Research Issues in Authorization Models for Hypertext Systems

Elisa Bertino Pierangela Samarati

Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano
Via Comelico 39/41
20135 Milano, Italy
e-mail: {ebertino,samarati}@dsi.unimi.it

Abstract

The characteristics proper of hypertext systems, such as absence of schema, connections among the different "chunks" of information, possibility of navigating in the hypertext, make conventional authorization models not adequate for their protections. These characteristics, on one side raise new protection requirements, thus making the problem of protection much harder; on the other side, they provide a flexibility in the specification of authorizations greater than in more structured data models. We are currently working on an authorization model for the protection of information in distributed hypertext systems. In this paper we illustrate the new requirements that arise and discuss some of the issues we are currently investigating.

1 Introduction

Hypertext/hypermedia systems represent today an effective approach to organize and present information to users. Their usage is rapidly increasing. Systems like World Wide Web (WWW) [4], use the same approach to allows users to "navigate" in a Wide World information space. The key point of such an approach is that every *piece* of information is connected, via *links* to related pieces of information. These links are graphically visualized together with the information, so that users only need to click on the link to get to the related information. Thus, the combination of navigation access facilities with graphical representation makes it very easy to reach many information sources.

Even though hypertext/hypermedia systems have been receiving a lot of attention from researchers, developers, and users [8, 10, 12, 13, 15, 22, 24, 25, 26, 27], the problem of access controls, and security more in general, in such systems has not been widely investigated. It may appear a contradiction to include an access control mechanism in systems whose goal is to increase information availability and ease of access. However, we believe that the increasing widespread usage of such systems will require tools to guarantee the correct access to information.

In this paper, we focus on issues related to discretionary access control for distributed hypertext/hypermedia systems. Mandatory access control is not addressed in the current paper, as well as other important issues such history-based access controls. We believe that discretionary access control is the first mechanism which can be reasonably incorporated into a hypertext/hypermedia system. Indeed, the information currently available in such systems are not in general so sensitive to require stronger protection mechanisms. However, investigation of stronger protection mechanisms will soon follow "on the road".

The definition of an adequate discretionary access control mechanism is quite difficult due to the peculiarity of the "data model"[1] of hypertext/hypermedia systems. It is not straightforward to adapt models, developed for Relational DBMS or OODBMS [7, 17] for a number of reasons. First, the conceptual elements of an hypertext/hypermedia data model are quite large in number; moreover, their definition and semantics is not uniform and varies from system to system. In defining, a suitable authorization model, the semantics of those elements must be clearly defined and the possible actions that can be executed on them must be identified. Second, access in those systems is mainly based on navigation through information pieces. This fact implies two requirements. First, if a user must be given access to an information item, it may not be enough to give him the right to access the item. Indeed, the user may not be able to reach it. Thus, accesses to paths reaching a given information item must be provided. Since those paths may traverse other information items, care must be provided in defining the proper paths. Moreover, mechanisms and tools must be provided to assist the security administrator. Second the same information piece can be reached starting from different entry points. Thus, if access to an information item must be controlled, it is necessary to be sure that no way of reaching it exists. Moreover, access to an information item may be given to a user depending on the specific entry point.

In this paper we present a discussion of some research issues concerning the protection of distributed hypertext/hypermedia systems and illustrate an authorization model on which we are currently working.

The remainder of this paper is organized as follows.

[1] We use the term "data model" in a broad sense.

0-8186-7318-4/95 $4.00

Section 2 recalls the main characteristics of hypertext systems. Section 3 discusses the research issues that arise in the protection of information in hypertext systems. Section 4 illustrates the start of the art and ongoing work in the protection of distributed hypertext systems. Section 5 illustrates the research we are carrying on in this area. Finally, Section 6 presents the conclusions.

2 The hypertext data model

A hypertext can be defined as a collection of *nodes* containing information and *links* connecting them. There is no unique model for hypertext systems. Each system has its own characteristics. However, some basic concepts, characterizing hypertexts, are common to all the approaches. These can be summarized as follows.

