User-Oriented Authorization in Collaborative Environments

Klaas Sikkel
Faculty of Faculty of Computer Science, University of Twente
PO Box 217, 7500 AE Enschede, the Netherlands
sikkel@cs.utwente.nl

Oliver Stiemerling
University of Bonn, Institute for Computer Science III
Römerstraße 164, 53117 Bonn, Germany
os@informatik.uni-bonn.de

ABSTRACT

Access rights for collaborative systems tend to be rather complex, leading to difficulties in the presentation and manipulation of access policies at the user interface level. We confront a theoretical access rights model with the results of a field study which investigates how users specify access policies. Our findings suggest that our theoretical model addresses most of the issues raised by the field study, when the required functionality can be presented in an appropriate user interface.

KEYWORDS
Authorization, Groupware, CSCW, User-orientation, Negative rights

1. INTRODUCTION

Access control in collaborative systems is a nontrivial issue. Greif and Sarin (1986) pointed out that access control models from the operating systems and database worlds are inadequate for collaborative systems. This challenge has been taken up by several authors, resulting in a variety of access models designed for specific groupware applications (e.g. Shen and Dewan, 1992; Coulouris and Dollimore, 1994; Edwards, 1996). Collaborative systems tend to be rather complex, hence these models are complex as well (Ellis et al., 1991). The literature on this point is rather theoretical and leaves unanswered the important question whether the users are able to cope with the (supposedly necessary) complexity of the proposed models.

An access rights model with particular emphasis on the simplicity was proposed by Sikkel (1997a). This simplicity refers to the underlying mathematical model, however, and there is no evidence that it is perceived as simple by the user. The model has been partially implemented in the latest version of the BSCW shared workspace server[1]) (Bentley et al., 1997), and first beta tests showed that some improvements in the user interface are still needed.

In a field study by Stiemerling (1996, 1997), access rights were approached from the user's perspective. Users in different contexts were interviewed in order to find out how they reason about access rights and phrase access policies. In this paper we investigate how well the BSCW model addresses these issues and discuss which changes are needed to improve its user-orientedness.

[1] See the BSCW home page http://bscw.gmd.de for more information. The software can also be downloaded from there.

In Section 2 the essential traits of the formal model are introduced and in Section 3 the main results from the field study are presented. Section 4 discusses how these match, and which problems remain. Conclusions follow in Section 5.

2. THE AUTHORIZATION MODEL

There is a difference between access rights, also called authorization, and access control, which should ensure that there rights are not violated. In this context we only consider authorization, not its enforcement by a concrete system.

Authorization commonly involves three parameters: a *subject* (also called principal) has a *right* to perform some operation on an *object*. A classical way to organize this is the Lampson Matrix (Lampson, 1974), enumerating subjects in one dimension and objects in the other. Each cell contains the rights for a given subject on a given object. A column of the matrix yields the capabilities of one subject (all her rights on all objects), a row is an *access control list* (*ACL*) of an object (describing all rights for all subjects). Sophistication can be added by various ways to structure the dimensions of the authorization space.

2.1. Groups

The BSCW model adds a single primitive notion, *groups* that can be composed of users and groups. Group structures can be – but need not be – composed according to organizational structure. Rights are given to groups. An ACL for an object is realized as a directed acyclic graph with the rights on the object as sources and the users as sinks. This is illustrated in Figure 1. At one side, users take on certain roles within the organization. At the other side, rights can be grouped into coherent clusters as well. All these are simply called 'groups'. The node labelled 'read', for example, can be seen as a group of access rights or as a group of users to which these rights are granted.

