

The UNITE System: Distributed Delivery and Contribution of Multimedia Objects Over the Internet*

Cedric Deniau*, Michael Swink*, Ron Aust+, Joe Evans*,
Susan Gauch*, Jim Miller*, Douglas Niehaus*

*Department of Electrical Engineering and Computer Science
+School of Education
University of Kansas
Lawrence, KS

Abstract

Over the past few years, the world has seen a growing interest in the Internet. E-mail initiated this interest and was the biggest traffic generator for several years. As the Internet grew in popularity, other tools emerged: ftp, gopher, archie, and the World-Wide Web. Connectivity to the Internet blossomed from a few computer specialists at research institutions to include businesses, schools, and home users. At the same time, the ability to create, store, and view multimedia information became widespread. Today, we see a proliferation of sites storing and distributing multimedia information on an ever-increasing range of topics to an exploding number of users.

This paper describes the UNITE system which provides browsing and search of taxonomically indexed resources in a wide range of media types (text, images, hypercard stacks, etc.). The server provides remote access to Science and Mathematics curricular materials by teachers and students in K-12, however it can be easily adapted to work with any taxonomically structured domain. The server software supports mirroring, which helps distribute the client load, and enables the client to try alternative servers if its first choice is unavailable.

The server can interoperate with standard WWW browsers (Mosaic, Netscape) but, in addition, we have developed our own client software. Since most of our users connect to the Internet via modems, the UNITE client has features which reduce network load, and thus improve performance on low bandwidth networks. It also provides a more tailored user interface to the system resources than is available from standard browsers. Finally, users are active participants in the project. Through a review mechanism, they can contribute new resources to the database.

1 Introduction

The Information Highway is becoming a reality. The increase in access to the Internet by the public at large, combined with the development of easy to use graphical browsing interfaces, for example, Mosaic and Netscape, have lead to an

explosion in the information being added. In particular, the World Wide Web (WWW) is being used to present an exponentially growing amount and range of information through which people can browse. Unfortunately, too much information can be the same thing as not enough information. If the information you seek is buried under an avalanche, is it really there? The WWW is growing at such a rate that it is hard to locate information of interest. To give a feel for the magnitude of the problem, the Lycos system indexes over 860,000 Web documents from 34,000 sites and is able to add 5,000 documents a day [4]. The WWW is growing quickly because it provides an easy to use interface (pointing and clicking at items on the screen) for users, uses simple standards (HTML, MIME) which allow multimedia documents to be exchanged, and provides a simple unified interface to a range of useful tools (ftp, gopher, news, etc.).

Development of the Unified Network Informatics for Education (UNITE) system was supported through the U.S. Department of Education OERI office Star Schools initiative. The challenge was to develop a system that allows educators and students to remotely contribute and access multimedia educational resources for advancing K-9 mathematics and science education. The initial target audience included educational partners and 52 schools in Michigan, New York and Pennsylvania that are part of the Great Lakes Telecommunications Collaborative. The K-9 teachers and students, who are the target users, generally have only minimal keyboard entry and mouse manipulation computer skills. Among the criteria that the design team faced during the early development were: 1) designing an interface that was easy for novices to use; 2) organizing educational resources in ways that are consistent with existing practices in schools (e.g., curricula and resource types) while encouraging them to consider new ways for using the resources; 3) developing a system that allowed distributed contributing and reviewing of submitted resources; and 4) providing both browsing and search mechanisms that accommodate the diverse strategies that users employ in locating educational resources.

UNITE provides a central repository for educational resource materials, allowing the information to be easily located. By creating a cus-

*This research and development was supported through funding from the U.S. Department of Education OERI office (R203A20020) that was administered by the Great Lakes Telecommunications Collaborative.

tomized graphical user interface, we have created a system which is accessible to casual computer users. Finally, we involve the users themselves in the evolution of the database by encouraging them to contribute resources that they create. However, we have a series of editors which approve and improve the contributed materials, providing quality control.

This paper first gives an overview of the WWW and how servers and clients work. Then it presents our approach to these problems, focusing on our search capability, the simple safe interface, and how UNITE supports the sharing of educational resources.

2 Overview of the World Wide Web

The WWW was started at CERN by Tim Berners-Lee in March of 1989 as the HyperText Project, and is officially described as a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents [7]. Initially, its main goal was to provide a common (simple) protocol for requesting human readable information stored on remote systems using hypertext as the interface and networks as the access method [9]. Hypertext is similar to regular text since it can be stored, read, searched, or edited, but with an important exception; hypertext contains connections within the text to other documents. The generality and power of the WWW becomes apparent when one considers that these links can lead literally anywhere in cyberspace; to a neighboring file, another file system, or another computer in another country.