Nodes A node is defined as a collection of data related to a specific topic. Node can be frames, with a fixed dimension, each of which contains some information [22] or windows with variable dimension which can be read by scrolling their contents. Nodes can be typed [9]. The type of a node depend on the kind of information contained in the node.

Links Links are references between nodes. A link between two nodes represents the fact that the information embedded in the source node is related to the destination node. Links can be of different types corresponding to different kinds of connections that can be specified. Links can connect, for example, different parts of a texts, note or comments to a text, or elements of a table or a graph. Connections can be hierarchical or of reference. Hierarchical connections organize information in a structured form, presenting the information at different levels of details. Reference connections represent semantic correlation between information. Link sources and destinations can be whole nodes or specific regions inside a node (anchors).

Scripts Scripts are procedures that can be executed upon the verification of certain conditions or occurrence of events. By associating scripts to anchors, conditional links can be defined whose destination is determined at run time depending on the evaluation of certain conditions expressed in the script.

Buttons A button is a visual signal indicating to the users that a link/script exists. By clicking the button a user can activate the corresponding link/script. Buttons can be of different types, e.g., expansion (by clicking the button text hidden from view is shown), reference (by clicking the button the user jump to a new node or position in a node), or command (upon clicking the button the script associated with it is executed).

Navigation tools Information stored in nodes can be accessed by navigating in the hypertext following the links. Different kinds of navigation tools are available:

Browsers A browser is a program that can display a diagram of a network of nodes. Browsers allow to visualize a subset of nodes of the hypertext and the links among them.

Query systems In hypertext allowing queries (e.g.,[1]), users can specify queries by requiring the system to return, for example, all the nodes that contain a given string.

Filters Users can select the information to be visualized by specifying conditions on attributes associated with nodes.

Trails A trail is a record of nodes that a user has accessed when navigating along the hypertext. The trail indicates the history of all the nodes examined by the user in arriving at the current node.

Tours A tour is a predefined trail that a user can navigate. When following a tour, users can move only along the links of the tour. Tours are intended to be traversed linearly.

Bookmarks Bookmarks are flags associated with nodes that allow the user to distinguish a specific node which they may wish to revisit.

Webs A web is a group of nodes linked together. Nodes can also be complex, i.e., be themselves a subnetwork of a larger network. Webs can be active or inactive. The buttons that point to the nodes in the web are visible only when the web is active.

3 Research issues

In this section we discuss some research issues that arise in the protection of distributed hypertext systems.

Authorization objects Authorization objects are nodes, links, and scripts. Authorization objects can also be portion of nodes. If a user has the authorization only for a subpart of a node, when accessing it, the part for which he does not have the authorization should be invisible to him (i.e., the user should not even know about its existence). From this point of view, the lack of structure of hypertext helps in protecting the existence of information. For example, in relational database systems, where each tuple in a relation has the same number of attributes, it is only possible to hide the value of the attribute for each particular tuple in the relation but not the attribute itself. By contrast, in hypertext systems, where no structure for nodes is provided, parts of the nodes can be hidden to the user without letting the user know about their

existence. On the other hand, the lack of structure may make more difficult the specification of authorizations referring to specific non structured parts of a node. (e.g., part of a text or of an image).

Context dependent authorizations The same user may be authorized to access a given node differently depending on the path followed in arriving at the node. To support this, beside the user/group, also the paths should be taken into consideration in the definition of authorizations.

Navigation tools Navigation tools can help in defining authorizations. For instance, specific tours in the hypertext can be defined for which groups of users can be allowed. The authorization on a tour may allow the users to follow the tour in the specified way. Alternatively, additional authorizations may be required for the users in order to complete the tour.