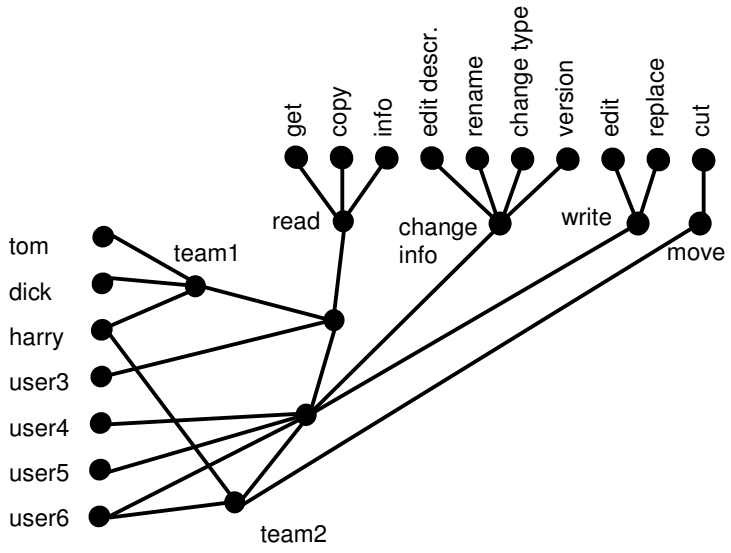

Figure 1: an ACL as a hierarchy of groups

The theoretical model makes a difference between groups that exist as independent objects (e.g. the group team1) and groups that are merely attributes of some object (e.g. "the readers of document *D*"). But this distinction is of no relevance here.

In many authorization models, the access space is structured by defining independent hierarchies for the three dimensions: subjects, objects, and rights. (Rabitti et al., 1991, Shen and Dewan, 1992). In the

BSCW model, subjects and rights are related to each other by a *single* hierarchy. Objects in a BSCW workspace are structured by means of hierarchical folders. Access rights can be given (or denied) to individual objects and folders, but also to a folder with its (transitive) contents. This is similar to the UNIX access rights model.

2.2. Ownership, Control, Delegation

In order to change the access rights to an object in the BSCW system, one needs a special right, the *control* right. The control right resides with the *owner group* of an object. Owners may also change the composition of the owner group – with one exception. There is a single responsible, who cannot be removed from the owner group unless he transfers this responsibility to a co-owner first. This set-up prevents "orphaned" objects that cannot be accessed and have no responsible.

Various forms of delegation (in which delegates may or may not further delegate rights) are supported by this general model. For example: one may create a new group *G* and give some rights to *G*. Making a second person member of G passes him these rights. Making a third person member of the *owner group* of *G* gives him the power to further delegate these rights to other users. See (Sikkel, 1997b) for a detailed formal treatment.

2.3. Negative rights

So far, only positive rights have been described. A user has the right to perform an operation if she is, directly or indirectly, member of the group to which this right has been granted. An extension to the model (envisaged, but not yet realized in Version 3.1, released December 1997), is the possibility to *exclude* a user (group) from a group. If group *H* is excluded from group *G* and user *U* is a member of *H*, then *U* is considered not be a member of *G* (even if she is listed as a member of *G* or one of its subgroups). Exclusion always overrules membership.

In the literature negative rights are often stated as a requirement for easy specification of access control. There are different models of negative rights based on different theoretical foundations. In the Andrew system (Satyanarayanan 1989), like in our model, negative rights override positive rights. Satyanarayanan's motivation is that negative rights are only used in exceptional cases (e.g. to immediately ban a user from the system). Shen and Dewan (1992) and Stiemerling and Cremers (1997) follow a different approach, in which there is no fundamental difference between positive and negative rights. We come back to this in Section 4.

2.4. Conditional rights

Another extension which is envisaged in the model but not yet implemented in the current BSCW system is the notion of *conditional rights*. Membership of a group (in principle any group, but typically the group to which some specific set of rights has been granted) can be made subject to a condition. The condition could be any piece of code, to be evaluated at run time, as suggested by Edwards (1996). But in order not to open the system to arbitrary access, a basic principle should be applied that evaluating a condition may only yield *denial* of a right that, had the circumstances been different, is granted in principle. That is: the user groups who have conditional rights should be specified within the system and evaluation of the condition decides whether this condition applies under the actual circumstances.

3. THE FIELD STUDY

The field study described in this section was conducted in the context of the POLITeam project (see Klöckner et al. 1995), in which the second author is currently involved. It encompasses, however, not only the POLITeam fields of application (public organizations – a ministry and a state representative body), but also a private and a semi-private (private set-up in public ownership) company in order to broaden the base of the results.

