Agent-based Access Rights Delegation utilizing Social Relationships

Gen Kitagata†, Kazuto Sasai†, Johan Sveholm†,
Norio Shiratori† and Testuo Kinoshita†

†Research Institute of Electrical Communication, Tohoku University, Sendai, Japan

Abstract

Access control to resources is one of the most important issues for supporting human activities in the digital space. However, existing access control methods are not effective for temporal activities such as visitor access. In this paper, we focus on visitor access control nature in real space, and propose a novel access control scheme utilizing social relationships which is effective for temporal activities. We also evaluate capability of our scheme through experimental results using a prototype system.

Keywords: access control, access rights delegation, socialware, agent, collaborative work

1: Introduction

Access control to resources is one of the most important issues for supporting human activities in the digital space[4][8][13]. Growth of information communication technology makes people's activities in digital space more popular than ever. Digital space is a kind of societies where people participate and interacts. Same as in real space, people in digital space should recognize society, and be able to take actions without anxiety and discomfort. Socialware [11] is a software technology to support people's activities in digital space by enhancing social reality. Socialware has two goals: 1) apply existing rules and knowledge used in real space to activities in digital space, and 2) create new knowledge and rules which are specific to digital space. Such knowledge, which is involved with social activities, is called Social Knowledge, and it will be important to enhance people's social reality in digital space. Based on the concept of socialware, we propose a novel access control scheme which realizes access rights delegation so as to achieve flexible access control to resources in digital space, which is indispensable for activities in digital space,

Access control methods are crucial technologies to enable people to act in digital space safely. Hence there are several efforts to achieve effective access control in computer system. RBAC (Role-Based Access Control) introduced a concept of role to reduce management task of access control configuration [1][9][15][17]. TRBAC (Temporal Role-Based Access Control) introduced the time constraints and role dependencies to deal with temporal roles [2][10]. Context-aware Access Control methods introduced the context-based constraints such as user's location and device availability to deal with dynamic change of access rights according to context [3][12][14][16]. However, there is a limitation that these methods are not effective for occasional situation. For example, in real space, when a visitor, e.g. a research partner belonging to another organization, comes to your office, he may be able to enter your office whenever there is a native member who share a certain social relationship, such as a member of the same research project. In this case, it can be understood that a kind of access rights such as "entering office" is dynamically delegated

from the native member to the visitor under the responsibility of the native member. On the other hand, it is difficult in digital space to deal with such an occasional visitor because existing access control methods lack the capability to delegate one's rights according to the presence of a guarantor.

In this paper, to overcome the above limitation, we propose an automated access rights delegation scheme which utilizes social relationships between native members and visitors. To manage presence of a guarantor explicitly, we introduce a personal agent and an authority place. The personal agent in digital space represents a corresponding person in real space, and if the personal agent exists on authority place, it means corresponding person is in real space. This personal agent and authority place make our scheme possible to give access rights to the visitor temporarily as long as his guarantor exists.

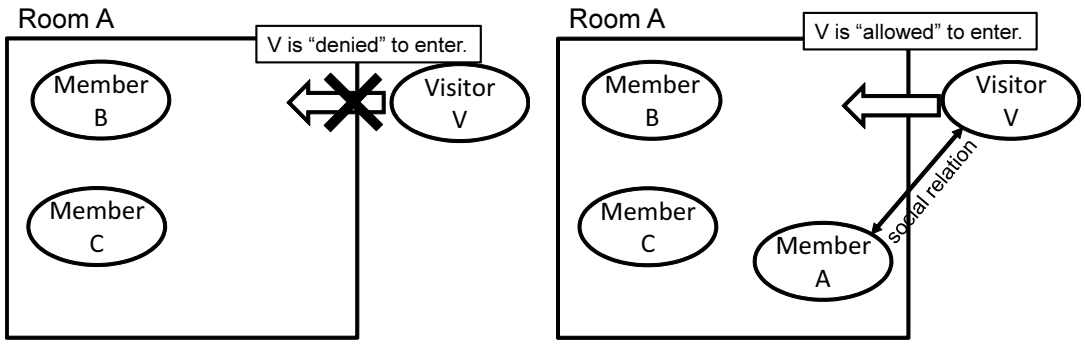
The rest of this paper is organized as follows. In Section 2, we introduce related work on access control and issues. The proposed scheme and its model are described in Section 3. Section 4 presents a an application of the proposed scheme, i.e., collaborative work support system with the proposed scheme, and Section 5 presents experiments and discussion with the system. Finally we conclude our work in Section 6.

2: Related work

RBAC (Role-Based Access Control) is one of the existing access control schemes, where users are assigned with roles, and roles with access rights[1][9][15][17]. This scheme has an advantage of management cost because roles are, in general, likely to be associated with positions in an organization. However, this scheme requires static configuration of acceptable roles in ACL (Access Control List). And it has to be done manually to add, change, and delete users and roles. Therefore, this scheme is effective where roles do not change frequently. However, in fact, there is a case where some users' accesses should be enabled temporarily. In the latter case, an administrator will be burdened by emergent update of ACL. And also there might be an issue of safety such that the administrator forgot to disable temporal access rights afterwards.