The WWW Project adopted a distributed client-server architecture. The client supports the user as she selects links inside documents by fetching the new document desired, while the server receives the requests generated by selecting a link and responds by providing the client with the required document. At the beginning of the WWW Project, the client was a line mode browser which performed the display of a hypertext document in the client hardware and software environment. For example, a Macintosh browser uses the Macintosh interface look-and-feel. In September of 1993, NCSA released the Mosaic browser for the most common platforms, X-windows, PC/Windows, and Macintosh. Since Mosaic allowed documents with images to be viewed and handled new media formats such as video and sound using helper applications, it became the WWW browser of choice for those working on computers with graphics capability. However, what may have been Mosaic's most important property was that effectively subsumed a number of traditional services (i.e. ftp, telnet, gopher ...), and given its intuitive hypermedia interface, it became the most popular interface to the WWW.

Today the WWW is growing at an astonishing rate. From January to December 1993, the amount of network traffic across the National Science Foundation's (NSF's) North American network attributed to WWW use multiplied by 187 times. In December 1993 the WWW was ranked 11th of all network services in terms of sheer traffic - just twelve months earlier, its rank was 127.

In June 1993, Matthew Gray's WWW Wanderer, which follows links and estimates the number of WWW sites and documents, found roughly 100 sites and over two hundred thousand documents. In March 1994 this robot found 1,200 unique sites. A similar program by Brian Pinkerton at the University of Washington, called the WebCrawler, found over 3,800 unique WWW sites in mid-May 1994 [7], and found 12,000 WWW servers in mid-March of 1995.

There is no indication that this torrid pace is slackening, quite the opposite in fact. The major challenge posed by the WWW is clearly one of organizing and making a wealth of information accessible, not of making it merely available. The rest of this section gives an overview of important properties of WWW servers and clients, which help determine what services the WWW can provide, and the processing and network support required to support them.

2.1 WWW Servers

WWW servers are programs running on host computers which support simultaneous access by multiple users, using their WWW clients, to the WWW resources resident on the host. In keeping with the client/server paradigm, they respond to a specific set of commands (their protocol) in predictable ways.

2.1.1 Protocols

The WWW has used the Hypertext Transfer Protocol (HTTP) since 1990. HTTP is an application-level protocol with the compactness and speed necessary for distributed, collaborative, hypermedia information systems. It is generic, stateless, object-oriented protocol which can be used for several kinds of tasks [2]. HTTP builds on the discipline of reference provided by the Universal Resource Identifier (URI), as a location (URL) or name (URN), for identifying the resource upon which a method should be applied. Messages are passed in a format similar to that used by Internet mail and use the Multipurpose Internet Mail Extensions (MIME)[2].

HTTP is based on a request/response between client and server. The client establishes a connection with a server and submits a request consisting of a request method, URI, and protocol version, followed by a MIME-like section containing request modifiers, client information, and optional body. For most implementations, the connection is established by the client prior to each request and closed by the server after each response. The closing of the connection by either or both parties always terminates the current request, regardless of its status [2].

A client request includes the method which should be applied to the resource requested, the resource identifier, and the HTTP version. There are seven different methods allowed in HTTP: GET, HEAD, PUT, POST, DELETE, LINK, UNLINK [2]. The GET method retrieves whatever information is identified by the Request-URI. If the Request-URI refers to a data-producing process, it is the produced data which is returned as the entity in the response and not the source text of the process [2]. The HEAD method is identical to GET except that the server must not re-

turn any entity body in the response. The meta-information contained in the HTTP headers in response to a HEAD request should be identical to the information sent in response to a GET request[2].

The POST method is used to request that the destination server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the request line. POST creates a uniform method to achieve the following functions: annotation of existing resources; posting a message to a bulletin board, newsgroup, mailing list, or similar group articles; providing a block of data (usually a form) to a data handling process; extending a database through an append operation[2].

The PUT method requests that the enclosed entity be stored under the supplied Request-URI. If the Request-URI refers to an existing resource, the enclosed entity is considered a modified version of the original. If the Request-URI does not point to an existing resource, and the requesting user agent is permitted to define the URI a new resource, then the server creates the resource with that URI [2].

The DELETE method requests that the server delete the resource identified by the Request-URI [2], while the LINK method establishes one or more *link* relationships between the resource identified by the Request-URI and other existing resources. The LINK method does not allow any entity body to be sent in the request and does not result in the creation of new resources [2]. The UNLINK method removes one or more *link* relationships from the existing resource identified by the Request-URI. The removal of a link to a resource does not imply that the resource ceases to exist or becomes inaccessible for future references [2].

