Mosaic and WWW as Part of an Overall Architecture

Bernard S. Hirsch

Hewlett-Packard Company
930 East Campbell Road
Richardson, Texas 75081
(214) 699-4197
(214) 699-2528 [fax]
bernie@bsh3185.ssr.hp.com
http://bsh3185.ssr.hp.com/home.html

Abstract

This article will benefit information technology architects and knowledge workers because it will help explain some of the business, architectural, and practical benefits and implications of deploying Mosaic and World-Wide Web (WWW) technology. A holistic view is taken with respect to (1) concrete applications and examples of Mosaic/WWW deployment, (2) benefits of such a deployment, and (3) how Mosaic/WWW integrates into an enterprise's information technology infrastructure. This article will explain how Mosaic/WWW can be used in such a way that the knowledge worker gains greater access to the information that is needed and thus becomes more productive. Further, the deployment of Mosaic/WWW leads to greater sharing of information and a focus on better communications and teamwork. It will be also be shown how Mosaic encourages utilization of current information and leverage and integration of existing technologies.

0 Introduction

The use of Mosaic and World-Wide Web (WWW) is being accelerated by individuals who need to access diverse information from multiple sources. This is largely due to the manner in which Mosaic accomplishes this information access to disparate, multimedia documents that reside on geographically distributed media around a workgroup, enterprise, or across enterprises. The ease of use and consistency through which different document types are accessed obviates the complex process by which information may need to be accessed. Further, the use of hypermedia linking technology allows navigation to additional related documents that would normally not be seen.

Other features of Mosaic and WWW have also contributed to its popularity as a front end browser to and repository of text, rich text, images, audio, video, and other document types. These include the concepts of (1) uniform resource locators (URLs), whereby all documents are accessed in the same, consistent manner, globally using a uniform information space, (2) an open architecture, whereby (a) the native format of documents are used as they become part of the Web and (b) existing protocols are all supported to access and integrate with the Web, (3) distributed infrastructure, that encourages distributed authoring, publishing, and sharing of information by end-users, and (4) multi-platform availability, whereby WWW servers and browsers, such as Mosaic, are available on most computing platforms.

In the first section, several practical applications of Mosaic/WWW in a production environment are described. Here, the creation, use, and administration of the environment is explained in detail. Also in this section, the perceived and projected benefits are described from the point of view of the user.

The technology perspective is taken in the next section, whereby the anatomy of Mosaic/WWW is examined so that it may be formally placed in an enterprise information technology architecture. Specifically, the information management services, the distributed services, and the utilities services of Mosaic/WWW are formalized and positioned among a complete framework of information technology. In this way, for example, Mosaic/WWW can be described as providing certain collaborative functionality along a spectrum of workgroup technology.

1 Applications of Mosaic and WWW

In this section, real world uses of Mosaic/WWW are presented in a detailed and practical fashion. The benefits of utilizing Mosaic/WWW in these examples is also discussed, and in some cases these benefits are perceived and projected due to the relative newness of the deployments.

1.1 Knowledge Worker Web

Figure 1

A knowledge worker Web has been created using Mosaic/WWW technology to collect, reference, correlate, and categorize information by subject area as needed by the worker. Specifically, a WWW server has been developed using the HTTPD software[1] that when combined with other WWW servers and the Mosaic browsing software[2,3] provides an environment which assists a software technology consultant access required documents and other information when needed and in a more productive fashion (see Figures 1, 2). Over time this document infrastructure can evolve as new information is collected and new information sources are identified.

Figure 2

Initially, as the technology consultant Web is created, local documents that are currently being used are added to the Web. This is accomplished most easily using the automatic indexing capability of HTML (HyperText Markup Language), the language used to specify the structure of documents for retrieval, and HTTP (HyperText Transfer Protocol), the protocol for transferring the document (see Figure 3). In this way, with very little startup costs, the technology consultant Web gains some useful information without having to modify the format or structure of any of the underlying documents.