TRBAC (Temporal Role-Based Access Control) is an access control scheme for dynamic and temporal changes to access rights assigned to roles[2][10]. TRBAC introduces time constraints and role dependencies to its scheme to deal with temporal roles. For example, assume that a person works part-time for some company from 9 a.m. to 1 p.m., and a role 'part-time-staff' is assigned to him. In this case, an administrator can activate the role 'part-time-staff' from 9 a.m. to 1 p.m. in order to give the staff an access right to the company's system with time constraints. In addition, the validity of a certain role can be controlled in response to the condition of other roles, which is called role dependencies. For example, a role 'nurse' can be active only if a role 'doctor' is active. With time constraints and role dependencies, TRBAC effectively deal with regular activities. However, TRBAC cannot deal with unexpected activities, such as occasional meetings caused by emergent situations. The administrator has to make changes to roles for temporal or emergency activities, and the higher the frequency of such activities, the heavier the workload.

Context-aware Access Control method was proposed to deal with ubiquitous computing environments [3][12][14][16]. This method has the ability to dynamically adjust access control decisions in order to adapt to changes of the situation or state of an entity such as location of a mobile user, availability of a device and etc. However, this method has a limitation that it can consider only the situation of the entity. In other words, it cannot adjust the decisions according to situations of other persons such as a guarantor to the user because it has no mechanism to manage presence of the person explicitly.



(a) No member has social relationship with *Visitor V*. (b) Person *Member A* has a social relationship with *Visitor V*.

Figure 1. An implicit rule of temporal visitor access in real space.

In addition, FIPA shows access rights delegation scenarios[5], and some access rights delegation methods are discussed [6][18]. These methods allow users to delegate their own rights to other persons, but they do not consider presence of a guarantor.

To summarize the above-mentioned existing methods, it is needed to realize more flexible and dynamic access grant scheme for temporal and emergent activities considering the presence of a related person such as a guarantor.

3: Proposal of Agent-based access rights delegation scheme

3.1: Access control in real space

To overcome the limitation of existing access control methods, we propose an agent-based access rights delegation scheme utilizing social relationships. The idea of this scheme is inspired from implicit access control nature in real space. In real space, there is a common rule such that a person belonging to an external organization can get a temporal access right under responsibility of a guarantor who already has access rights to a certain resource.

Figure 1 shows a case of emergent visitor access in real space. In the figure, *Member A*, *Member B* and *Member C* are native members of the laboratory L, whereas *Visitor V* is not a member. Let us suppose *Visitor V* is involved in a cooperative research project with *Member A*, and *Visitor V* wants to enter L. As shown in Figure 1(a), due to the fact that *Member A* is not in the laboratory, *Member B* and *Member C* are unable to identify *Visitor V*. Therefore, *Visitor V* will not be allowed to enter the laboratory L. On the other hand, as shown in Figure 1(b), when *Member A* is in the laboratory, *Member B* and *Member C* can identify *Visitor V* through his social relationship with *Member A*. Therefore, *Visitor V* is allowed to enter the laboratory L. In this way, even if a certain resource, i.e. the laboratory space L in this case, that normally would be out of reach for an outsider, *Visitor V* can obtain access rights to the resource due to the social relationship.

While *Visitor V* is inside the laboratory, *Member A* becomes a guarantor to *Visitor V* and is responsible for his behavior. For that reason, *Visitor V* can have the same or less access privileges as *Member A*. It also can be said that a kind of access rights such as "entering laboratory" is dynamically delegated from the native member to the visitor under the responsibility of the native member due to a social relationship.