2.1.2 Server Features

The features provided by different servers vary, but currently there are two popular servers, those produced by NCSA and CERN. The features discussed in this section are common to both, and are representative of services which any reasonable HTTP server should provide. One feature, directory indexing, allows users to view contents of directories on the server using their WWW clients. Depending on how the server was configured, the listing might specify distinct icons for different file formats. A header and trailer file could be included in the listing to give the user more information on the directory contents.

CGI scripts, a particularly powerful feature of HTTP servers, are used to run programs on the server side. These scripts are primarily used to act as gateways between the WWW programs and other software like *finger*, *archie*, or database software. Image maps, which associate HTTP links with different areas of an image, are another popular use of CGI scripts. The images are virtually segmented so when a user clicks on different parts of the image, he is taken to different URLs.

Server features allow the server administrator to standard include files within all HTML documents provided by the server, creating the ability to include a signature block with every document. When the signature contents change only one file

needs to be changed instead of having to change every file containing the signature. The server can also restrict access to certain documents or directories. There are two ways this can be done: (1) in a configuration file, the server administrator can specify certain hosts that are allowed or denied access to documents; or (2) the administrator can specify that the server should ask for a username/password when access to a particular file or directory is requested.

The features mentioned above are a subset of the features implemented by full fledged WWW servers. Although these features assist the user in navigating the Internet, the most important feature of a WWW server is its understanding and response to a standard protocol, providing access to documents from a variety of browsers.

2.2 WWW Clients

WWW clients, often called browsers, mediate between the user and WWW servers by presenting the documents retrieved in a manner best suited to the user's platform, and makes requests to the appropriate server when the user selects a hypertext link. Currently, the most popular browsers are Netscape and Mosaic, both of which are available for multiple platforms (PC, Mac, UNIX based stations).

2.2.1 HTML

The HyperText Markup Language (HTML) is a simple markup language used to create hypertext documents that are portable across platforms. HTML documents are SGML documents with generic semantics appropriate for representing information from a wide range of applications. HTML markup can represent hypertext news, mail, documentation, hypermedia; menus of options; database query results; simple structured documents with in-lined graphics; and hypertext views of existing bodies of information [3].

HTML has evolved over time, leading clients to render HTML documents differently. Currently there are three versions of HTML, the most common being HTML 2.0. HTML 2.0 introduced *forms* which support more complex interaction between users and servers by enabling them to supply information beyond simple item selection. For example, forms are commonly used by the user to specify character strings for searching, to provide user-specific data when interacting with a business's WWW page, and to provide written text of many kinds in other situations. The Netscape browser has extended HTML by adding extra tags and tag modifiers (i.e. CENTER, BLINK, ...) which provide an enriched set of document formatting controls to the HTML author. Implementations of HTML 3.0 recently become available, which adds the features of tables, mathematical equations, and text wrapping around pictures.

2.2.2 Client Features

The most popular Web browsers, Netscape and Mosaic, provide similar feature sets. They have a consistent mouse-driven graphical interface, and support the idea of using point-and-click actions

to navigate through documents. They have the ability to display hypertext and hypermedia documents in a variety of fonts and styles (i.e. bold, italics, ...), layout elements such as paragraphs, lists, numbered and bulleted lists, and quoted paragraphs [7]. All of these are defined in the HTML text of the WWW document being rendered.

The browsers have the ability to use external applications to support a wide range of operations. For example, they can be used to view MPEG or QuickTime movies, listen to audio files, or display graphical images. With *forms* support, they can interact with users via a variety of basic forms elements, such as fields, check boxes and radio buttons. They provide hypermedia links to and support for the following network services: FTP, telnet, gopher, NNTP, and WAIS. In addition, they can: (1) allow remote applications to control the local display; (2) keep a history of hyperlinks traversed; and (3) store and retrieve a list of documents viewed for future use.

WWW clients often add new abilities along divergent design paths. However, through HTML, they continue to provide a unified and uniform interface to the existing information which is the basis of the WWW's popularity.

3 UNITE

The UNITE project has developed an enhanced WWW server and a Macintosh client which provide access to a multimedia database of K-12 Mathematics and Natural Science educational resources. Database contents are contributed by our users after which they undergo a two-stage review process (see Figure 1). The database can be browsed via the hierarchically structured curriculum taxonomy or the graphical search window can be used to intuitively specify Boolean queries. Natural language searching via WAIS is also planned.

3.1 User Community Characteristics

Rapidly evolving information technologies are significantly altering social and economic structures both globally and locally. Schools, once dominated by rigid top-down systems, are responding by encouraging more grass-roots opportunities for teachers and producing students who are capable of deriving flexible solutions for a wide-range of problems. One indicator of this restructuring is the movement toward teacher collaboration and empowerment [6, 8].

To address the needs for school restructuring and teacher empowerment, systems for distributing educational resources must provide: 1) mechanisms that allow teachers and students to contribute their ideas, 2) a review process to measure the consistency and 3) quality of resources and structures for easily locating valuable resources.