Authorizations on buttons A button is a visual representation of a link or a script in a node. Different kinds of authorizations can be considered on buttons. For example, it should be possible to hide completely the button to the user, show it to the user and allow him for the activation, or showing the button to the user but without having the user recognize the button as such. In this latter case, the user sees the information in the button but he does not recognize it as an active object.

Authorizations on scripts Scripts are procedures associated with nodes or parts of nodes. During execution, a script may require the execution of other operations (e.g. traversing a link or producing a sound). This operations can be controlled with respect to authorizations specified for either the user who activated the script or the script itself (or for the node with which it is associated). This second option require the possibility of scripts to appear as subjects of authorizations.

Authorization subjects Hypertexts generally represent large collections of information that several users can traverse, and several users often share the same authorizations on a hypertext. It seems therefore appropriate to specify authorizations for user groups instead than for single users.

Authorizations on types If nodes are typed, authorizations can be specified on single nodes as well as on groups of nodes of a same type. For example, in a hospital hypertext, an authorization can state that subject group `doctors` can access all nodes of types `patients`.

Authorizations on kinds of data In a hypermedia system, nodes can contain different kinds of data: graphics, audio, video, images, and so on. Authorizations can be specified depending also on the kind of data to be accessed. For example, access to image nodes, which is expensive form a system point of view, can be allowed only to specific groups of users.

Authorization administration In a hypertext system, users may be authorized to insert information (e.g., to create new nodes or "notes"). Nodes created by different users can then be connected by links. In some situation, users are considered owners of the information they insert (e.g., the WWW page with the description of a user is of property of the user who therefore can change it at any time). For the complexity of the system and the interrelationships that can exist between the different nodes and different "pieces" of information, the ownership concept of administration often used in traditional authorization models may not be appropriate. Alternative, or complementary, administration policies chould therefore be devised.

Authentication Authentication is a prerequisite for correct access control. When a user requires access, access control decides whether to grant or deny it with respect to the authorizations of the user. It is therefore important that the identity of the user be authenticated. When the system is distributed, the user is generally required to authenticate himself at every site at which he requests to operate. This approach may be very impractical since users will tipically browse the hypertext following links pointing to nodes stored at different sites (of which they might even ignore the existence). Requiring the users to identify themselves at each accessed site implies requiring the users to insert a login and a password at every access. Moreover, it would require knowledge of the user about where the node to be accessed has been defined (identity and password needed will depend on the specific site). Different approaches to authentication should therefore be devised. For instance, the identity of the user at the site where it is connected can be used in the access control. This approach requires each site to trust the identity of the user submitting a request (possibly by requiring some form of certification [11, 23]). For instance, the Kerberos cross-realm authentication mechanism [23] can be used to this purpose.

4 State of the art and current proposals

Although several hypertext models/systems have been proposed and the research in the hypertext field has been very active recently, authorizations and access control issues have not been adequately investigated. The reason is that several research efforts have been devoted to the problem of data representation and retrieval, whereas very little attention has been devoted to the problem of its protection. The motiva-

tion of this lack is mainly due to the fact that hypertext sytems are generally intended for sharing of information among users. It is true, however, that as the quantity of the information to be shared grows, the need to selectively share it and make it available only to specific users or for specific uses immediately arise. Few hypertext models/systems provide today protections capabilities.

In KMS [2] a user creating a frame is considered its owner and can restrict access to the frame from other users. The owner can specify that others cannot access the frame, can access it but not make modification, or can only add annotation items without modifying existing items. In HAM [6] access authorizations can be specified for users and user groups to access the objects of the hypertext (graphs, contexts, nodes, links, and attributes). Four privileges are considered: access (to view the data associated with the object), annotate (to attach links to a node), update (to perform non destructive updates on an object), and destroy (to delete an object). Intermedia [22] allows users to specify, for each document, the privileges that can be exercised on it. Privileges can be specified for the owner of the document, the owner's group, all other users, and the intermedia administrator. In Trellis [26] the hypertext is represented by a Petri Net whose nodes are the hypertext documents and whose transitions correspond to link traversals. Access control is enforced by using marked nets. User classes are identified and each class is assigned an initial marking. The marking associated with a class of users determines the transitions in the net that can be fired and hence links in the hypertext that users in that class can traverse.