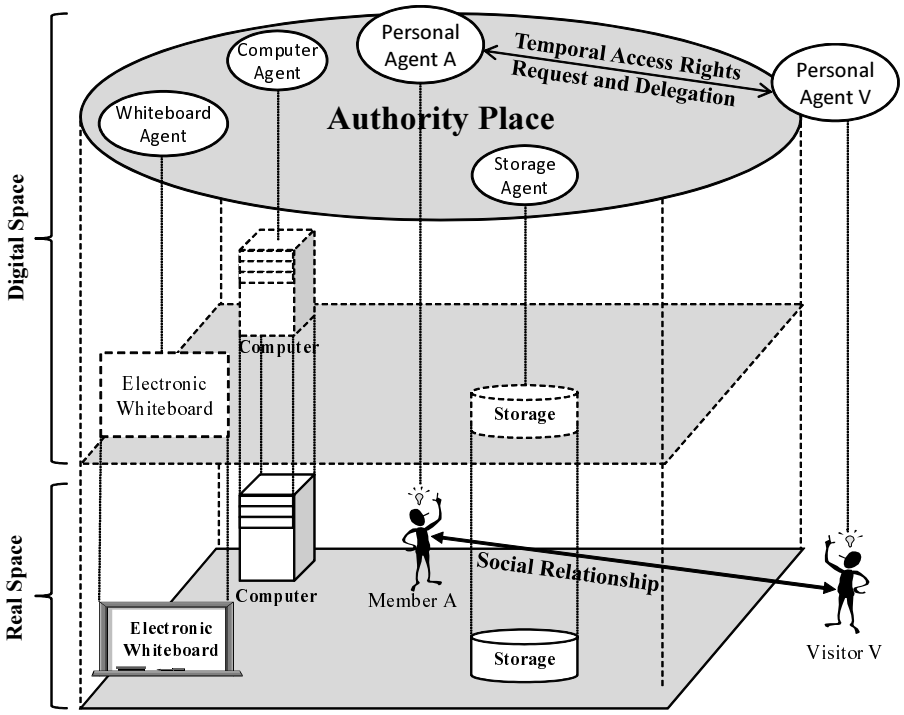


Figure 2. Personal agents and authority place in digital space.

3.2: An access rights delegation scheme utilizing social relationships

In a similar way as of real space, we propose a novel access control scheme in digital space with a notion of access rights delegation based on social relationship. A remarkable feature of our scheme is that it can explicitly manage the presence of a guarantor at the location where a certain job or task is being undertaken. Therefore, we consider the following two functions are important:

- F1 - explicit representation of people in digital space.
- F2 - delegation of rights based on social relationship.

To realize these two functions, we newly introduce the following components:

- A personal agent and an authority place.
- Social relationship based access rights filtering and delegation mechanisms.

3.2.1: Personal agent and authority place

In real space, our access rights to certain resources are determined dynamically depending on the presence of a guarantor. In order to introduce this concept into digital space, we have to explicitly state the presence of people and their working places in digital space. To realize this, we introduce a personal agent and an authority place. Figure 2 shows the relation among personal agents, the authority place and digital space. So far, this has not been considered by existing access control models. The personal agent represents a corresponding person, and if the personal agent exists in authority place, it means the corresponding person is present. By introducing the personal agent and the authority place, we can now define whether a certain person is present in the place or not. The authority place in this case is what we defined in Section 3.1 as L. Inside the authority place,

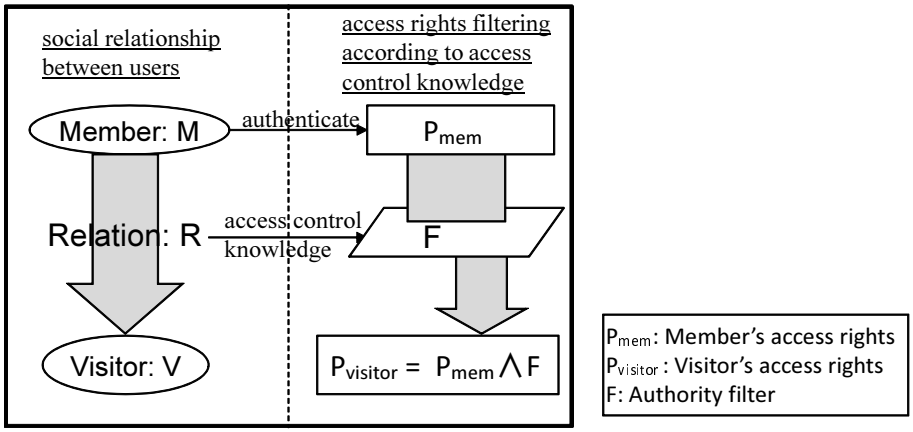


Figure 3. Delegation of access rights with authority filter.

there are several resources like a computer, storage, a printer etc., which are also represented as resource agents.

3.2.2: Delegation of access rights based on social relationship

When *Member A*, who is a guarantor, delegates his access rights to *Visitor V*, the following requirements should be satisfied:

- The delegation of rights occurs in conformity with the social relationship between *Member A* and *Visitor V*.
- *Visitor V* can get the same or less access privileges as *Member A*.

In order to meet the requirements, we introduce an authority filter which limits the access rights to be delegated from *Member A* to *Visitor V*. The access rights of *Visitor V* will be the intersection between the set of rights of *Member A* and set of rights accepted by the filter. The control knowledge, i.e. access control rules, consists of social relationships and filtering conditions. As shown in Figure 3, we need to setup the authority filter based on access control knowledge in order to delegate access control rights based on social relationships.

If the access rights delegation occurs without limitation, it may cause degradation of trustworthy. Therefore, we introduce the concept of "delegation allowance". Only a person with delegation allowance can delegate his access rights to others.

