior is in concordance with the system requirements, her rating will automatically increase, and therefore be granted access to more resources within the system. Otherwise, she is automatically downgraded and access rights will be stripped off. It is also in her best interest to create and invite reliable social nodes so that she can upgrade her ranking within the system.

We can portray this dynamic system on a scale of trust like below:

Class	Trust (%)	Resources
Expert	81-100	House, Room, PC_0, PC_1, WB
Pro	61-80	House, Room, PC_1, WB
Beginner	41-60	House, Room, WB
Newbie	21-40	House, Room
Outsider	0-20	House

$Class_{Expert}$={Alice, P_0, P_1}
$Class_{Pro}$={Bob}
$Class_{Beginner}$={Claire}
$Class_{Outsider}$={Dave}

During the project or once it is complete, Bob, Claire and Dave can be downgraded or upgraded depending on their behavior or higher authority will. All this happens automatically without the administrator's interference, based on the preset rules of each social system (House or Room).

Example of system rules – pseudocode:

```
System rules {
        If User(A) in House {
          If User(B) related with User(A)
            User(B) assigned to Class_Outsider;        (1)
          Else exit;
        }
        Else exit;

        If User(A) work > 10 PM
          Class(User(A)) = - - Class(User(A));         (2)
        Else Class(User(A)) = ++Class(User(A));


        User rules {
          If User(B) has common project
            User(B) assigned to - - Class(User(A));    (3)
          Else exit;
        }
}
```

We define two types of interactions between agent: vertical ones and horizontal ones. The vertical type of relationship is based on power and submission, whereas the horizontal one is the collaborative and trust based relationship. We encounter this kind of behavior in our daily interactions with our superiors – vertical type – and with our friends – horizontal one. Each and everyone is free to prioritize any of the two.

In the above pseudocode, the rules were build with the vertical type of relationship as first priority. Therefore, the agents that chose to interact with each other, have to obey first the rules of the system – in this case the rules of the house imposed by P_0 and P_1 – and as second priority the rules of the small social environment – in this case the rules of Alice's room imposed by Alice.

The rules above are the House rules imposed to Alice and everybody interacting within the boundaries of it. In the first block a check is being made if Alice is or not inside the system at a certain moment in order to verify the social relationship between A and B, C. If the social relationship is not being authenticated, then the system will reject the AC request.

The second bloc is another rule imposed to A by the system which states that user A should be downgraded in case it violates a certain rule, otherwise should be rewarded.

The third bloc is a rule made by the user. Notice that user A choses in the first place to obey the system rules, therefore its rules are incapsulated in the rules of the system. In case user B has a common project with user A, user A can chose which class to assign user B in order to achieve the common goal.

The agent's behavior is recorded at all times and it is used in later interactions to grant authorization to resources. This process will be explained in the following chapter.

3.3 The Authentication Model utilizing social systems

The proposed method to be implemented in the previous model is taking into consideration the fact that decisions to grant access will be based on future expectations.

The prototype system is divided in independent modules interacting with each other. The scheme is represented and explained below.

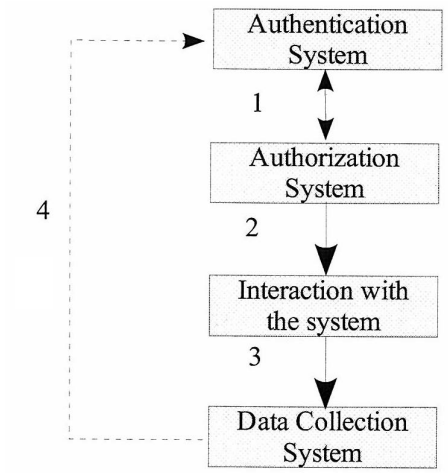
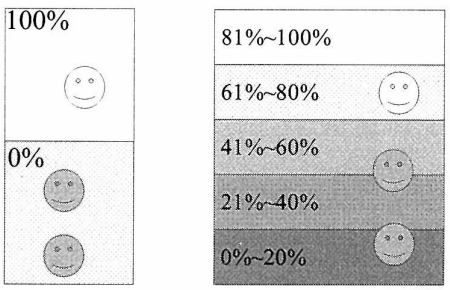


fig. 03 - proposed system

A very important factor of decision in granting access or not is the historical data - past interactions should be recorded and evaluated according to their satisfactory level. A function will update the history of the agents after each interaction which it will play a key role in future access control decisions.

The process has 4 steps (fig. 03). Step 1 consists in the interaction between the *Authentication System* and *Authorization System* whenever an autonomous agent asks for permissions. The Authentication System has a feed-forward function which informs the Authorization System about the value of risk in granting permissions to an agent. An agent reaches step 2 once access was granted to it. At all times, a Data Collection System will record and analyze agent's behavior (step 3) and at the same time will check for malicious behavior. The data will be fed back to the Authentication System for future evaluations (step 4).

Considering the hypothetic situation in which an agent, flagged as undesired according to the historical data, is trying to improve on its behavior, the existing system still wouldn't allow it to have access to resources. Giving as example the login process, it is obvious that if the password string doesn't match, the authorization process fails. However, implementing the social based AC, the agent still has a chance to redeem itself and gain access to certain resources.



a) nowadays system *b) proposed system*

fig. 04 - system comparison

In real life, before taking a decision, we consult with our friends, relatives, people with experience which we can trust or we had satisfactory interactions with. Therefore, it is important to ask

for the opinion of other agents before taking a decision. Veteran users with whom we had interactions with, tend to be more trustworthy. Therefore, time of acquaintance will be a parameter in the AC decision making process.