Figure 3

In fact, the only step required to add these documents to the server consists of adding the following single line to an HTML document, thus pointing to a directory, DCE, on the local disk, for example:

Here are some DCE documents.

When browsing this HTML document via Mosaic, then, the user clicks on the hyperlink, *DCE documents* , and an indexed listing of the various documents is presented. Then, by clicking on one of these highlighted document names, the native document is viewed, regardless of its format (e.g., ascii, postscript, Amipro). This process can then be repeated to add other documents to the Web for any other local documents that the consultant needs to reference in the course of doing consulting work. Examples include white papers on other technologies, customer correspondence, and project plans. For this consultant, however, there are other documents in remote locations that also need to be accessed, and thus should also become apart of the Web. These include technology evaluations, technology demonstrations, education classes, sales plans, consulting proposals, and engagement summaries. The process to add an HTML reference to a remote document is also straightforward. The following lines are added to the previous HTML document, thereby producing a reference to a white paper available via anonymous ftp:

Here is an OLTP performance modeling
paper.

When browsing this HTML document, the consultant clicks on the hyperlink, *paper*, and Mosaic automatically retrieves the paper for viewing.

As another example, Usenet newsgroups which hold threaded asynchronous conversations on various technology topics can also be accessed in a similar manner:

Here is the hp.dce newsgroup.

This process, then, of incrementally adding more documents to the technology consultant Web eventually yields one that is rich in many different types of both local and remote information. All that is left is to logically organize the information in a way that makes sense and is easily navigatable to the consultant (see Figure 4).

Figure 4

The benefit to the consultant, then, is much faster and more productive document access due to a single point of reference to the voluminous documentation and disparate information sources required to adequately consult on software technologies. What previously required:

- locating the information source document
- accessing the document, potentially with cryptic commands
- viewing the document
- locating, accessing, and viewing related documents

has now been reduced to a simply mouse click. Further, all of the information can be accessed from a web of interrelated documents with a single tool. Finally, the process for the technology consultant in Dallas to access these documents, is the identical process for the technology consultant in Santa Clara, for the product marketing manager in Cupertino, as well as for the sales representative in Atlanta. This leads to a smaller learning curve, easier and greater information sharing, as well as commonality of experiences and best practices.

1.2 CommerceNet

A CommerceNet Web (see figure 5) has been created by Enterprise Information Technologies (EIT), the Stanford Center for Information Technology, and BARRNet, a Bay Area Internet service provider, and with funding from the U.S. Government. Sponsors include Hewlett-Packard Company, Apple, Bank of America, Citicorp Services, Pacific Bell, and VISA International, among others. HP, for example will put its support, education, and workstation products on CommerceNet.

Figure 5

It is an open, internet-based infrastructure on which electronic commerce may occur. Customers may shop for products from various companies and also order products they wish to purchase. Information on companies' products are presented interactively to prospective buyers using full color images, audio, and even video. A "shopping mall" metaphor is presented to the user by using the Mosaic/WWW hypermedia technology. In this way, customers enter "stores" of the various companies via hyperlinks where they can then buy goods, just look around, or visit another store of interest. The stores reside on different computers on the internet and the Mosaic/WWW hyperlinks present these to the customer as a monolithic shopping mall.

Each company, then, is responsible for creating and maintaining the appearance of the Mosaic/WWW store, just as in an regular store. CommerceNet, as is the case with a mall, acts as the advertising medium and entrypoint. Issues that need to be considered during creation of the store include customer convenience features such as placing the merchandise in easily reachable locations, and fast, friendly help and information when questions arise.

