Local Control over Filtered WWW Access

Brenda S. Baker
Eric Grosse

Abstract:

This paper describes a software system called Signet that provides local control over restricting access to resources on the World Wide Web (WWW). The strategy is to make it easy for a parent or teacher to rate WWW resources and place the tables of ratings into a proxy server designed to restrict Internet access according to the access permissions of the users. Ratings can be made dynamically as the rater browses, so that the rater can insert new ratings into the ratings database at any time. Ratings can also be shared by different raters or organizations so that rating of WWW materials can proceed in a distributed fashion.

Keywords:

World Wide Web, proxy, firewall, ratings, censorship, Exon amendment

Introduction

Senator Exon proposed an amendment to the United States Telecommunications Competition and Deregulation Act of 1995 that would make it unlawful to transmit indecent material over the Internet. This amendment was approved by the Senate with an overwhelming majority in June, 1995. Subsequently, the House of Representatives approved a bill calling for an evaluation of technical means of restricting distribution of unwanted material. As of October, 1995, these bills have been referred to a conference committee. Other countries are also considering the issue [1,8].

Many computer scientists have been opposed to the amendment on the grounds of freedom of speech, the projected adverse effects on the Internet if all resources and e-mail must be censored, and the impossibility of restricting access to foreign sites containing prohibited material.

However, the political issue would be defused if Internet access to inappropriate materials could be restricted for children through technical means. This paper describes a flexible, dynamic method that could serve the needs of schools and parents without depriving adults of free speech rights and without imposing a uniform national standard.

The ability to limit Internet access could also be useful for reasons other than to protect children from pornography. A teacher may want to limit students to viewing resources in the day's lesson plan; otherwise, as described by a middle school computer laboratory teacher [16], a teacher facing the backs of a room full of monitors cannot tell whether students have left the lesson content to browse among more entertaining topics. Some managers are concerned that employees are squandering work time on "surfing the Net" [6]; they might like to restrict employees to access only work-related resources. In these instances, it may be desirable to change access permissions dynamically, e.g. for the current class hour or for lunch break browsing.

Some people express the opinion that it is unlikely that a person will come across inappropriate material accidentally. However, the authors know of incidents where adults or children have unexpectedly encountered "adult" images or subject matter without being able to predict from the preceding page what the link would lead to. In addition, what is inappropriate depends on the viewer; for example, two mature women have asked the authors for protection from encountering "adult" material.

Internet access can be direct or through a proxy server and a firewall. In the latter case, a client sends requests for remote resources to a WWW proxy server which forwards them through a firewall to the remote server and forwards the responses from the remote server back to the client. The proxy server is a focal point at which access permissions can be checked for remote resources. If the firewall is configured to forward requests only from the proxy servers, client machines cannot bypass the permission checking, no matter what software is surreptitiously loaded onto them. Ordinary routers can act as such a firewall. The situation is illustrated in Figure 1.