The hypertext paradigm has recently become very popular thank you to its application in the World-Wide Web environment. Information in the WWW is organized as a set of linked hypertext documents. Documents are defined through the use of the HyperText Markup Language. Document specifications are therefore basically files.

Although security issues in the WWW environment have been investigated, most researches have concentrated on the problems of authentication and data encription and very little attention has been devoted to the problem of authorization and access control. Authorizations mechanisms available today in the WWW environment are very primitive. Moreover, they enforce protection essentially by seeing each document as a file, therefore not exploiting the characteristics of the hypertext paradigm.

Mosaic 2.0 and NCSA httpd [14] allow to restrict access at the directory level. Authorizations can restrict access to the information contained in a directory to allowed hosts or authenticated users. Authorizations are specified by associating with each directory a list of hosts to which access must be authorized and a list of hosts to which access must be denied. Each host can be an actual host name, a domain name, an IP address (or a part of it) or the keyword `all`. Whether denials or permissions must be evaluated first (i.e., which is overridden by the other) is also specified. Authorizations can also restrict access from authorized hosts to specific users or groups. In this case users connected from authorized hosts will be required to identify and authenticate themselves and accesses are allowed only if the restrictions on the user identity or group membership are satisfied. Users, passwords, and groups used in authentication and access control do not correspond to accounts on Unix machines but are explicitly defined for the Web. Specification of users, passwords, and groups is based on the concept of authorization realm. Each directory can be specified as belonging to an authorization realm. For each authorization realm, users identifiers, passwords associated with users identifiers, user groups are specified. The realm is the name returned to a user required to identify himself so that the user knows the username and password to use.

This approach suffers from several limitations. Authorizations can be specified only for whole directories and not for single nodes and only for a single privilege (*get*) which allows to retrieve documents and execute scripts in the directory. Single nodes or links are not considered. If two nodes belong to the same directory they are subject to the same authorizations. Moreover, this approach does not take into consideration the different nature of nodes and links. A user allowed to access a node can see and activate all links leaving from the node. Although the user will be returned the node, destination of the link, only if authorized for that, this approach does not allow to protect the relationships between nodes. A user authorized to access two nodes can also traverse all links between them. Furthermore, enforcing identity-based access control requires users to identify and authenticate themselves at the access time. In the worst case, where all nodes apply identity-based restriction, this approach may imply the need for the users to enter a username and a password at every access.

In the CERN httpd [21] server access authorizations can be specified on directories and files. Two privileges are considered: *get*, to retrieve objects, and *post*, to create objects. Authorizations can be specified for users as well as for groups, where a group is a set of users or groups. Users are characterized by a user name and either an IP address or an IP address template. Authorizations for a set of files can be specified either at the level of the directory or by using the wild-card character in the specification of the file to which authorizations are referred. Also in this approach, user names do not correspond to Unix accounts. Each user, for whom authorizations can be specified or that can be member of groups, has a password. Users wishing to access information will need to identify and authenticate themselves to the system so that the proper authorizations can be evaluated. Moreover, access authorizations can be referred only to nodes, a user allowed to access a node can also activate all the links leaving from the node. This approach thus suffers from the same limitation as the previous one.

Kahan [16] proposes an approach to protect information in the WWW environment where the authorization to access an hypertext document implies the authorization to traverse all the links leaving from the document and to access the documents destination of

these links. Authorizations can also be specified on presentation trees (sets of objects connected by links rooted at a given document). The authorization to access a presentation tree implies the authorization to access all documents in the tree. The goal of the approach is, given the capability of a subject to access a document, to automatically generate capabilities for the subject to access all the documents reachable from it. Authorizations and policies for their propagation are specified by the single administrator at the time of the installation phase. Creation of new documents by users requires contacting the security administrator. The model is therefore appropriate in situations where documents have a long lifetime and can be administered by a central entity.