3.3: Design of the proposed scheme

3.3.1: Design model

In our scheme we introduce an access control model which consists of the following seven elements, i.e., *resources, users (visitor and member), access rights, authority filter, social relationship, authority place and access control knowledge*. Figure 4 shows our access control model. We will explain it in detail as follows.

- Resource I: computing or information resources.
- Visitor V: a visitor who wants to use the resources but is not having access rights regularly.
- Member M: a member of the organization who already has access rights for the resources.

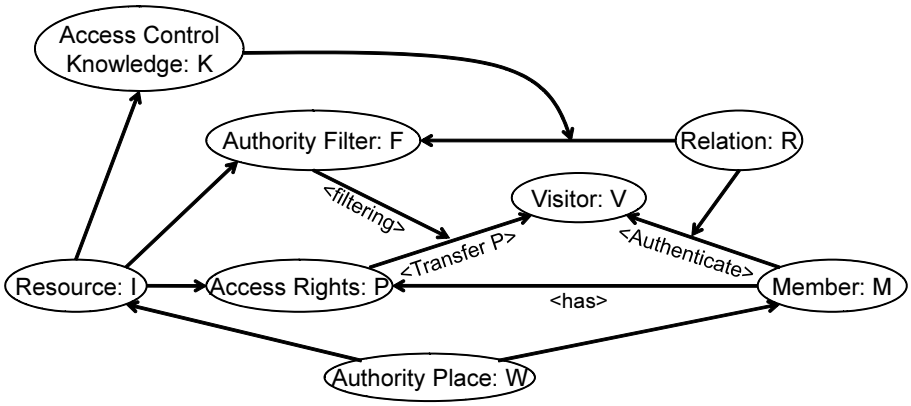


Figure 4. Access control model of our scheme.

- Access rights (permission) P: a set of permissions for the resources.
- Authority filter F: a set of access rights which can be delegated.
- Social relationship R: a social relation between persons (e.g. if there is a certain research project, the participants of that projects are bound by social relation).
- Authority place W: a place where personal agents and resource agents are located.
- Access control knowledge K: this knowledge is used to choose authority filter utilizing social relationships between the persons.

When a Visitor V wants to use a Resource I, the Visitor V choose a Member M who shares a Social relationship R. After the Member M authenticates the Visitor V, the Member M requests access rights delegation for the Resource I via a Authority place W. The Resource I generates Authority filter F by referring both Access control knowledge K and the Relation R. Then Access rights P, which is a subset of Member M's Access rights, are transferred to the Visitor V. Here, the Access rights P are not a combination of Access rights of several members, but Access rights of only one member. Thus our method can prevent a breach of security or unauthorized access caused by a combination of several access rights.

Using above model, access rights given to the Visitor V can be decided as subset of Member M's Access rights P according to the Social relationship R.

3.3.2: Authority place

The authority place is a one of the important elements which manages the presence of members and resources available in digital space. We define an instance of authority place $w(\in \mathrm{W})$ as follows:

$$w = \langle m_{list}, i_{list} \rangle$$
$$m_{list} = \langle m_0, m_1, ..., m_n \rangle$$
$$i_{list} = \langle i_0, i_i, ..., i_m \rangle$$
$$w \in \mathrm{W}, m_n \in \mathrm{M}, i_m \in \mathrm{I}$$

In this case, m_{list} and i_{list} are sets of personal agents and resource agents in an authority place respectively. In other words, our authority place is formed of personal agents and resource agents. The personal agent's presence in authority place might change according to member's log-on and

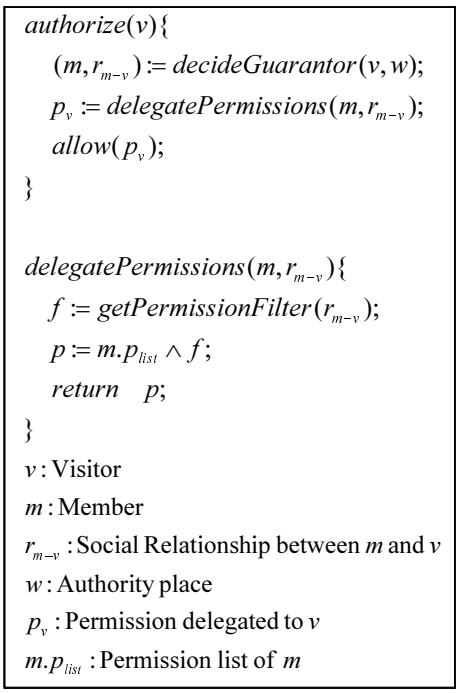


Figure 5. Algorithm for access delegation.

log-off activities. Therefore w also changes reflecting the presence of members. A personal agent m is defined as:

$$m = \langle d_m, rv_{list}, p_{list} \rangle$$
$$rv_{list} = \langle rv_0, rv_1, \cdots, rv_n \rangle$$
$$rv_k = \langle r_k, v_k \rangle$$
$$p_{list} = \langle p_0, p_1, \cdots, p_n \rangle$$
$$r_k \in \mathrm{R}, v_k \in \mathrm{V}, p_k \in P$$

Here, d_m represents the member's data and rv_{list} represents the set of social relationships of the member and a visitor.