HP, for example, in creating its Mosaic/WWW store held an internal and informal electronic focus group whereby HP employees interested in its success could enter the store, use it, and then submit comments and suggestions for improvement. This took place on HP's internal network using Mosaic/WWW client software, HTML comment pages, and internal threaded group discussion software. Several suggestions were submitted relating to appearance, speed of access, and creation of pointers to other information. One issue for HP during creation of its Web pages relates to the logistics of pulling together information on products that are designed and manufactured in different parts of the world and presenting that to the customer in a concise, easy to use and understand format. The concise presentation of this information is accomplished in large part by using the Mosaic/WWW imagemap feature to show pictures of products overlayed with buttons that show the customer where to look next and that remember where the customer has already looked (see Figure 6). Using this feature, the customer links to different HTML documents (e.g., to different products in different portions of the store) by clicking on a specific part of the image map. Underlying information is organized into news, products, contacts, and services and support. Product information, for example, is then organized into computers, peripherals, components, and test and measurement.

Figure 6

The obvious benefit to the participating company on CommerceNet is opening up a completely new channel of commerce to a potentially very large customer prospect base. Further, by using the Mosaic/WWW software technology to perform this commerce, the cost of sales becomes extremely low since no direct or even indirect sales force is involved in any marketing or sales transactions. The cost of sales, instead, becomes the creation and maintainance of the CommerceNet Web. The more automation that occurs lowers the cost of sales. The Mosaic/WWW CGI (e.g., gateway) feature, for example, allows the potential automatic linkage of the placement of a customer order with the company's order entry system. Finally, since the

Mosaic/WWW software is freely available and also ubiquitous across multiple platforms, barriers to customers participating are minimized.

2 Integration into Enterprise Architecture

Many corporations are looking for a complete, functioning, and integrated information technology architecture to help support their business needs and to help evolve their business strategy. Such an architecture would be composed of technologies and products that solve different business needs and that originate from different sources.

A model of this architecture [4] (see Figure 7) is therefore important in that it helps describe the characteristics of the components and positions them relative to each other in terms of their:

- functionality
- interoperability
- completeness
- usability
- overlap, redundancy
- synergy
- interstitial transparency
- openness
- standards compliance

Figure 7

In this section, a formal evaluation and decomposition of the Mosaic and HTTPD software is performed such that its various components are mapped into this architectural model. Specifically, Mosaic and HTTPD are classified according to three of the underlying functionalities or services of the architectural model. These services are:

- Information Management Services
- Distribution Services
- Interface Services

2.1 Information Management Services

Information management services are the services that organize, store, and retrieve information on a computing system. Example functionality includes file systems, access methods, database systems, concurrency, replication, and information semantics. This module of service is most closely aligned with the builder or developer role of the overall architecture.

Trust issues here relate to data formats, authorization and authentication on data access, replication for high availability and performance, transactional integrity, and data validation.

Control issues here relate to transaction logging and rollback, backup and recovery, monitoring, auditing, and analysis, resource capacity planning, management and reconfiguration.

2.1.1 HTTPD as an Information Management Service

It is important to first note, when describing Mosaic/WWW as an information management service, that HTML, which is a subset of SGML (Standard Graphics Markup Language), is the underlying mechanism in Mosaic/WWW for providing a rich text representation of a document. This includes bold, italics, varying sized headers, bulleted lists, numbered lists, and inline images. This also includes the capability to create hyperlinks (or references) to other documents, as well as to specific locations within the current or other documents.

This service portion of Mosaic, or HTTPD, fits into this module as an information management repository (e.g., Web) for HTML, ascii, images, audios, videos, word processor documents, and other multimedia documents. Further, there is the capability via the common gateway interface (CGI) to add linkages to other program shell scripts and other information management systems that are also part of an enterprise's overall architecture. In this way, reuse of and interoperability with some other elements of the architecture is achieved.

Another characteristic of Mosaic/WWW is the integration of documents using the native formats of the documents. This is accomplished due to the way that documents are added to a repository via HTML references, instead of being completely consumed or translated into an internal format as is done in other information management systems. HTML and CGI documents are the only ones that need be added to the HTTPD repository. These could be more appropriately called "meta-documents" in that they reference or interface with the other real documents or components of the architecture. In effect, the repository here becomes a non-obtrusive binding agent between otherwise disparate elements, linking documents that have some natural relation.