Value is relative. Educational resources that are valuable in one school may be inconsistent with the curricula needs of another school. Moreover, resources are only valuable when they are used; and they are more likely to be used if they have advocates. Educators and students need easy-to-use mechanisms for contributing resources so that they can tailor resources to local needs, and in so doing, become vested in the idea of sharing and using resources. The mechanisms for reviewing the resources should also be

distributed so that individuals who are familiar with the local needs can be involved in the review process.

Earlier research and development efforts in designing network information services for educators indicated that teachers initially found hierarchical curricular browsing structures to be an easy way to locate information [1]. As the teachers used the browsing mechanism they became familiar with available resources which included; lesson plans, field trip descriptions, lab activities, videos and professional development and student created materials. With this familiarity of the information domain also came a desire to more precisely focus their queries. They were no longer content with wading through the resources in the "Natural Science" curriculum on "Ecology." Instead, they began to focus their queries more precisely with questions such as, "I need ecology lab activities for grade 4 students that help to develop observation and analytical skills." These kinds of questions require more advanced document indexing and query mechanisms than the parent-child hierarchy needed for a curricular browsing structure.

3.2 Database

UNITE uses databases to organize the available resources. Each database has a configuration file associated with it which describes the structure, format, and treatment of the database records. Databases can store several classes of information and must be capable of managing significantly different kinds of data (i.e., software, text, video, sound, etc.). The database configuration language is used to specify record structure, and defines four basic objects: TABLE, ENUMERATION, RECORD, and DATABASE OBJECT. This language provides a centralized user-readable and modifiable specification of the data stored and its treatment by the system. Figure 2 illustrates a simple example of a database configuration file.

The DATABASE OBJECT section defines a UNITE resource's fields and field attributes, using one line per field. The first attribute is the field type which can either be a predefined or a user defined type. The predefined types are: string, integer, and freetext. The user defined types are either enumerations or records. The next attribute specifies how many items the field can contain: One, OneOrMore, ZeroOrMore, or Zero. The third attribute specifies how the field is used during a search, while the last attribute is the name of the field used by the database.

In the example of Figure 2, the last line of the database record section specifies that the field "PhysMedia" is of type "PhysMediaT" which is an ENUMERATION representing the set of values "CD", "LP", and "VHS". The "PhysMedia" field may only hold one entry. If "PhysMedia" needed to hold a list of one or more entries then "One" would have been "OneOrMore". The RECORD objects use the same set of parameters as the DATABASE OBJECT, but the record defined is used as a type for a field in the DATABASE OBJECT rather than defining an object directly.