3.3.3: Delegation of access rights

In our scheme, when a visitor v wishes to use resources in an authority place w, at first v should search and select a member m with whom v has a social relationship r. Then v should be identified and authenticated by m. After that, the authority filter is generated using access control knowledge and the social relationship between v and m. As a result, the access rights p_m held by a member m is intersected with an authority filter set f, and a set of access rights p_v which is delegated to a visitor v is decided. The process of delegation of access rights from the member m to the visitor v is shown in Figure 5. This algorithm starts with the selection of the socially related member by the $DecideGuarantor$ function. Then the $delegatePermissions$ function calls the $getPermissionFilter$ and returns the proper access rights. In Figure 6 we show how the access knowledge works. In this case, if a visitor v and a member m have a relationship such as "CooperativeResearcher", access rights $p1, p2, p3$ and $p4$ are delegated to v. Otherwise if the relation is

```
getPermissionFilter(r_{m-v}){
  if (r_{m-v} =="CooperativeResearcher")
    return[p1, p2, p3, p4];
  elseif (r_{m-v} =="OldBoy")
    return[p3, p4];
  elseif (r_{m-v} =="Friend")
    return[p4];
  else
    return[];
}
```

Figure 6. Access control knowledge for a laboratory.

"OldBoy" or "Friend", limited access rights are delegated.

3.4: Process of access rights delegation

A process of access rights delegation is shown in Figure 7. Personal Agents A, B, V represent *Member A*, *Member B* and *Visitor V* respectively. Here we assume *Visitor V* wishes to use a resource which is controlled by resource agent 1. Also we assume that *Visitor V* and *Member A* participate in the same research project, so they have a social relationship. The detailed process of access rights delegation is as follows:

1. Personal Agent V sends a request for access rights to Personal Agents in the authority place.
2. The Personal Agents of members who have any social relation with *Visitor V*, in this case Personal Agent A, reply to it.
3. Personal Agent V finds *Member A* and *Visitor V* to have a social relation, then chooses Personal Agent A as a guarantor.
4. *Member A* becomes the guarantor to *Visitor V*, and Personal Agent A tries to authenticate *Visitor V* by a certain authentication method.
5. If the authentication is succeeded, Personal Agent A requests delegation of access rights to Resource Agent 1.
6. Resource Agent 1 decides access rights which can be delegated from *Member A* to *Visitor V* by following filtering steps described in Section 3.3.3. Then access rights are gives to Personal Agent V.

After Step 6, when *Member A* leave the authority place, access rights delegated to *Visitor V* are revoked immediately by Resource Agent 1, and the authorization process is restarted from Step 1.

4: Application of the proposed scheme

4.1: Overview of the system

To confirm effectiveness of our scheme, we designed and implemented a cooperative work support system with the proposed scheme. This system realizes the access control for resources owned

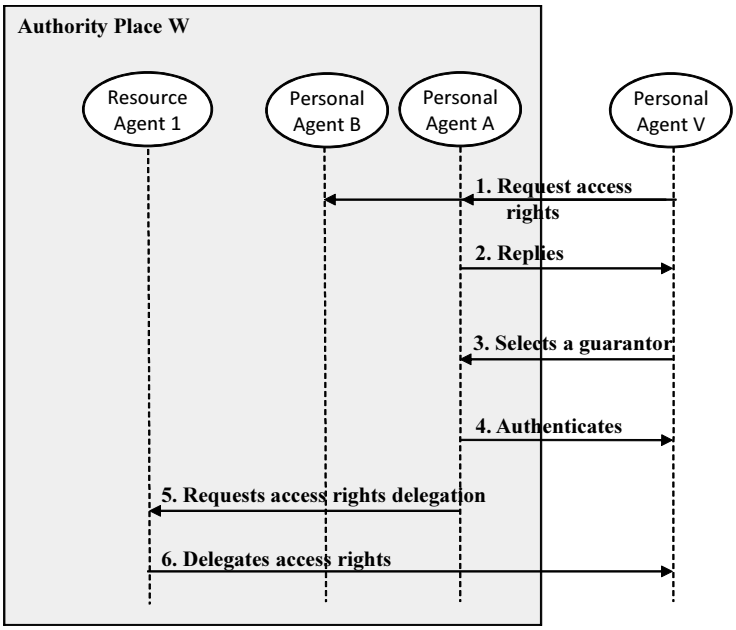


Figure 7. Flow of access authorization.

by organizations, and also the system can be applied for emergent activities of members and visitors by utilizing social relationships. Figure 8 shows a functional structure of the system. In Figure 8, there are organization X and Y. These organizations proceed cooperative project. The social relationship "cooperative project member" is constructed among persons who have joined the project. Due to this social relationship, a person who is a member of both organization Y and the project can use the resources in organization X. In contrast, a person who belongs to organization Y but is not a member of the project cannot use the resources in organization X. Also a visitor who does not belong to both organization X and Y, and has no relationship with member of organization X cannot use the resource in organization X.