A native document is transmitted from the server through some underlying protocol and to the Mosaic client for viewing by the user. At the client side, using MIME associations[5], a document becomes associated with a document viewer program which Mosaic then automatically executes once the referenced document is retrieved. This MIME association, which occurs at the client side, contributes to the document format independence present in the HTTPD information management repository.

It is worth noting that it is possible to use another model of the HTTPD information management service, by having all documents in native HTML format, either through format conversions or direct authoring in HTML. Currently, however, this is not feasible due to the current lack of HTML editors and due to the lack of rich text expression currently available in HTML. This will be shortly remedied with the expected availability of HTML editors as well as the support for HTML+ in a future version of Mosaic. The tradeoff, here, then becomes easier inline viewing vs. more costly document conversions. That is, native HTML is more usable since no other "out of line" document viewer needs to be started to view the document. However, these conversions are timely and costly and probably not feasible until document authoring programs and corporations support and embrace SGML and its derivatives as widely accepted document format standards.

In the area of trust, a myriad of functionality exists for creating authorization on specific documents and groups of documents in the repository. Specific clients or groups of clients can be granted access to these documents, or alternatively, no authorization can be performed. In the future, a Kerberos-based three party, private key authentication system is planned for Mosaic and HTTPD, when Kerberos becomes more widely deployed.

In the area of control, HTTPD supports monitoring and rudimendory auditing via activity logs that date/timestamp access to specific documents. Numerous software is available to post-process these logs into readable, rich text audit reports. An extract from one of these reports appears below:

HTTP Server General Statistics

Server: http://bsh3185.ssr.hp.com/ (NCSA)
Local date: Wed Apr 13 12:27:10 PM CDT 1994
Covers: 01/31/94 to 04/13/94 (73 days).
All dates are in local time.
Requests last 7 days: 5698
New unique hosts last 7 days: 70

Total unique hosts: 650
Number of HTML requests: 3804
Number of script requests: 1221
Number of non-HTML requests: 16296
Number of malformed requests (all dates): 79
Total number of all requests/errors: 21400
Average requests/hour: 12.4, requests/day: 297.9
Running time: 47 seconds.

HTTP Server Weekly Statistics (See <u>general statistics</u>)

Each mark (#) represents 50 requests.

Week of 01/31/94: 106 : ##
Week of 02/07/94: 1247 : ########################
Week of 02/14/94: 1636 : ################################
Week of 02/21/94: 1194 : #######################
Week of 02/28/94: 1472 : #############################
Week of 03/07/94: 2481 : ##||####
Week of 03/14/94: 1828 : ####################################
Week of 03/21/94: 2308 : ##||####
Week of 03/28/94: 2088 : ###
Week of 04/04/94: 5473 : ###||###
Week of 04/11/94: 1488 : #############################

There are no additional interstitial costs to system managers that occur when administering multiple HTTPD repositories. This is because HTTPD administration is orthogonal across all platform types.

2.1.2 Spectrum of Collaboration

When examining the various information management products that might comprise an enterprise's architecture, then, one can think of them as creating a spectrum of collaborative functionality. The different products provide different information sharing services, ranging from personal browsing tools to workgroup synchronous tools (e.g., shared applications, shared whiteboard, videoconferencing) to enterprise-wide asynchronous messaging to enterprise-wide asynchronous threaded discussions to enterprise-wide asynchronous hypermedia browsing. It can not be expected in an open environment, that one product can effectively provide all of the functionality that an enterprise requires. To be truly effective and provide maxium synergy, not only do the information management products need to interoperate, but more importantly the underlying information needs to be easily exchangable from one product to another. This can be accomplished somewhat with gateways and filters between product repositories, but can be made much more effective by having the various information management services use and understand several common well-known data formats and standards, rather than inventing or mandating their own.