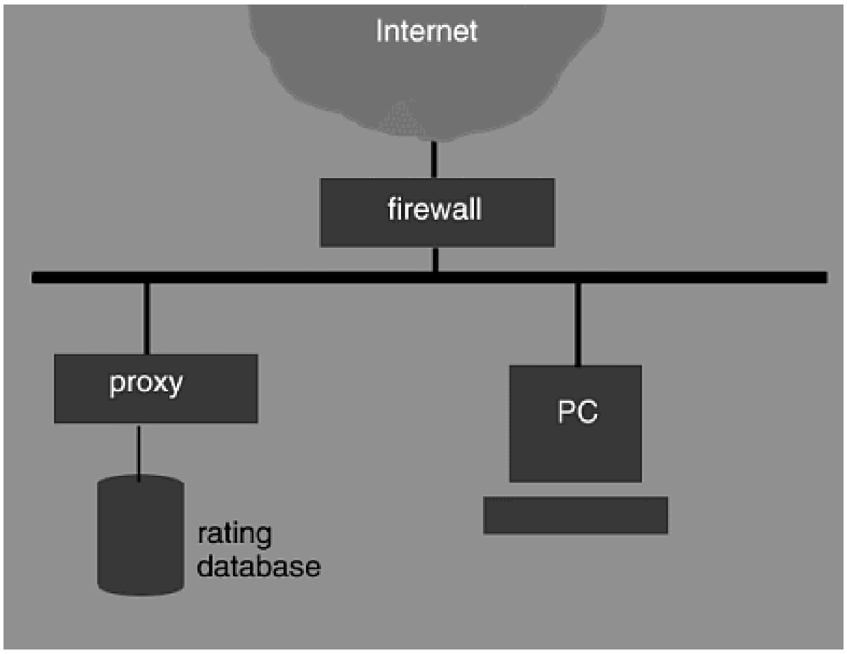

Figure 1. The Signet model

This paper describes a system called Signet that operates in this manner to provide restricted access to the WWW under local dynamic control. Signet consists of a modified HTTP proxy server and a ratings server. The ratings server maintains a ratings database. The Signet proxy server has two modes, rating mode (which is also an unrestricted browsing mode) and restricted browsing mode. By communicating with the rating daemon through HTTP, authorized people rate pages, directories, or sites while browsing and place access permissions dynamically into a database of permissions used by the proxy server. In restricted browsing mode, every attempt to access a remote resource causes the proxy server to check access permissions. Consequently, a student cannot follow a link from an approved page to an unapproved page.

The focus of Signet is on allowing students access to approved sites while denying them access to unapproved or unrated sites. This approach can be viewed as a Seal of Approval (SOAP) approach. The name Signet was chosen because a signet is a seal used to mark approval on a document.

Signet could be used immediately by teachers planning class lessons using WWW resources or by parents who find particular resources they would like their children to access. As more ratings are created, sharing of ratings created by trusted individuals or organizations will make wider browsing feasible under restricted mode.

There are many thousands of pages on the WWW, and more are created daily. Consequently, no organization will be able to stay completely up to date in rating everything on the WWW. Using our system, parents and teachers can supplement ratings by organizations with their own.

An Internet service provider running Signet would provide an authentication mechanism for accessing rating mode and could make available ratings by individuals or organizations. However, decisions about inclusions of ratings and access permissions would be up to the authorized parent or teacher and would not be the responsibility of the Internet service provider.

Approaches Suggested Previously

Some schools ask students to abide by policy statements by which they agree to restrict their exploration of the WWW, such as by agreeing not to download obscene material. An example of such a policy is the California Department of Education Electronic Information Resources Acceptable Use Policy District Guidelines [2]. However, students trying to comply with such a policy are not protected from accidental downloading of resources that are not readily identifiable as inappropriate prior to downloading.

Quality Computers [12] offers to schools a service called LINQ that selects resources appropriate for school children and downloads them to schools. In addition, LINQ offers direct Internet access using a browser provided with a menu of links to resources whose links are believed not to lead indirectly to inappropriate material; the browser does not permit requesting an arbitrary URL. However, this does not permit inclusions of certain resources that are themselves suitable for kids, such as a very popular and well-designed kids' page from which six clicks led to very adult images as of March 30, 1995.

One approach has been software that runs on the client machine and blocks access to a list of sites considered inappropriate for children. SurfWatch [19], Microsystems Software's CyberPatrol [9], and Solid Oak Software's CYBERsitter [17] have taken this approach. These companies offer updates to their lists either monthly or on request. Unfortunately, sites undetected by these companies or new since the last update can still be accessed. New*View's NetGuardian [11] promises to offer a choice of blocking of explicitly forbidden resources and blocking of all resources not explicitly approved; it requires software both on the client machine and at a central site. SurfWatch also blocks access to resources that contain certain words likely to be associated with inappropriate content. Net Nanny [20] and CYBERsitter [17] use a similar approach for monitoring chat groups and email. However, word filtering alone will not detect obscene pictures.