4.2: Agent organization and implementation

We designed and implemented the prototype system as a multi-agent system. The system consists of three types of agents as follows:

1. Personal Agent: This agent represents a person in digital space. It uses resources instead of a real person according to the person's request.
2. Resource Agent: This agent represents resources and holds access control knowledge and an authority filter. For instance, we implemented a resource agent which represents a printer and a projector.
3. Authority place Agent: This agent realizes an authority place. It keeps track of the presence of personal agents and resource agents, and also mediates messages among personal agents and resource agents.

Figure. 9 shows the agent organization of the prototype system. In the Figure. 9, Personal Ag A, B and V are personal agents of *Member A*, *Member B* and *Visitor V* respectively. Authority Place Ag is an authority place agent. The authority place is administrated by Authority Place Ag.

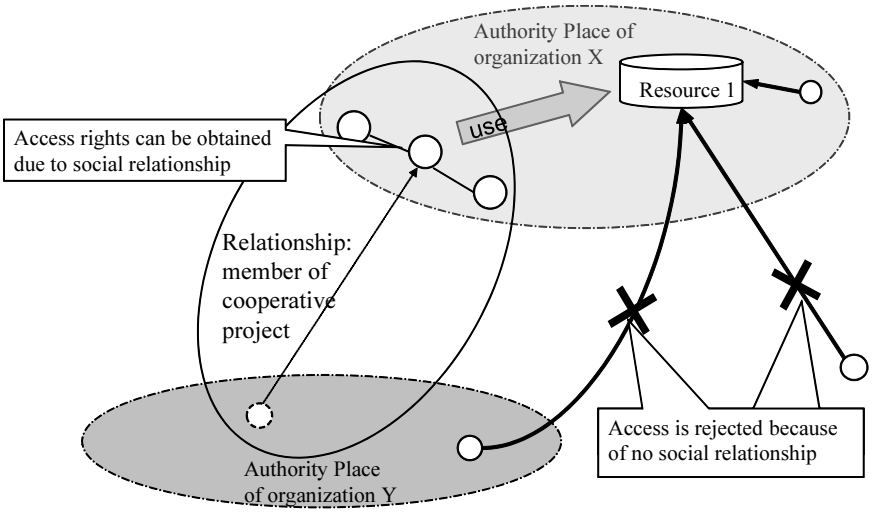


Figure 8. A functional structure of cooperative work support system.

Table 1. Performatives of agents in the prototype system.

Agent	Performatives	Descriptions
Personal Agent	requestDerivation	Request access rights for a Resource Agent.
Personal Agent	askForGuarantor	Ask for Personal Agents of members being guarantor of this agent.
Personal Agent	relatedUsers	Inform guarantor chose by this agent.
Authority place Agent	requestAuthentication	Request authentication of the visitor for the guarantor.
Personal Agent	authenticationSuccess	Inform that the visitor is authenticated successfully.
Personal Agent	authenticationFailure	Inform that the visitor is not authenticated.
Resource Agent	derivationResult	Inform delegated access rights to the visitor.

Here, presence of a person is represented as the existence of a personal agent in authority place. For example, when a personal agent of *Visitor V* sends a message of access rights requirement for a resource but no response is returned, it implies that there are no member who has social relationship with *Visitor V*, and as a result access for requested resource is denied.

Each agent communicates with KQML (Knowledge Query Manipulation Language) based agent communication language. Major performatives used in the prototype system is shown in Table 1.

We implemented the system by using the DASH [7] system which is rule-based agent framework, and IDEA [19] which is an integrated design environment for DASH. We used the Java language to implement base processes controlled by DASH agents. We implemented the agents described in Figure. 9 with 37 Java classes, which have totally 898 steps for agents and 4783 steps for Java classes.