Several projects for supporting authorization-based access control in the WWW environment are currently being carried out [3, 18, 19, 20].

5 An authorization model for a distributed hypertext system

We are currently working on an authorization model for a distributed hypertext system [5]. We consider two different levels at which the system can be viewed: the *hyper* level is the logical view of the hypertext documents and the relationships among them, the *base* levels is the level at which information defining the documents is stored. At the base level information is stored in *nodes* and *basic objects*. A node is a frame describing a document. For each document in the hypertext, a corresponding node exists and viceversa. A basic object is a container of information that can be included in a document. To allow for the specification of authorizations referred to specific parts of nodes, each node can be partitioned into slots. A slot is a portion of a node identified by some label. Links can be defined between nodes and between nodes and objects. A link between two nodes, called *navigation* link, models a correspondence between the information in the nodes and, therefore, between the corresponding documents. A link between a node and an object, called *inclusion* link, indicates that the object is to be included in the node when producing the corresponding document. *Anchors* are used to indicate the start and ending point of links inside nodes. An achor can also include some content *handle* which is the portion of the document to be highlighed when the anchor represent the starting point of a navigation link.

We consider a distributed system where information can be stored at different sites. Although nodes and objects can be distributed, their physical distribution is invisible at the *hyper* level where users see a collection of documents regardless of whether the information they contain is physically stored. Both navigation and inclusion links can cross site boundaries, i.e, start and ending point of a single link can be at different sites. Figure 1 illustrates an example of two hypertext documents together with their specification at the base level. In the figure, notations <slot> and </slot> indicate the beginning and the end of a slot respectively. Analogously, notations <anchor> and </anchor> indicate the beginning and the end of an anchor. Links are described as triples of teh form $\langle$*link-id*, *source*, *destination*$\rangle$, where *link-id* is the identifier of the link, *source* is the anchor representing the starting point of the link, and *destination* is the destination of the link (it can be an anchor, a node, a slot, or an object).

In our model users creating nodes and objects are their owners and determine how to share them with other users by specifying authorizations. We consider three different types of authorizations: *browsing*, *authoring*, and *usage*. *Browsing* authorizations allow users to see the content of nodes (their corresponding documents) and traverse navigation links, *authoring* authorizations allow users to create and modify nodes and links, and *usage* information allow users to include objects in their nodes (corresponding documents).

Subjects of the authorizations are users holding accounts at some sites. Groups can also be defined, locally at each site, as sets of users as well as groups. Each request to access an hypertext document (either directly or through the activation of a link) is translated into requests on the objects composing the document (i.e., the corresponding node, possible objects to be inserted in it, and navigation leaving from the node). Access controls can therefore involve several sites: the one from which the user issues the request, the one where the node is defined, and those where possible basic objects to be included in the node are stored. At the sites of the nodes and the basic objects, access control must be performed to ensure that only authorized information is released. In particular, at the site where the node is stored, the request must be checked against the authorizations to view the node and to navigate links leaving from it, to ensure that only the slots and links for which an authorization exists are returned. At the each site where an object to be included in the node is stored the authorizations of the owner of the node to use the object must be evaluated and the object returned only if its use is authorized.

Since the number of objects in the system can be very large, the specification of authorizations at the node level can be very impractical. We are currently extending the model towards a decentralized administration policy. This policy is based on the concept of *authorization domains*. An authorization domain is a set of nodes and links among them grouped together for administrative purposes. Users can allow administrators to include the users' nodes in the domains they administrate. Administrators can allow users to post the users' nodes in the domains they administrate. Authorizations on a domain, which can only be granted, and revoked, by the administrator of the domain, apply to all the nodes of the domains. Distributed domains, that is domains that include nodes stored at different sites, raise some interesting issues with respect to authorization management and communication between sites. We are currently investigating these issues.