Some people have called for voluntary self-rating of sites by their creators. One such proposal [10] describes an encoding called KidCode that would encode in the URL whether material was appropriate for children. For example, *http://www.sizzle.com/KidCode.21.violence/*, would indicate that the directory contained material suitable only for people age 21 and over because of violence. However, there would be no guarantee that everyone would rate pages, have the same standard, or even be honest in the ratings.

Finally, standards committees are considering setting standards for protocols that would allow communication of content ratings, so that software could request a rating for a URL from a ratings organization and then filter access based on the rating. On September 11, 1995, the World Wide Web Consortium announced the development of a Platform for Internet Content Selection (PICS) [21]. The Internet Engineering Task Force (IETF) is exploring formation of a workgroup for Voluntary Access Control. An earlier group, the Information Highway Parental Empowerment Group (IHPEG) [7], formed by Microsoft, Netscape, and Progressive Networks, has merged with the PICS effort.

Services that provide ratings of content will be useful. However, no one rating scale will be satisfactory to everybody. It is well known that different communities have different standards; what is acceptable in Berkeley, California may not be acceptable in Little Rock, Arkansas. Furthermore, different groups will disagree on what content children should see. For example, some librarians have objected to SurfWatch's blocking access to gay and lesbian materials [4]. Finally, if a new resource is created, there may be a delay before an organization rates it and distributes the rating.

Therefore, it seems valuable to allow for flexible and dynamic rating schemes under local control. Our method can be used independently or can be used to supplement the ratings provided by other organizations. A parent or teacher hearing of a new resource can place a rating in the database immediately without waiting for an organization to get around to rating it.

Description of Our Method

The key elements of Signet are the following.

- Access is controlled by a proxy server that enforces limits per machine or user, with a firewall to prevent bypass.
- Ratings are made using unmodified Web browsers, small modifications to a standard proxy server, and a rating daemon that maintains the ratings database.
- No cooperation from WWW servers is required.
- Any authorized person can rate materials and set access permissions in the proxy server, and raters can share rating lists.

- Permissions can be a combination of allowed and disallowed URLs and wild cards.

Signet addresses the parts of the Internet that can be addressed via a URL, such as HTTP, WAIS, FTP, Gopher, and Usenet news. It does not address email or chat groups.

For concreteness, imagine a school full of PCs running Web browsers and an Internet connection with firewall and proxy server. (The latter could be in the school or part of the network service offering.) There are two classes of users in our system: users who are authorized to control rating tables and those who are not. For convenience, we refer to the former as teachers and the latter as students.

Ratings

We assume a rating scale that would have levels in the same way that the movie industry uses G, PG, PG-13, and NC17. For the purposes of discussion, we will use a scale with three levels: "anyone", "13 and up", and "18 and up." There is also an orthogonal grouping into categories. A category could represent a subject such as history or even a day’s lesson plan for a teacher. The rating scale applies within each category, so that each rating is a category-scale pair. In addition, the rater of each rating is recorded.

Ratings can be given either for individual URLs (Uniform Resource Locators) or for expressions consisting of a URL followed by a wild card * that matches any string. Thus, a rating can specify that every resource contained (recursively) within a particular directory is rated with the same category-scale pair.

A resource may match the expression for more than one rating in the same category. In this case, the most specific rating applies. Thus, a subdirectory rating or URL rating takes precedence over a directory rating. For example, it is possible to specify that a whole directory is rated as "13 and up" except for a particular subdirectory rated as "anyone." Obviously, use of wildcards requires some judgment as to when it is appropriate.