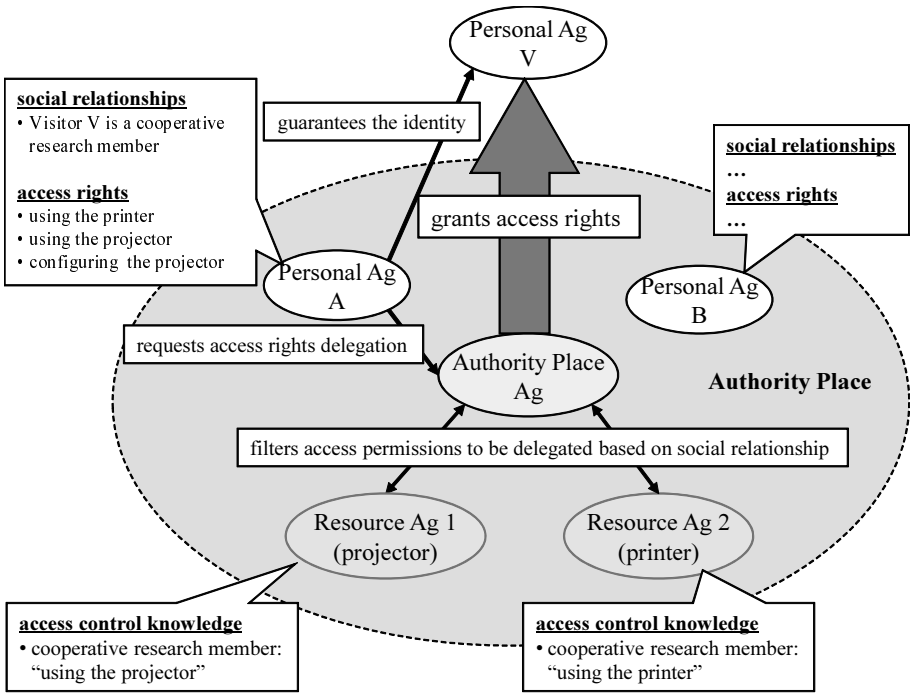


Figure 9. Agent organization of prototype system.

5: Experiments and evaluation

To evaluate our scheme, we conducted experiments with some scenarios under certain conditions depicted in Figure 10. *Member A* and *Member B* are members of a laboratory, and *Visitor V1* and *Visitor V2* are visitors. *Visitor V1* and *Member A* are members of the same cooperative research group. *Visitor V2* and *Member B* are also members of another cooperative research group. So there are social relationships according to these cooperative research groups. In addition, *Member A* is a student of the laboratory and *Member B* is a staff of the laboratory, so *Member B* has more access rights than *Member A*. We assume the situation that a visitor comes to the laboratory's authority place to proceed cooperative research. Here, we conducted experiments as followings: (1) no one is in the laboratory, (2) only *Member A* is present, (3) only *Member B* is present, (4) both *Member A* and *Member B* are present.

In each scenario, the Authority Place Ag and the Resource Ags are always activated. On the other hand, the Personal Ag A and B are activated in accordance with presence of *Member A* and *Member B*. In the experiments, the Personal Ag V begins access rights delegation process through the steps described in Section 3.4. This process was finished within 0.8 seconds for all scenarios.

Figure 11 shows the experimental results. We confirmed that both *Visitor V1* and *Visitor V2* could obtain access rights by a social relationship as "cooperative research member", but the access rights of *Visitor V1* and *Visitor V2* were not the same. This is because *Visitor V1* and *Visitor V2* are delegated access rights from different members. On the other hand, in scenario (1), we found that no access rights are delegated for both visitors. In this scenario, since no member with social relationships to the visitor is present, the system cannot verify identity of the visitor, and no access rights are delegated. In other words, absence of *Member A* means that cooperative work of *Member A* and *Visitor V1* cannot be proceeded. Therefore *Visitor V1* cannot get access rights when

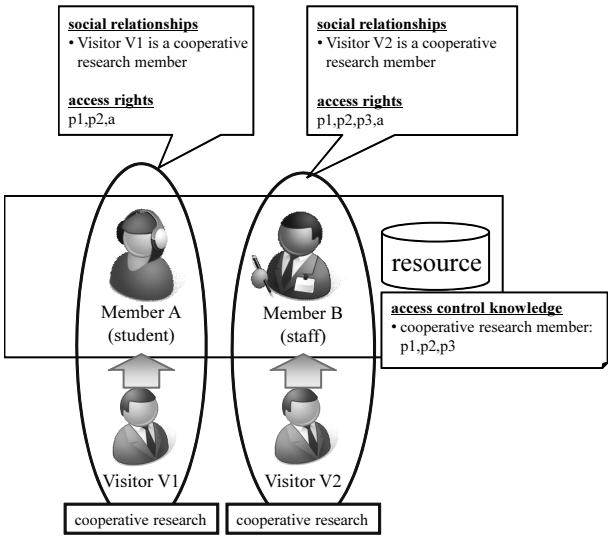


Figure 10. Condition of experiments for evaluation.

	Visitor V1	Visitor V2
No member exists	-	-
Only member A exists	p1,p2	-
Only member B exists	-	p1,p2,p3
Both member A and B exist	p1,p2	p1,p2,p3

Figure 11. Experimental results.

Member A is absent.

From the above results, we verified that the Authority Place Ag and the Personal Ags successfully represented presence of members, and Personal Ag could obtain access rights from the Resource Ags in accordance with social relation ships. Hence, it is confirmed that our scheme can delegate access rights for temporal activities utilizing relationships of persons who are present in digital space.