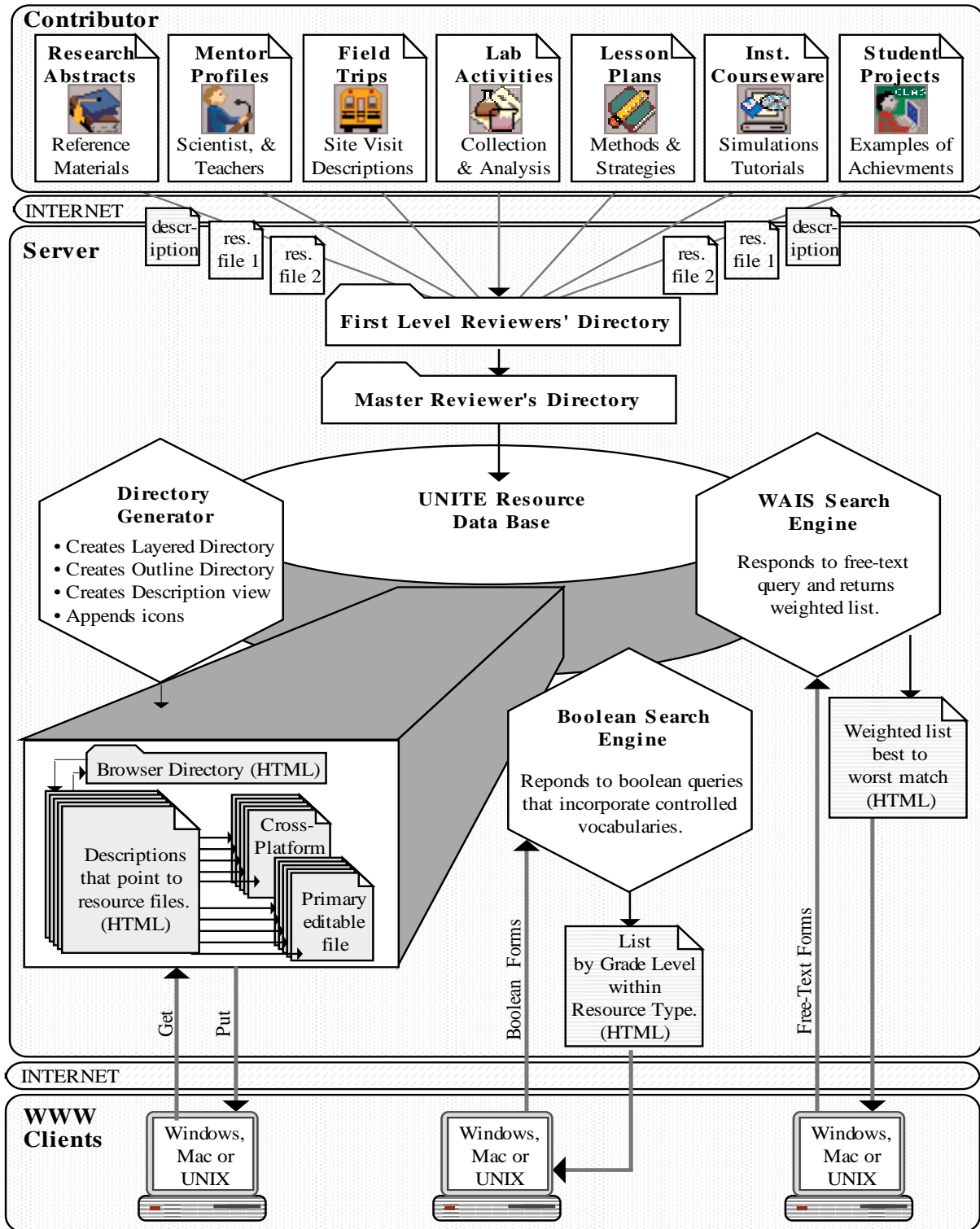


Figure 1: UNITE: An Information Service for Contributing, Coordinating and Distributing Educational Resources

In our example the field "FileDesc" is of type "FileDescT" which is a record containing the "FileSize" and "FileName" fields. The ENUMERATIONs defined are also used as type definitions and specify a specific set of field values. In the example, "PhysMediaT" is a simple list, while "CurricT" is a hierarchical list.

The TABLE section gives extra flexibility to the system by defining a mapping from one set of values to another. In the example, the table "PhysMedia_Table" maps the elements of the enumeration "PhysMediaT" to the icons used to represent them in the generated HTML. Another example might be to map each field of a database record to its proper printing format. Both of these tables would be used to help give a consistent look and feel to the HTML documents produced.

Following the definition of a database, the records need to be entered and ultimately presented to the user. The records are indexed using the CSO database and are then rendered in HTML. The HTML generation is currently done at contribution time but could be done on-demand if it were desirable to trade time for space.

```
TABLE "PhysMedia_Table" {
"CD"           "CD_icon.GIF";
"LP"           "LP_icon.GIF";
"VHS"          "VHS_icon.GIF";
"DEFAULT_ENTRY" "Default_icon.GIF";
}

ENUMERATION "PhysMediaT" {
"CD" "LP" "VHS"
}

ENUMERATION "CurricT" {
"Mathematics"
"Natural Science" {
"General Natural Science"
"Physical Science" {
"General Physical Science"
"Properties of Matter" {
"General Properties of Matter" }
"Electricity-Magnetism" }
"Common Themes"
} }

RECORD "FileDescT" {
"integer" "One" "NoSearch" "FileSize";
"string" "One" "Keyword" "FileName";
}

DATABASE_OBJECT UNITEResource 1994092001 {
"string" "One" "Keyword" "Title";
"uid" "One" "NoSearch" "IDNumber";
"FileDescT" "One" "NoSearch" "FileDesc";
"CurricT" "One" "Keyword" "Curriculum";
"PhysMediaT" "One" "Keyword" "PhysMedia";
}
```

Figure 2: Data Base Configuration Language Example

3.3 Server

The UNITE server is based on HTTP which allows it to be used as a regular Web server. It supports the GET, DELETE, POST, PUT, and SEARCH methods. It runs CGI scripts and supports user directory access. On the other hand, the UNITE server does not support directory indexing, authentication, and a number of other services which were not required for our driving application.

The SEARCH method is a unique feature of the UNITE server. It was created to allow the server to directly respond to queries from the client rather than via CGI scripts. It also defines a search syntax, which has yet to be done by the Web community. To support access from other WWW clients, which do not support the SEARCH method, a generic forms interface to the search capability was built. This interface allows the user to select which database and which fields of the database to search on.