Resources on the WWW can be changed at any time. We store the MD5 checksum [14] and the "Last-Modified" date with the rating for a file and allow access permissions to specify that the student proxy server should refuse access if the checksum has changed; the proxy could also notify the rater that the page should be rechecked. Obviously, checksums can be used only for individual URLs and not for directory or site ratings. For sites that are trusted not to lie about dates but not trusted to restrict material, the proxy might save the cost of recomputing checksums by only checking for a changed date. Some resources such as WebWeather [5] change frequently, and for these, the rater will have to rely on judgment about the consistency and trustworthiness of the site rather than on checksums.

Use of directory or site ratings is risky in that it is not generally possible to obtain a list of all URLs within a directory or site and in any case, resources can be modified or added. However, a rater may choose, for example, to trust a government agency directory with the name K12 that appears to contain resources for children. Note that approving a directory or site does not imply approval of links leading to other directories or sites, respectively, so that if the rated site or directory is itself trustworthy, the "six clicks to pornography" example described above for the LINQ approach [12] is avoided.

Rating sites, directories, and pages separately can lessen the impact of Web page changes on permissions in a way that a single rating per URL cannot handle. For example, a rater can rate individual pages as "anyone/unchanged-checksum", but also rate all resources at the site as "13 and up." In this case, access to a changed resource at that site could be allowed to older children with permissions including "13 and up", but denied to young children authorized only to access "anyone" pages. If this approach is used by raters, teenagers could browse widely even among changing pages, while young children would be more restricted.

Access permissions for restricted browsing mode are specified as a list of category-scale-rater triples. For example, if Mr. Smith's ratings are trusted by a fellow history teacher Mrs. Jones, Mrs. Jones could specify

History/13 and up/Jones
History/13 and up/Smith

to set access permissions for a history class. This would result in allowing access to resources whose most specific rating is "13 and up" or "anyone" but not to resources that are unrated or whose most specific rating is "18 and up". Wild cards can be used as well, so that */13 and up/* would specify any materials rated as "13 and up" by any rater.

Rating mode

In rating mode, the user can browse the WWW without restriction. When the user wants to rate a resource, the user requests a rating page.

Our method of requesting a rating page is to append a '!' to the current URL and submit this request to the proxy server. (The '!' should go before #anchors, if present, to avoid being truncated by the browser.) In rating mode, the proxy server detects the '!', and immediately returns the rating page, illustrated in Figure 2. This includes the current rating or ratings, possibly comments about why the rating was given (e.g. "includes sexual material"), and a form that allows submission of a new rating to the rating daemon. The form includes buttons for specifying the category-scale pair and for applying the rating to anything in a directory (including files and subdirectories) or to anything at the site. (We plan to extend the form to allow for creation of new categories as well.) This method of requesting rating pages fits within the current HTTP protocol without requiring that a special browser be used.

Figure 2. A rating page

An HTML page may contain inline images that would not normally be viewed separately. The rating for an HTML page is also assigned to URLs for inline images on the page, so that they do not need to be rated separately. In case of conflict between ratings derived from separate pages, the least restrictive rating applies, on the assumption that whatever caused the more restrictive rating was somewhere else on the page.

Restricted mode

The basic sequence in student mode is the following. The browser sends a request to the proxy, the proxy looks it up in the configuration tables for access permission, and if access is permitted, the proxy requests the resource from the remote server and forwards the response to the client browser. This mechanism is sufficient to handle requests for many resources. A typical request would be for an HTML page, possibly with inline images, or perhaps a PostScript or image file.

There are several common situations in which a request for a URL may be met by a response containing or referring to a resource with a different URL. One is when the user clicks on an image map, and the resulting GET is for a URL containing a '?' and a query string encoding the mouse click position. A second is submission of a form. A third is a directory name request, which may be redirected to another URL. A fourth is the "random link" or other cgi-bin command.

Under our rating procedures, a resource is normally rated after being received by the browser so that the response URL is known at rating time and placed in the ratings database. Unfortunately, in restricted browsing mode, the response URL is not known to the proxy at request time for comparing with the access permissions.