6: Conclusion

In this paper, we proposed a novel agent-based access rights delegation scheme which utilizes social relationships between a native member and a visitor. Our scheme can deal with emergent situation effectively because access rights delegation of our scheme is dynamically proceeded depending on presence of a guarantor. We conducted the experiments using a prototype system. Through the experimental results, we confirmed the effectiveness of our scheme. In future work, we will apply our scheme to several applications in digital space such as 3D collaborative environments, and investigate the effect of our scheme for social reality.

Acknowledgement

This work is partially supported by the Research and Development of Dynamic Network Technology program of NiCT.

References

[1] Gail-Joon Ahn, Longhua Zhang, Dongwan Shin, and B. Chu. Authorization management for role-based collaboration. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, volume 5, pages 4128 – 4134 vol.5, oct. 2003.

[2] E. Bertino, P.A. Bonatti, and E. Ferrari. Trbac: A temporal role-based access control model. *ACM Trans. Information and System Security*, 4(3):191–233, 2001.

[3] Jung Hwan Choi, Dong Hyun Kang, Hyunsu Jang, and Young Ik Eom. Adaptive access control scheme utilizing context awareness in pervasive computing environments. In *Performance, Computing and Communications Conference, 2008. IPCCC 2008. IEEE International*, pages 491 –498, dec. 2008.

[4] A. Dersingh, R. Liscano, and A. Jost. Utilizing semantic knowledge for access control in pervasive and ubiquitous systems. In *Networking and Communications, 2008. WIMOB '08. IEEE International Conference on Wireless and Mobile Computing,*, pages 435 –441, oct. 2008.

[5] Wang Fengying and Zhou Lili. Ucondfnnd - an effective delegation model. In *Web Information Systems and Mining, 2009. WISM 2009. International Conference on*, pages 557 –561, nov. 2009.

[6] Foundation for Intelligent Physical Agents. Fipa policies and domains specification. http://www.fipa.org/specs/fipa00089/, 2001.

[7] S. Fujita, H. Hara, K. Sugawara, T. Kinoshita, and N. Shiratori. Agent-based design model of adaptive distributed systems. *The International Journal of Artificial Intelligence, Neural Networks and Complex Problem-Solving Technologies*, 9(1):57–70, 1998.

[8] C. Grompanopoulos and I. Mavridis. Towards differentiated utilization of attribute mutability for access control in ubiquitous computing. In *Informatics (PCI), 2010 14th Panhellenic Conference on*, pages 118 –123, sept. 2010.

[9] Gail joon Ahn. Role-based authorization constraints specification. *ACM Transactions on Information and System Security*, 3:207–226, 2000.

[10] J.B.D. Joshi, E. Bertino, U. Latif, and A. Ghafoor. A generalized temporal role-based access control model. *Knowledge and Data Engineering, IEEE Transactions on*, 17(1):4 – 23, jan. 2005.

[11] Tetsuo Kinoshita, Susumo Konno, Gen Kitagata, Takahiro Uchiya, and Hideki Hara. Symbiotic system: Coexistence and mutual respect of human, society, environment, and information system, forward: Socialware. *IPSJ*, 47(8):817–824, 2006.

[12] Sven Lachmund, Thomas Walter, Laurent Gomez, Laurent Bussard, and Eddy Olk. Context-aware access control; making access control decisions based on context information. In *Mobile and Ubiquitous Systems: Networking Services, 2006 Third Annual International Conference on*, pages 1 –8, july 2006.

[13] Tae-Hum Lim and Sang-Uk Shin. Intelligent access control mechanism for ubiquitous applications. *Computer and Information Science, ACIS International Conference on*, 0:955–960, 2007.

[14] Fang Pu, Daoqin Sun, Qiying Cao, Haibin Cai, and Fan Yang. Pervasive computing context access control based on ucon abc model. In *Intelligent Information Hiding and Multimedia Signal Processing, 2006. IIH-MSP '06. International Conference on*, pages 689 –692, dec. 2006.

[15] Chun Ruan and S. Shahrestani. Role based access control for web-based teaching systems. In *Computational Intelligence and Software Engineering (CiSE), 2010 International Conference on*, pages 1 –4, dec. 2010.

[16] A. Samuel, A. Ghafoor, and E. Bertino. Context-aware adaptation of access-control policies. *Internet Computing, IEEE*, 12(1):51 –54, jan.-feb. 2008.

[17] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. Role-based access control model. *Computer*, 29(2):38–47, 1996.

[18] R. Tamassia, Danfeng Yao, and W.H. Winsborough. Independently verifiable decentralized role-based delegation. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 40(6):1206 –1219, nov. 2010.

[19] Takahiro Uchiya, Takahide Maemura, Kenji Sugawara, and Tetsuo Kinoshita. Interactive design environment for agent-based system. In *Transaction of the Institute of Electronics, Information and Communication Engineers D-I*, volume J88-D-I, pages 1344–1355.