The POLITeam software is an extension of the LINKWORKS-groupware platform (DEC, 1995). This platform is intended for use in many different areas of application and is highly tailorable (see Stiemerling et al. 1997). However, The authorization model of LINKWORKS appeared to be not appropriate for the POLITeam user groups, and this was the prime reason to do the field study.

In order to capture a wide range of possible access policies, it was decided to undertake the field study in three different fields of application (cf. horizontal dimension in figure 2). Additionally – in order to capture intra-organizational differences – we selected interviewees from different hierarchical levels (cf. vertical dimension in figure 2) of these organizations. Our selection scheme is heuristic, designed to gather qualitative data (i.e. a wide range of access policies) rather than to produce quantitative or statistical results. For a detailed discussion of our design approach see Stiemerling et al. (1997).

Organization Level	**Private sector org.**	**Public sector org.**	**Private org. in public ownership**
Management	2 interviewees	1 interviewee	1 interviewee
Subordinates	2 interviewees	1 interviewee	2 inteviewees

Figure 2: Interviewee selection scheme

The interviews were semi-structured according to an interview guide containing 17 questions. These questions were designed to elicit access policies relating to the documents within the interviewee's scope of work (e.g. "What documents do you work with?" and "Who may read these documents?").

3.1. Example access policies

To give the reader an impression of the access policies encountered in the field, we briefly describe two examples, details of which we will use later on to exemplify our general observations.

The first access policy concerns the documents used in the admin department of a medium size (70 employees) software house. The admin department consists of the CEO of the company (Kurt[2]), his assistant (Melanie), three accountants (Gabriele, Alexandra, and Daniela), and a temporary student employee (at the time of the interviews: Sonja). The three accountants are mainly responsible for writing invoices, paying bills and maintaining documents containing aggregated financial information (pay roles, turnover, etc.). Other accountancy functions are outsourced to an independent tax consultant. The temporary employee usually goes through invoices of specific time-periods (e.g. prior years) adding up amounts to support the aggregation of financial data or looking for invoices so far unpaid. The invoiced are kept on a file-server running the Windows NT operating system.

During the interview with one of the accountants the following access policy was described: "*There is no difference between the three of us* [the accountants]. *If one of us is ill or on vacation the other two have to be able to use all important documents. The temporary employee has to be able to read all invoices of the last year* [for her current task]. *One specific person* [she was referring to her boss Kurt] *is allowed to read all documents, but since we have made bad prior experiences he is not allowed to change anything*." The last statement of the access policy refers to an incident, when the CEO changed an important document without telling anybody and, thus, causing misunderstandings and inconsistencies. Important documents in this context are, for example, the main journal or pay role related documents.

---

[2] Names are changed to protect the anonymity of our interviewees.

The second example is from a German federal ministry. At the time of the interviews politically rather controversial new law proposals were prepared for the parliament in this ministry. On prior occasions, supposedly neutral civil servants, who sympathized with the opposition party, had leaked a draft to the press, causing great embarrassment for the ministry. Thus, an access policy in the respective ministry was to deny or grant access to such controversial documents according to known political affiliations: "*Members of the opposition party are not allowed to read the draft proposal*."

In the remainder of Section 3 we present our analysis of the results of the field study. We begin with the formulation of access policies by the interviewees. Then we describe the factors, upon which the granting or denying access in the policies depends. Finally we describe how the interviewees dealt with unanticipated accesses during absences and illnesses.

3.2. Formulation of access policies by the interviewees

Analyzing the way access policies were expressed by the interviewees was quite difficult, due to differences of language and communication skills, but we have identified a few general principles. Of course we also observed exceptions to these principles. Here we give only brief description. For a more detailed and differentiated discussion of methods and results we refer to our other work (Stiemerling 1996 and Stiemerling et al., 1997).

Observation 1: Access policies are stated as sets of permissions and denials

Usually interviewees stated their access policies in form of basic statements of permission or denial. The following sentences give an exemplary overview over the form of the basic statements (translated from German):

1. Members of the admin department are allowed to read and change invoices.
2. Kurt is allowed to read the main journal.
3. Kurt is not allowed to change invoices.

While the first two statements are formulated as permissions, the last statement is an explicit denial. This reflects the observation stated in related work (cf. Shen and Dewan 1992) that the support of explicit negative rights in access control systems would facilitate the specification of access policies.