However, the forms interface uses several separate HTML pages to present the search interface, which requires either the client or the server to preserve information across request boundaries, which contradicts the stateless orientation of HTTP. To solve this problem, the server generates HTML documents which preserve the required information in a form invisible to the user. This information is then sent back to the server with each exchange providing the server exactly what it needs to know from previous user interactions. This effectively builds state into the stateless HTTP protocol.

The current search engine used for UNITE is CSO. CSO was originally written for a simple name service, a computer resident phone book, but required only slight modified to fit UNITE's needs. It can keep relatively small amounts of information about a relatively large number of objects, and provide fast access to that information over the Internet [5]. CSO also allows for wild card expansion which permits users to be conveniently vague when formulating queries. The main problem with CSO is that it is inappropriate for large target text items and it does not have Boolean search capabilities. This motivated us to implement set operations (i.e., and, or, contains, ...).

Another search engine that is currently being integrated into the UNITE server is WAIS. WAIS (Wide Area Information Server) is a free text search engine which would support natural language queries and allow the user to perform inexact searches. Another advantage of WAIS is that it returns a ranked list of matches. This allows the user to select resources that have the best match to the query instead of having to browse through a set of resources to find the best.

A necessary part of future work for a truly robust system would be the addition of authentication. A design has been discussed but not implemented. The design calls for a separate database containing user information (i.e. name, username, password, etc ...). Every time a request comes in, the server would query the user database for a proper username/password combination. Users could be a member of a group or groups and each group or user would have specific permissions associated with them. Each

database record would also have permissions associated with them, noting which groups or users are allowed to view them. At this point, the server would match users with their groups and then would try to match the user's groups with the database record's groups.

Similar to other more generic Web servers, the UNITE server needed to handle a large number of requests in a small amount of time. To test response, an HTML document containing more than 200 in-lined images was generated and a Web client requested the document. After approximately 50 GETs for the images, the server ceased responding. To address this problem, we propose adding a new method to the server similar to the FTP *mget* method. For example, a client would send the MGET method with a list of the documents it wants to download. The server then serves the client by sending each file with pre-defined separator between documents. This would reduce the load on the server since only one request is performed at a time. This idea is currently being studied as a possible solution to the problem. Another way to alleviate this problem, which is currently implemented by the UNITE client, is to cache images on the client side. This reduces the number of requests since the images are already on the user's machine.

3.4 Client: User Interface

We based the initial design for the client's user interface on a prototype developed during earlier pilot projects [1]. This design used a layered approach to represent a curriculum hierarchy browsing structures similar to the approach used to represent directories in typical graphical user interfaces. Novice users understand how to navigate this structure and they are successful in locating useful resources. They also appreciate the use of icons to represent the various resource types. The pilot users also provided several suggestions for improving the client interface. Key among these were suggestions for a more efficient browsing view of the curriculum's hierarchy, and the ability to locate items using multiple selection criteria.

We began the design process for the current user interface in early 1993. In 1993, the most prevalent browsers for distributing resources on the Internet the University of Minnesota's "Gopher" and Dartmouth's "Fetch." However, WWW development was underway at the University of Kansas, most notably the "Lynx" text-based browser, and NCSA was demonstrating an early version of the Mosaic client for the UNIX platform. One of the reasons we decided to pursue WWW development was that HTML offered a more extensible means for designing user interfaces. We knew WWW clients were planned for other platforms but a Macintosh client was not available and the 52 pilot schools in the Great Lakes Collaborative were seeded with Macintosh 610 computers. We decided to develop a Macintosh client that was tailored for the needs of this user community while maintaining server and document compatibility with other WWW development.

The first client delivered in the Fall of 1993 offered a layered folder view as well as an outline view that showed the entire curriculum hi-

erarchy in a single scrolling window. Icons are used to represent resource types in lists and the grade level designation appears at the end of list items. Resources are accessible through a simple point and click interface. We refined the scheme for indexing the documents to be consistent with the emerging standards of the National Science Teachers Association (NSTA) and the National Council of Teachers of Mathematics (NCTM). This indexing allowed us to implement a search window for specifying queries according to several dimensions including; TITLE, GRADE LEVEL, CURRICULUM, PROCESS SKILLS, RESOURCE TYPE and MEDIA TYPE.

Our recent user interface development has centered on incorporating recent additions to HTML for presenting an easy-to-use interface for constructing Boolean queries using standard WWW clients. We have also implemented features in the Explorer client to easily identify selections in extensive hierarchical lists. The Client Search Window (Figure 3) shows the user constructing part of a Boolean query by specifying CURRICULUM values. Note that the selected CURRICULUM field is highlighted in the separate window on the left side of the screen. Having selected the segment of the "Natural Science Curricula" representing "General Properties of Matter" the parent portions of the curriculum hierarchy "Properties of Matter, Physical Science" and "Natural Science" are shown as partly filled. Curriculum is one of the controlled vocabularies used in indexing the educational resources. Other controlled vocabularies shown in this view include: RESOURCE TYPES, PHYSICAL MEDIA, GRADES and PROCESS SKILLS. These controlled vocabulary fields may be coupled with the remaining text entry fields to form complex queries for specifying resources.

