Access Rights –
The Keys to Cooperative Work/Learning

Thorsten Hampel
University of Paderborn, Computer Science, Heinz Nixdorf Institute, Fürstenallee 11, 33102 Paderborn, Germany
hampel@uni-paderborn.de

Abstract. It is almost impossible to distinguish and keep track of all the cooperative systems and computer-supported learning environments currently available. One of the key characteristics of any cooperative system is the chosen access rights model. Most systems of this sort employ a model that is specifically tailored to one application area, e.g. a number of fixed roles. This article begins by examining the different basic characteristics of access rights in the area of cooperative work, and then goes on to present an open and flexible rights model. In addition to covering classical access rights, this model enables rights to be delegated to others and inherited contextually. The presented model is implemented in the cooperative sTeam open-source work environment. Finally we will present some evaluation results of using sTeam at various courses at our university.

1 Introduction

Promising cooperative knowledge organization and modern computer-supported cooperative learning are characterized by open and flexible forms of interpersonal cooperation and handling of material. One feature of such self-organized and open environments is that they allow user groups to be created flexibly and on a self-organized basis. Another is the need to make the handling of material (documents) adaptable to this new and flexible form of cooperative work [12]. Access rights play a crucial role here. They govern the assignment of objects to groups of users or learners, thus clearly defining individual users and user groups. Such access structures must, however, be flexible enough to accommodate different cooperative processes between users, e.g. the exchange of documents, without the need to explicitly make complicated adjustments to the respective user rights.¹

A key concept here is, for example, the context in which a document (object) is embedded.

The process of assigning user rights must therefore be seen as an essential criterion of a cooperative learning environment’s self-administration and self-organization.

¹ In practice, many systems avoid the problem of having to make manual adjustments to a large number of access rights when handling objects – e.g. moving a document – by choosing quite open rights structures: “all users have full access rights”. However, such an approach can only be considered a viable solution in limited use contexts.
Powerful mechanisms for assigning, transferring (delegating) and deriving access rights from the use context are the basic elements of a learning environment that can be largely organized and structured by the learners themselves – in other words, a user-centred learning environment.

By the early 1990s, Ellis et al. had recognized that all existing CSCW system concepts were overly complex and said nearly nothing about mapping the underlying rights models to the user interface [5]. This conclusion must be qualified when applied to the situation today, but it continues to be valid: existing access rights models of cooperative systems are either trivial or highly complex.

There are in addition a large number of generic models and concepts for administering user rights in non-cooperative applications. Unfortunately, such models cannot be applied to cases in which cooperative teaching and learning or work environments are supported. This might be one of the reasons why in recent years few cooperative learning environments have been equipped with powerful user rights models [24, p. 51]. The rights systems of cooperative environments can also be extremely complex. Requirements with respect to clarity (comprehensibility) and suitability of the user interface thus also acquire importance.\(^2\)

Elaborated models of access rights are found in many systems, such as [2], [3], [4], [9], [18] or [23].

The user rights model presented here is flexible and adaptable enough to cover the whole range of cooperative knowledge organization processes. And it is designed to minimize the complexity and effort involved in setting user rights. Key concepts here are the specific right to delegate user rights, the inheritance of access rights and their derivation from the context/environment of an object.

2 Access Rights – The Keys to Cooperative Knowledge Organization

A general distinction should be made between authentication, i.e. logging on to a system and thus securing a specific status depending on the respective user (“Who are you?”, “Who said that?”), and authorization, i.e. checking access rights between documents, objects and a person (“What are you allowed to do?”, “Who is allowed to access this document?”).\(^3\) Conventionally, operating systems have made access rights to a specific file dependent on membership of the relevant user group. In the UNIX operating system, for example, the user’s or user group’s access rights to a file are defined in terms of only three attributes: read, write, execute. These rights are evaluated depending on a user’s membership of specific groups.

The classical model for assigning access rights to users and user groups (domains) and objects is the Access Control List (ACL) as defined by [19], [20]. Lampson’s notation consists of a set of objects \(O\), a set of domains \(S\) and an access matrix \(A\).

\(^2\) End users must be able both to easily recognize existing or refused access rights and to modify such rights (see [5]).

\(^3\) Henceforth, the terms “access rights” and “access control” are used synonymously.