A complete access policy usually consisted of a set of such basic statements. Some statements were rather general, like statement 1 above, encompassing several users and a number of documents. Other statements, like statement 2, were more specific, determining the access rights of only one user on one object (cf. access matrix model).

Observation 2: Access policies are refined by stating exceptions to more general statements

Some users began describing access policies using very general statements (cf. statement 1 above). Then they refined the policy by stating exceptions to general statements (cf. statement 3 above). There was no bias towards either permissions or denials on the different levels. On the one hand, general denials ("Nobody may read documents on my desk") were refined by more specific permissions ("[But] Mr. Hillebrand may read documents on my desk"). On the other hand, general permissions ("All members of the admin department are allowed to change invoices") were refined by more specific denials ("[But] Kurt is not allowed to change invoices", with Kurt being a member of the admin department).

This way of describing access policies implies a natural way of interpreting a set of permissions and denials, based on the principle "most-specific-statement-holds" (cf. conflict resolution rules in Shen and Dewan 1992).

We also observed that very general statements were not explicitly included in the access policies but implicitly assumed. The prominent example was the statement "Everything that is not explicitly allowed is forbidden." Our interviewees unanimously choose this more conservative approach, instead of the equally possible alternative "Everything that is not explicitly forbidden is allowed."

3.3. Scope of permission and denial

The statements usually expressed permission and denial for a certain *scope*. The "size" of the scope determines the level of generality. The following factors were used to specify scope.

Users and documents

For the most part, the statements contained some description of the users and documents the statements applied to. The users and documents were either specifically named in the statement ("Sonja is not allowed to read the main journal") or described in more general terms ("All users from the admin department are allowed to read invoices").

Groupings of users were described on the basis of organizational units, roles and other more individual criteria (political affiliation, level of trust, etc.). Groupings of documents were usually described by content-based criteria ("invoices") and location ("documents on my [physical or virtual] desk").

Sometimes documents were specified by their status of completion: "My boss only may see documents which are completed".

Relationships between users and documents

In some statements the scope was determined by stating certain relationships between users and objects, e.g. owner-, current-user-, or past-user-relationships (cf. Greif and Sarin 1986). In LINKWORKS it is possible for users to sign documents electronically. Therefore, another useful relationship would be the relation *signed_by*. Access to a document could be granted only if the document is signed by a certain person, e.g. the supervisor.

Other factors determining scope

Some statements included the time dimension, either in recurrent form ("... on weekdays ...") or in absolute form ("... from May 1st 1996 to June 4th 1996 ..."). Other statements referred to groups of operations, e.g. "The owner of a document may do everything with a document", or "Melanie may read and change invoices".

3.4. Dealing with unexpected accesses

Classical access control systems cannot not deal with unanticipated urgent needs for access, when the owner of the data is away from the office. However, our interviewees were quite ingenious when it came to finding improvisations to get around the limits of the access control systems. In the public sector organization, the passwords for the virtual desktops in the groupware system are written on a paper and placed inside an envelope in the office safe. A trusted person (actually two: the department head and the system administrator) has the keys to the safe and in case of an unexpected need for access, the person wanting the data has to ask the trusted person to surrender the password from the safe.

In another field of application from the POLITeam project, the virtual desktops are totally private. The common workspaces, however, can all be accessed by a trusted person (the head of the typing pool), who can be asked to find and copy the data in the case of an unanticipated access.

4. DISCUSSION

We will now review the issues drawn from the field studies and indicate how the BSCW access control model (and, where relevant, other models) address these issues.

Access policies are stated as sets of permissions and denials; access policies are refined by stating exceptions to more general statements

In Section 2.3 we made a broad distinction between biased models (negative rights override positive rights) and unbiased models (both have equal, but opposite value). Although there is a bias in the generally shared assumption that "everything that is not explicitly allowed is forbidden", the field study seems to favor an unbiased approach.

A technical problem (which was not perceived as a problem by the users) is what to do with inconclusive or contradictory access policies. Consider, for example, the hypothetical statements "employees in Department *X* are allowed to read document *D*" and "members of the opposition party are not allowed to read document *D*". What is the rule, and what is the exception? Another example, from (Shen and Dewan, 1992): university staff has been granted access to some document, and student are denied access. What happens if a person is both staff and student?