However, all is not lost. The proxy's action depends on the response code.

Often the remote server returns a redirection status code 301 or 302 with another URL, and the browser is expected to send a separate request for the new URL. In this case, the proxy forwards the response to the client because the new URL will be checked in a separate request.

If the remote server returns the resource itself along with a new URL, the proxy can check the URL for access permission. The HTTP standard makes no guarantees that the URL is correct; however, if false URLs should be a problem, correctness could be verified by comparing the resource checksum with the URL checksum stored in the ratings database. Alternatively, a redirection could be sent to the client.

The checksum of a resource can also be useful as an index into the ratings database. One such situation is when a response URL is ephemeral even though the resource itself exists over an extended period of time, as when the server encodes "session identification" in a URL such as *http://www.pathfinder.com/@@7XmEsaFcpwIAQEU8/time/magazine/magazine.html*. A second such situation is when (under HTTP/0.9) a remote server returns a resource without a new URL.

With an HTML form, submission causes the browser to send a POST or GET that includes a query string with information typed into the form. Forms are difficult to deal with because both the POST sent on submission and the URL (if any) sent on response may be too varied to be stored in the database and checksums are also so varied as to be useless. The most reasonable way to handle forms may be the use of wild cards in ratings and permissions.

These problems illustrate a fundamental naming inconsistency in the WWW: a resource can have more than one name, the name relationship is not necessarily known to the proxy, and names can change over time. The proxy server handles some instances of multiple names by canonicalizing the URL through simplification. Mirror sites must currently be handled explicitly in the ratings. In the future, naming problems may be reduced by the use of Uniform Resource Names (URNs) [18].

Security

Security is a fundamental problem when trying to impose restrictions on a previously free environment. We assume that the proxy server resides on a machine other than the client browser and that this machine is inaccessible to the students being restricted.

For our prototype, we use IP addresses to control security. All IP addresses are in restricted mode except when specifically authorized to be in rating mode. A teacher desiring unrestricted access to the Internet for browsing or rating pages requests an authorization form. If desired, authentication could use a challenge-response password scheme to avoid sending the password in the clear across a network. An authorization program then places the IP address of the teacher in a file of rating-mode IP addresses on the proxy machine. When the teacher signals the end of the session, the IP address is removed from the rating-mode IP address file.

Note that rating must never be enabled on the proxy machine, since access by a student through the proxy would have the rating IP address and be inadvertently authorized.

In order to get unrestricted privileges, a student hacker would have to forge the IP address of a machine currently in use by a teacher for ratings. The duplication of IP addresses would be likely to be detected quickly, unless the teacher’s machine was turned off. Leaving authenticated machines unattended and accessible to students should obviously be forbidden. To discourage such unsafe practices, in our implementation submission of a new rating does not immediately place it in the data base; it is held temporarily until the teacher does a "commit" and simultaneously exits rating mode.

Proxy caching proceeds as usual for non-rating requests in both rating and restricted modes. Rating pages are not cached but are constructed on the fly. Some browsers do local caching, however, and local caching of inappropriate materials during rating sessions could make them accessible to students if students have access to the machine later. The rater should in this case turn off local caching. If this degrades browsing performance unacceptably, the rating-mode proxy can be modified to place an immediate expiration date in the header, so that the local browser should not cache.

Access permissions for students may change from hour to hour if the teacher chooses to restrict students to accessing the lesson plan for the current class in a shared laboratory. There is no protection against students having the browser save material to a file for later viewing when the permissions have changed. But at least this material will be material that has been approved by some teacher for student use.

Implementation

We have implemented the prototype by modifying the publicly available source of the CERN httpd proxy server [3]. We have modified the source to allow for both a rating mode and a restricted mode as described above. Modifications involved touching about 7 places in five files and adding about 600 lines of C and included checking authorization for rating mode, generating rating pages, and checking response codes. A separate HTTP server called the rating daemon, about 1200 lines of C, is invoked via HTML forms for rater authentication and ratings database transactions.