3.5 Distributed Aspects

The success of UNITE as a model for distributed access to collections of information across the Internet depends on a number of factors, but the single most important is ensuring that the system provides good support for adding to the database. Our driving application is a particularly good example of this since the educational materials are contributed by the users of the system, rather than by some central authority. However, we believe that this is one of the strengths of the Internet and represents an important aspect of systems which look toward the future of the National Information Infrastructure.

First and foremost, the success of such a database requires the participation of users, who are often the best qualified people to generate source material as practitioners in the field. With this in mind, we implemented a method we called the Contribution Process, supported by software called the Contributor. The Contributor must first know to which database the user wishes to contribute a record. Then the Contributor prompts the user to enter information for each field of the database. For example, if the user wanted to contribute a record to the database defined in Figure 2, the Contributor would prompt the user to enter information for the "Title", "FileSize", "FileName", and so on.

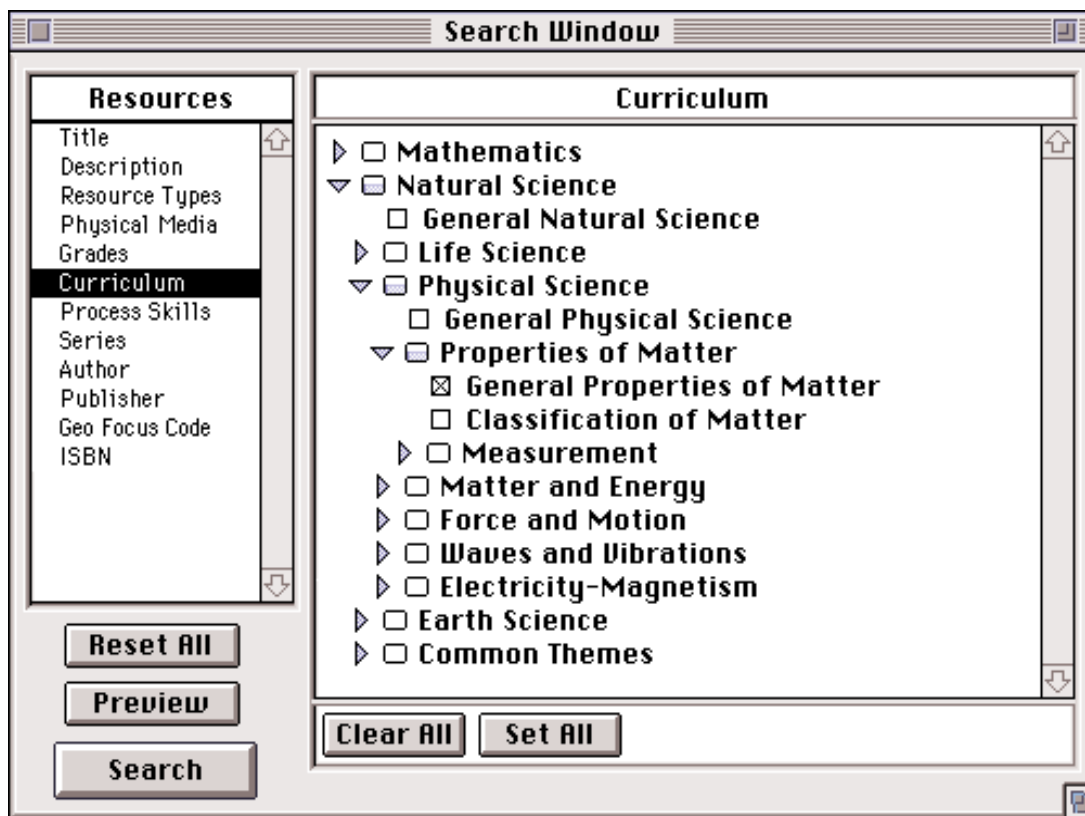


Figure 3: The Explorer Client Search Window

The Contributor then sends the newly defined record to a local reviewer. The local reviewer's duties are to make sure the record relates to the application area to which it is being contributed, that it is properly formatted, and is well written. The local reviewer then passes the record along to a master reviewer whose duties are to check the local reviewer's work and approve or reject the record for inclusion in the database. From there, the record is sent to the UNITE server for integration in the database. Currently this is done using FTP but in the future the PUT method will be used. The idea here is that the record is sent to a centralized server, keeping the databases consistent by ensuring that there is only one place where new information is introduced to the system.