It is a justified concern of the system designers that access rights specifications must be unambiguous. Hence, some disambiguation rules must exist. In Shen and Dewan's model, all other factors being equal, the ordering of the statement is relevant. Stiemerling and Cremers (1997) order the relevance of the predicates used to describe the access rights. Both solutions work, in sense that sense that ambiguities are ruled out, but there is no guarantee that the implications of the system's disambiguation rules coincide with the user's intentions. A rather more simple disambiguation policy "in case of doubt (i.e. contradictory specification) no access is granted" is biased, but it is a lot easier to explain such a policy to the user.

The proposed authorization model can be phrased in a user-oriented fashion as follows:

- If nothing is specified, access is denied.
- Access [to a certain operation on a certain object] is granted for named groups (users)
- This access is denied to named groups (users) on the exception list

In principle, the BSCW model supports exceptions to exceptions, but is it is rather hard to explain to the user how hierarchies of exceptions are handled.

Should the authentication model allow for specifications of arbitrarily complex access policies? Or it is better to restrict the expressive power of the model to access policies that users can understand? It depends on the purpose of the system, of course, but in order to keep a system user-oriented, there is something to say for the latter.

The use of roles

Roles are explicitly catered for in the model. Groups of any kind can be created, as long as a group is not, directly or indirectly, a subgroup of itself. This includes "odd groups" (i.e. groups typically not foreseen by the system designers) such as civil servants being member of a particular political party.

Grouping objects

One of the interesting conclusions of the field study is that users group objects according to *content-based* criteria ("members of the admin department have access to *invoices*"; "my boss should not be able to read *drafts* of my articles").

In the BSCW authorization model, similar to UNIX, access is related to documents and/or to folders. The only way the user can group access rights on objects is by grouping the objects into an appropriate folder hierarchy.

A possible way to get forward is to pursue the idea of conditional access: User U is granted access to all objects O in a given location – but only if the predicate `invoice(O)` evaluates to `true`.

Other factors determining scope of access rights statements

Various kinds of time constraints arose from the field study: "Only from May 1st to June 4th 1998," "Only on weekdays," "Only if no later version of this document exists." All of these can be handled with conditional access rights. Grouping of access rights is supported as well.

Dealing with unexpected need for access

The old conundrum of how to anticipate the unexpected. Situations in which a document must be retrieved and, for whatever reason, the only person able to do so it not around, are part of daily life. The question arises whether this should be solved *within* the system or by social protocol *around* the system. The UNIX solution, to have an omnipotent "superuser" with unlimited power, is too simple;

there is no reason why the system administrator dealing with user administration and file backup should be the same person who is able to read an important classified document, when suddenly needed.

In systems with no superuser, the solution to put passwords in envelopes and lay down a protocol who may open these envelopes is an intelligent workaround (with a minor flaw: there is no guarantee that the password is the right one).

In the BSCW model there are several answers. A technical solution is possible (but not currently supported) in which the system administrator could transfer all rights of a user *U* to a group which contains, say, *U* and *V*. A better alternative would be to let the users *delegate* this right to some trusted person.

Currently there is a possibility to handle this outside the system, when really needed. A user can obtain a new password without knowing the old password. One can request a special key to be sent to the user's email address. With this key the password can be changed. Hence, if one wants to "break in" to a user's BSCW environment, one has to persuade the system administrator of the user's system (not the BSCW sysadm) to retrieve the key from his mailbox. Getting this organized involves a considerable amount of negotiation outside the system – and rightly so – but provides a back door for emergencies.

5. CONCLUSIONS

The field study involved users in different application fields on different levels, hence it should be representative for a broad group of end users. We consider it important, because user-orientation has not received much attention in the literature on access control.

The BSCW authorization model was designed to be general, but based on a system from an application field (collaboration over the internet, particularly in distributed research projects) that was not represented in the field study. Most of the required functionality unveiled by the field study is indeed addressed by the proposed authorization model, which provides circumstantial evidence for the claimed generality.