The original CERN server is a good base on which to implement restricted mode because its configuration file can specify rules including wild cards to determine which resources should be passed or failed. The CERN server applies the first rule it finds. Since our permission specification requires that more specific rules take priority over less specific rules, we generate the configuration rules in lexicographic order so that more specific rules precede less specific rules. The main change to the code to implement restricted mode was to allow for delaying authorization until a response code is received, as described above. However, further modifications will have to be made to implement checksums and to improve efficiency for large ratings databases.

Planned Extensions

We plan to extend our initial prototype implementation to improve efficiency as the number of ratings gets large. In addition, we plan auxiliary programs to facilitate the rating process.

Efficiency

Efficiency has not been of concern as yet with the prototype because it has been used only experimentally. For the moment, we use linear search on the ratings database and on access permissions. As demands on the server grow, we plan to hash individual URLs and use a trie data structure for the rating expressions with wild cards.

Eventually, garbage collection may be useful to shrink the ratings database by eliminating "dead" or redundant ratings.

Facilitating the rating process

Just as people can get "lost" while browsing on the WWW because it is hard to keep track in your head of where you are in a large graph, people will have trouble keeping track of what pages they have rated, when they feel it is necessary to rate each page separately. Our plan is to provide a rating-progress tool for the rater.

The simplest form of rating-progress display would show a list of the links in the HTML resource, with color-coded marks or icons to show which ones have already been rated. It would be straightforward to implement this within the proxy by parsing the HTML and looking up the URLs in the database. More generally, it would be desirable to show rating progress for pages reached indirectly through multiple links from a given page. To restrict the search space, the tool could show rating progress as a breadth-first search of bounded depth or could show rating progress for links remaining within the same directory or site.

A tool that extracts information from bookmark files and annotation databases [13] to guide the rating process would also be valuable. Such rating-progress and annotation tools might run as servers, in keeping with the strategy of letting raters use their favorite unmodified browser, or in the browser if a market niche opens for specialized rating browsers or applets.

Social Issues

A generic problem faced by all schemes based on filtering at the client machine is preventing bypass. This could involve merely running a different browser or, in the extreme case, booting a different operating system. Our proposal, using a proxy and firewall, is secure against such attack. However, one might be satisfied with administratively requiring that approved browsers be used and only monitoring firewall logs occasionally for violations.

The effectiveness of Signet will depend on the quality of the ratings versus the dynamic nature of the WWW. Restricting children to explicitly rated resources via checksums is safe but will prevent access to many resources that have changed but are still suitable for viewing by children. Conversely, use of directory or site ratings would speed up the ratings process and allow children access to changing pages, but will not protect against unexpected introduction of inappropriate content. Finally, creation of resources on request through cgi-bin commands makes it impossible for a rater to be sure what might appear at some later date.

Many companies, agencies, and organizations would undoubtedly be happy to cooperate in facilitating filtering of access for children by following conventions such as redirection responses to image map and cgi-bin requests when a fixed resource is returned, use of client-side image maps [15], and self-rating of pages and directories.

No technical system can be totally and eternally safe. Children may walk down the street to a home with a less restrictive browsing mode, servers may deliberately issue misleading ratings or masquerade as trusted server. Ultimately it is more satisfactory to raise children, or hire employees, who have the maturity not to abuse the resource. We think of Signet as playing the role of a sturdy guardrail at a scenic vista, not an 8 foot fence topped with barbed wire.

Conclusions

Our prototype proxy server provides a flexible, dynamic means of rating resources on the WWW and controlling access. We hope that the availability of such a method will satisfy Congress that it is possible for parents and teachers to control WWW access so that children will not encounter inappropriate material. Unlike existing proposals, our method places control of both standards and access in the hands of parents and teachers so that they can apply their own local standards.