Once the record is transferred to the server, a series of steps are taken to add the record to the proper section of the database. The first step is to generate an HTML document following the format of the database record definition. Note that this is done on the server and not by the user, keeping a consistent look and feel for all the HTML representations of the database records. Then a database record is created and added to the database. Then it is time to rebuild the layered and outline views and the data structures which will allow the users to request or search for the newly added record. This Contribution Process is run nightly and therefore the turn around time for a newly defined record to be added to the database is usually 24 hours.

To distribute server load and improve avail-

ability, UNITE supports a method of creating multiple copies of a database on multiple server machines, which is called *mirroring*. The mirroring process is run every night and operates in two modes. The first mode makes a complete copy of the database file structure, including all HTML documents and all indices built by CSO, to the mirrored server. This method is usually used for newly added servers or those that have been inactive for a long period of time. The second method is used for updates to active mirrors. It determines the set of files modified since the last update of the mirrored server and sends. None of mirrored servers are allowed to receive contributions, thus helping to ensure database consistency.

4 Conclusions and Future Work

This paper described several aspects of the design and development of the UNITE system at the University of Kansas. The system provides the ability to browse and search hierarchically indexed resources in a wide range of media types (text, images, multimedia, etc.). The server provides remote access to Science and Mathematics resource by geographically distributed K-12 teachers and students, but it can be easily adapted to work with any hierarchical structured domain. For example, we have recently constructed a similar database of information about area businesses for the Lawrence, KS Chamber of Commerce.

The server software supports mirroring, which helps distribute the client load, and enables the

client to try alternative servers when its first choice is unavailable. The growth of the database is supported by the contributor software which helps manage the introduction of material produced by geographically distributed users into the database. The system has been in use by its target audience for over two years and services thousands of requests per week. The experience gained in implementing the system has demonstrated a number of ways in which providing usable services with the WWW presents unique challenges. As such it has demonstrated the need for modifications of current methods, the need for new abilities, and the fact that the WWW is still a vital and evolving entity.

One area of new research that is underway concerns the relative benefits of different browsing structures on the user's understanding of the information domain. The browsing structure based on a single indexing dimension (e.g. curriculum) are easy to use but they provide a somewhat constrained understanding of the scope of the resource. We have recently implemented the "EduLette" browser that randomly selects resources from a given domain. We plan to refine this random browser so that user become actively involved in identify the dimensions of the domain they wish to investigate. We anticipate that this targeted random browsing coupled with the existing browsing structures will elicit a more robust understanding of the domain and result in the user constructing more meaningful free text queries.

We are continuing to refine the interface and features of the UNITE system based on user recommendations and the goal of developing a useful system for a wide range of users. This includes accessibility from numerous platforms, improvements to the contributing and review functions and the ability to easily locate meaningful resources in the rapidly expanding collections on the Internet.

We are also investigating the application of the UNITE platform to other possible research areas. We are beginning to apply this technology to the needs of a small working groups. This will give us the opportunity to investigate how to use WWW and HTML methods to provide effective user interfaces for tools supporting group activities. We are also interested in applying this technology to providing user interfaces for sophisticated information retrieval approaches to database access, and for providing access to new types of information including real-time video.

References

- [1] R. Aust. Designing Network Information Services for Educators. *Machine-Mediated Learning*, 4(2&3), 1994, pp. 251-267.
- [2] T. Berners-Lee, R. T. Fielding, H. Frystyk Nielsen, K. Hughes. Hypertext Transfer Protocol - HTTP 1.0. *INTERNET-DRAFT*. March 8, 1995
<ftp://ietf.cnri.reston.va.us/internet-drafts/draft-ietf-http-v10-spec-00.txt>
- [3] T. Berners-Lee, D. Connolly. HyperText Markup Language - 2.0. *INTERNET-DRAFT*. March 29, 1995

<ftp://ietf.cnri.reston.va.us/internet-drafts/draft-ietf-html-spec-01.txt>.

- [4] J. December. New Spiders Roam the Web. *Computer-Mediated Communication Magazine*, 1 (5), September, 1994.
- [5] S. Dorner. The CSO Nameserver: A Description. Technical Report, Computing Services Office, University of Illinois at Urbana-Champaign. July 1989.
- [6] J. Goodlad. Better teachers for our nation's schools. *Kappan*, 72(3), 1990, pp. 184.
- [7] K. Hughes. Entering the World-Wide-Web: A Guide to Cyberspace. Enterprise Integration Technologies, May 1994.
- [8] G. I. Maeroff. A blueprint for empowering teachers. *Kappan*, 69(7), 1988, pp. 472-476.
- [9] WorldWideWeb: Proposal for a HyperText Project CERN, 1989.
<http://www.w3.org/hypertext/WWW/-Proposal.html>.