Not adequately supported in the BSCW model is the possibility to relate access rights to the contents – rather than location – of documents (but this holds for most systems). This topic merits further research.

The work reported here still needs to be carried further. The next step is the design of a user-oriented user interface for stating access policies in the BSCW system.

REFERENCES

Bentley, R., Appelt, W., Busbach, U., Hinrichs, E., Kerr, D., Sikkel, K., Trevor, J. & Woetzel, G. (1997) Basic Support for Cooperative Work on the World Wide Web. *International Journal of Human-Computer Studies* **46**, 827-846.

Coulouris, G. & Dollimore, J. (1994) A security model for cooperative work. Technical Report 674, Dept. of Computer Science, Queen Mary and Westfield College, University of London, 1994.

DEC, Digital Equipment Corporation: LinkWorks Version 3.0 – User's Guide. Part number AA-Q3KYB-TE, 1995.

Edwards, W.K. (1996) Policies and Roles in Collaborative Applications. *ACM Conference on Computer Supported Cooperative Work* (*CSCW'96*), Cambridge, Mass., 11–20.

Ellis, C.A., Gibbs, S.J. & Rein, G.L. (1991) Groupware: Some Issues and Experiences. *Communications of the ACM* **34**, 1, 35–58.

Greif, I. & Sarin, S. (1986) Data Sharing in Group Work. *ACM Conference on Computer Supported Cooperative Work* (*CSCW'86*), Austin, Texas.

Kloeckner, K., Mambrey, P., Solhlenkamp, M., Prinz, W., Fuchs, L., Kolvenbach, S., Pankoke-Babatz, U. & Syri, A. (1995) POLITeam – Bridging the Gap between Bonn and Berlin for and with the users. *European Conference on Computer Supported Cooperative Work* (*ECSCW'95*), Stockholm, 175–183.

Lampson, B. (1974) Protection. *ACM Operating Systems Review* **8**, 18–24.

Rabitti, F., Bertino, E., Kim, W. & Woelk, D. (1991) A model of authorization for next-generation database systems. *ACM Transactions on Database Systems* **16**, pp. 79–86.

Satyanarayanan, M. (1989) Integrating Security in a Large Distributed System. *ACM Transactions on Computer Systems* **7**, 247–280.

Shen, H. & Dewan, P. (1992): Access Control for Collaborative Environments. *ACM Conference on Computer Supported Cooperative Work* (*CSCW'92*), Toronto, Canada, 51–58.

Sikkel, K. (1997a) A Group-Based Authorization Model for Cooperative Systems. *European Conference on Computer Supported Cooperative Work* (*ECSCW'97*), Lancaster, UK, 345–360.

Sikkel, K. (1997b) A Group-Based Authorization Model for Computer-Supported Cooperative Work. Arbeitspapiere der GMD 1055, GMD, Sankt Augustin, Germany.

Stiemerling, O. (1996) Anpaßbarkeit in Groupware – ein regelbasierter Ansatz. M.Sc. Thesis, Dept. of Computer Science, University of Bonn.

Stiemerling, O. & Cremers, A.B. (1997) A user-centered approach to access control in collaborative environments, *Second International Workshop on CSCW in Design*, November 1997, Bangkok, Thailand.

Stiemerling, O., Kahler, H. & Wulf, V. (1997) How to Make Software Softer – Designing Tailorable Applications. *2nd Conference on the Design of Interactive Systems*, Amsterdam, 365–376.

RESUME

Les droits d'accès mis en place dans les systèmes collaboratifs sont en général de nature assez complexe. La présentation et la manipulation de politiques d'accès posent notamment problème au niveau de l'interface utilisateur. Nous comparons un modèle théorique de droits d'accès avec les résultats d'une étude analysant sur le terrain comment les utilisateurs spécifient les politiques d'accès. Nous constatons finalement que le modèle theorique que nous proposons couvre la majorité des points soulevés par l'étude auprès des utilisateurs, pour peu que les fonctionnalités requises puissent être présentées au moyen d'une interface utilisateur adéquate.

To appear in the proceedings of the Third International Conference on the Design of Cooperative Systems (COOP'98), Cannes, France, May 1998.