Agents should be endowed with self-estimate abilities. Therefore, a new parameter should be the *quality of work* based on their *curriculum vitae* (CV). Time estimates will reflect the possible time of completion, whereas quality estimates will reflect the possible performance of an agent to do a certain job.

Another input parameter that we should consider is *recommendations*. Recommendations play an important role in social relationship based AC, especially if it comes from a trusted party. Even in real life we tend to trust more an official act signed by a prestigious institution. We can try to apply the same rule in virtual world.

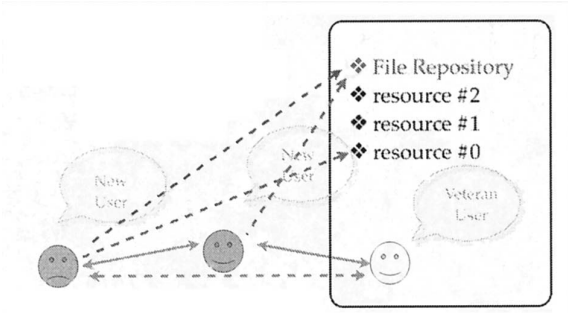
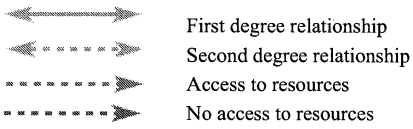


fig. 05 – social based AC



The system will treat a certain new user with a low level of trust at the beginning, but due to the previous successful interactions, a veteran user will have a different rating, therefore it will allow the new user to access important resources in order to achieve a common goal. A second degree relationship won't be treated the same as a first degree relationship. (fig. 05)

4. Design And Implementation

The prototype system will be client-server architecture based on LAMP (Linux, Apache, MySQL, PHP). However, due to its modularity it is easily portable on other platforms.

Small scale simulations revealed interesting results regarding the interaction between users and the system. With simple set of rules, the system is able to automatically upgrade the user based in its contribution to the society, therefore making the intervention of the administrator not necessary. The results are not yet conclusive since the prototype itself is still under development.

In this framework, we intend to introduce not only the above mentioned parameters but also other competitive decision making mechanisms to determine the level of effectiveness of different agent interaction strategies in a dynamic behavioral environment. It is also important to evaluate the performance variation and the stability of relationships between agents as well as the effects of task failure.

Once our system will have its modules in place, we will investigate if our suppositions were true by gathering statistical data and plotting the results.

5. Conclusions

We've proposed a social relationship based access control system, which we believe it will be a milestone in further development of access control systems. Although the proposal is not backed up by solid facts and tests due to its novelty, we believe we can achieve interesting results trying to implement the above explained model.

In this paper we hypothesized that agent encounters leading to cooperative behavior can maximize the utility of the underlying system and at the same time can bring to the creation of solid, trust based communities in which malicious agents will be swayed out.

6. Reference

[1] "*Trusted Computer System Evaluation Criteria*" - http://www.radium.ncsc.mil/tpep/library/rainbow/5200.28-STD.html

[2] *"The Bargaining Problem"* - John F. Nash Jr.; Economietrica Vol.18, No.2 (Apr. 1950) pp. 155-162

[3] *"When Cyborgs Meet: Building Communities of Co-operating Wearable Agents"* - Gerd Kortuem, Jay Schneider, Jim Suruda, Steve Fickas, Zary Segall; University of Oregon

[4] *"Trust negotiation in Identity Management"* - Elisa Bertino, Anna Cinzia Squicciarini, Abhilasha Bhargav-Spantzel; IEEE Security & Privacy; March/April 2007

[5] *"Maximizing Utility of Mobile Agent Based E-commerce Applications with Trust Enhanced Security"* - Ching Lin, Vijay Varadharajan, Yan Wang; Springer 2005

[6] *"Emergence and Stability of Collaborations Among Rational Agents"* - Sandip Sen, Partha Sarathi Dutta, Sabyasachi Saha, Springer 2003

[7] *"Risk Aversion and Expected-Utility Theory: A Calibration Theorem"* - Mathew Rabin; Econometrica Vol. 68, No. 5 (Sep. 2000), pp 1281-1292

[8] *"Expected Utility Theory"* - Philippe Morgin; Handbook of Economic Methodology (1997), pp. 342-350

[9] "Interactive Design Environment for Agent system" (Japanese only) - http://www.ka.riec.tohoku.ac.jp/idea/index.html

[10] "Distributed Agent System based on Hybrid architecture " (Japanese only) - http://www.agent-town.-com/dash/; the manual http://www.ka.riec.tohoku.ac.jp/idea/html/index.html

[11] "Access Coontrol in Cooperative Environments utilizing Social Systems" - George Serban Radescu et al., Technical Report IEICE (2007)

[12] "A flexible and Secure Access Control Scheme using Social Behavior in the Real World" - Satoshi Ogawa et al., Technical Report IEICE (2007)