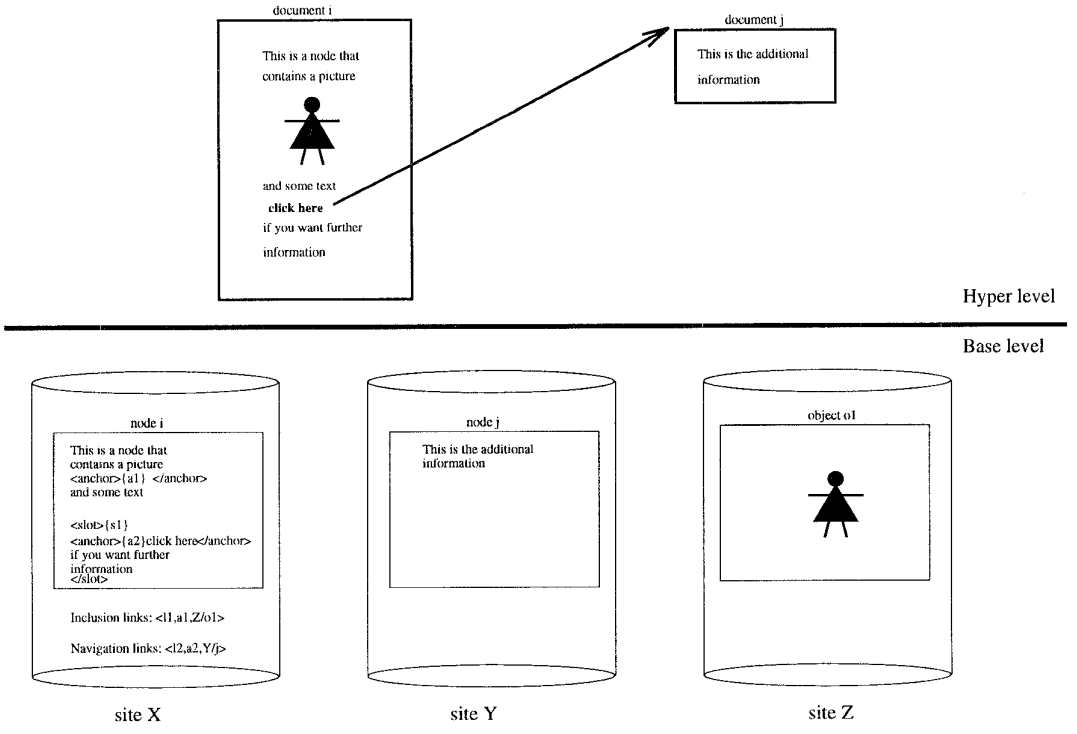

Figure 1: Base and Hyper levels

6 Conclusions

Conventional authorization models are not adequate for the protection of information in hypertext/hypermedia systems. In this paper we have discussed some research issues that arise in the application of discretionary control policies in distributed hypertext systems and illustrated an authorization model on which we are currently working.

References

[1] F. Aftrati and C.D. Koutras. A hypertext model supporting query mechanism. In A. Ritz, N. Streitz, and J. Andre', editors, *Hypertext: Concepts, Systems, and Applications*. Cambridge Series on Electronic Publishing, 1991.

[2] R.M. Akscyn, D.L. McCracken, and E.A. Yoder. KMS: A distributed hypermedia system for managing knowledge in organizations. *Communicatiosn of the ACM*, 31(7):820–835, July 1988.

[3] K. Andrews, F. Kappe, and H. Maurer. Serving information to the Web with Hyper-G. In *3rd World-Wide Web Conference*, pages 919–926, April 1995. http://www.igd.uiuc/www/www95/papers/105/hgw3.html.

[4] T. Berners-Lee, R. Cailliau, A. Luotonen, H.F. Nielsen, and A. Secret. The World Wide Web. *Communications of the ACM*, 37(8):77–82, August 1994.