References

1. Ang, Peng Hwa, "Censorship and the Internet: A Singapore Perspective", *Proceedings of INET '95* , *http://inet.nttam.com/HMP/PAPER/132/abst.html*, June 22, 1995.

2. California Department of Education, "Electronic Information Resources Acceptable Use Policy District Guidelines", *gopher://goldmine.cde.ca.gov:70/00/C_D_E_Info/Technology/Acceptable_Use/Policy*, December, 1994.

3. CERN, "CERN httpd", *http://www.w3.org/hypertext/WWW/Daemon/*, April, 1995.

4. Cisler, Steve, "Children on the Internet (Draft)", *ftp://ftp.apple.com/alug/rights/kids.internet*, Apple Computer Company, June 20, 1995.

5. Davenport, Ben, "WebWeather", *http://www.princeton.edu/Webweather/ww.html*, 1995.

6. Hayes, Mary, "Working Online, or Wasting Time?", *Information Week* , May 1, 1995, pp. 38-51.

7. Information Highway Parental Empowerment Group, "Leading Internet Software Companies Announce Plan to Enable Parents to "Lock out" Access to Materials Inappropriate to Children", Netscape Press Releases, *http://home.netscape.com/newsref/pr/newsrelease29.html*, 1995.

8. Jackson, Colin, "Internet Policy in New Zealand", *Proceedings of INET '95* , *http://inet.nttam.com/HMP/PAPER/078/abst.html*, June 22, 1995.

9. Microsystems Software, "CyberPatrol", *http://www.microsys.com/cyber/default.htm*, August 18, 1995.

10. New, D., and N. Borenstein, "KidCode: Naming Conventions for Protecting Children on the World Wide Web and Elsewhere on the Internet Without Censorship", *ftp://ietf.cnri.reston.va.us/internet-drafts/draft-borenstein-kidcode-00.txt*, June 5, 1995.

11. New*View, Inc., "NetGuardian", *http://www.newview.com/*, August 20, 1995.

12. Quality Computers, "The LINQ - Custom Internet Access for Education", 1995.

13. Röscheisen, M., C. Mogensen, and T. Winograd, "Beyond Browsing: Shared Comments, SOAPs, Trails, and On-line Communities" *Proceedings of the Third International World Wide Web Conference*, Darmstadt, Germany, April 1995, *http://www-pcd.stanford.edu/COMMENTOR/*.

14. Schneier, Bruce, *Applied Cryptography: Protocols, Algorithms, and Source Code in C,* New York, Wiley, 1994.

15. Seidman, James, "A Proposed Extension to HTML : Client-Side Image Maps", IETF Internet-Draft, *ftp://ietf.cnri.reston.va.us/internet-drafts/draft-ietf-html-clientsideimagemap-01.txt*, August 8, 1995.

16. Skarecki, Eileen, Columbia Middle School, Berkeley Heights, NJ, personal communication, June 19, 1995.

17. Solid Oak Software, "CYBERsitter", *http://www.solidoak.com/cybersit.htm*, 1995.

18. Sollins, K., and L. Masinter, "Functional Requirements for Uniform Resource Names", Network Working Group, Request for Comments: 1737, *http://ds.internic.net/rfc/rfc1737.txt*, December 1994.

19. SurfWatch Software, "SurfWatch", *http://www.surfwatch.com/*, 1995.

20. Trove Investment Corporation, "Net Nanny: the best way to protect your children and free speech on the Internet", *http://giant.mindlink.net/netnanny/home.html*, 1995.

21. World Wide Web Consortium, "W3C Content Selection: PICS", *http://www.w3.org/pub/WWW/PICS/*, September 11, 1995.

About the Authors

Brenda S. Baker [*http://www.cs.att.com/csrc/baker.html*]
Eric Grosse [*http://www.cs.att.com/csrc/grosse.html*]
AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974
bsb@research.att.com
ehg@research.att.com