[5] E. Bertino, S. Jajodia, and P. Samarati. An authorization model for a distributed hypertext system. in preparation.

[6] B. Campbell and J.M. Goodman. HAM: A general purpose hypertext abstract machine. *Communications of the ACM*, 31(7):856–861, July 1988.

[7] S. Castano, M.G. Fugini, G. Martella, and P. Samarati. *Database Security*. Addison-Wesley, 1994.

[8] J. Conklin. Hypertext: an introduction and survey. *IEEE Computer*, 20(9), September 1987.

[9] J. Conklin and M.L. Begeman. GIbis: A hypertext tool for exploratory policy discussion. *ACM-TOIS*, 6(4), 1988.

[10] N. Delisle and M. Schwartz. Neptune: a hypertext system for cad applications. In *ACM SIGMOD Int. Conf. on Management of Data*, Washington, DC, May 1986.

[11] W. Ford. *Computer and Communications Security*. Prentice Hall, 1994.

[12] F. Garzotto, P. Paolini, and D. Schwabe. HDM, a model-based approach to hypertext application design. *ACM Trans. on Office Information Systems*, 11(1), 1993.

[13] K. Gronbaek and R.H. Trigg, editors. *Communications of the ACM - Special Issue on Hypermedia*, February 1994.

[14] NCSA httpd Development Team. Ncsa httpd. http://hoohoo.ncsa.uiuc.edu/docs/Overview.html, July 1995.

[15] C.J. Kacmar and J.J. Leggett. PROXHY: A process-oriented extensible hypertext architecture. *ACM Trans. on Office Information Systems*, 9(4):399–419, October 1991.

[16] J. Kahan. A distributed authorization model for WWW. In *Proc. of INET'95 Conference*, Honolulu, Hawaii, http://www.isoc.org/HMP/PAPER/107, 1995.

[17] C.E. Landwehr. Formal models for computer security. *ACM Computing Surveys*, 13(1), 1989.

[18] M.G. Lavenant and J.A. Kruper. The Phoenix project: Distributed hypermedia authoring. In *Proc. of the 1st World-Wide Web Conference*, http://www.cern.ch/PapersWWW94/j-kruper.ps, 1994.

[19] S. Lewontin. The dce web toolkit: Enhancing WWW protocols with lower-layer services. In *3rd World-Wide Web Conference*, pages 765–771, April 1995. http://www.igd.uiuc/www/www95/papers/67/DCEWebKit.html.

[20] S. Lewontin and M.E. Zurko. The dce web project: Providing authorization and other distributed services to the World-Wide Web. In *2nd World-Wide Web Conference*, 1994. http://www.ncsa.uiuc.ued/SDG/IT94/Proceedings/Security/.

[21] A. Luotonen. Protected CERN server setup. http://www.w3.org/hypertext/WWW/Daemon/User/AccessAuth.html, January 1995.

[22] N. Meyrowitz. Intermedia: The architecture and construction of an object-oriented hypermedia system and application framework. In *OOPSLA'86*, September 1986.

[23] B.C. Neuman and T. Ts'o. Kerberos: an authentication service for computer networks. *IEEE Communications*, 32(9):33–38, September 1994.

[24] J.L. Schnase, J.J. Leggett, D.L. Hicks, and R.L. Szabo. Semantic data modeling of hypermedia associations. *ACM Trans. on Office Information Systems*, 11(1):27–50, January 1993.

[25] J.B. Smith and S.F. Weiss, editors. *Communications of the ACM - Special Issue on Hypertext*, July 1988.

[26] P.D. Stotts and R. Furuta. Petri-net based hypertext. *ACM Transactions on Information Systems*, 7(1):3–29, January 1989.

[27] F.W. Tompa. A data model for flexible hypertext database systems. *ACM Trans. on Office Information Systems*, 7(1), 1989.