It is in this area of information exchange and openness that Mosaic and HTTPD particularly excel, in that no specific formats are mandated and yet many widely implemented formats and standards are understood and accepted. These include HTML, GIF, TIF, JPEG, MPEG, AU, WAV, Postscript, and MIME. Further, there are hooks to include other formats (e.g., word processor, spreadsheet) and gateways (e.g., CGI) to interoperate with other information service products.

2.2 Distribution Services

Distribution services provide the functionality that links multiple, separate computer systems into a distributed system. Examples include network protocols, network services, invocation services, location services, security services, load balancing services, and system coordination services. This module of service is most closely aligned with the administrator or manager role of the overall architecture. Trust issues here relate to error detection, data encryption, and replication for high availability and performance.

Control issues here relate to error tracking and recovery, performance monitoring, and automatic startup, reconfiguration, and shutdown of services.

2.2.1 Distribution Services for Mosaic/WWW

HTTP acts as the primary protocol used by Mosaic/WWW to distribute documents from the HTTPD information management services to the Mosaic clients. The client/server protocol is a higher level session protocol that uses an underlying TCP/IP transport protocol to transfer HTML, ascii, multimedia, and all other documents that are referenced using the HTTP protocol. This reference is made at the time of the creation of the HTML document via an embedded hyperlink within the document. A reference within an HTML document would look like:

protocol://internet_domain_name.machine_name[:port_number]/document_pathname

where, for example, "protocol" may be HTTP, but may also be some other protocol, such as FTP. This way of referencing or designating document locations is known as a Uniform Resource Locator (URL)[7]. Several URL examples using different URL forms and protocols follow:

http://www.yoyodyne.com/pub/files/foobar.html
http://www.yoyodyne.com:1234/pub/files/foobar.html
ftp://iworks.ecn.uiowa.edu/pub/comp.hp/
file://ftp.yoyodyne.com/pub/files/foobar.txt
gopher://gopher.yoyodyne.com/
news:comp.multimedia

Note, then, that URLs decompose into different elements. The "protocol" identifies the connection mechanism between information service and information requestor client. The "internet domain name" specifies a name to internet address mapping that is resolved using whatever underlying location service that is being utilized on the client. This is typically the Internet Domain Name Service, but could also be some other location service such as X.500 or a host to address mapping file. The last step in the name resolution process is to resolve the pathname of the document on the file system of the information service. Note that this could be a file local to that system, but it could also be a file connected to that filespace using some distributed file system. At this point, then the server can communicate with the client using the protocol/address/pathname combination to distribute the document from service to client.

Two important characteristics become apparent when considering this distribution service. First, multiple common protocols are understood and utilized for distribution of documents. As such, existing documents and methodologies for access (e.g., anonymous ftp) can be reused and the access process can be automated. Further, the network infrastructure (e.g., routers, bridges, operating systems, etc.) does not need to change and processes for dissemination of documents do not need to change.

Second, since URLs uniquely describe how to access a particular document from within a workgroup, enterprise, or across enterprises, the mechanism for access becomes location independent. In this way, when a user happens to be in different location and needs to access a document, the process by which this is accomplished is identical to how it is normally done. This is also known as uniform name space, or in this problem space of information management, more aptly called uniform information space. This means that URLs provide a one-to-one mapping to to any information within any Mosaic/WWW repository. Another benefit of this is that since accesses to information are uniform, the actual URL can therefore be shared across workgroups perhaps via some other information management service. For example, a knowledge worker could use an asynchronous messaging product such as electronic mail or threaded discussions to communicate the URL of a document to a group of project peers, and thereby also communicate the existence of that document and the process for how that document is accessed.

This reusing of protocols and document dissemination processes, and also interworking and information exchange with other products in the enterprise architecture contributes to the overall cultural shifts discussed earlier, whereby sharing, collaboration, and teamwork are fostered by the combined innate synergies within the enterprise architecture.

In the area of trust, error detection and correction is currently supported, but only as a by-product of the underlying TCP/IP transport protocol. Secondly, there is no support for data replication for high availability or performance. As such, when an HTTPD information management service is unavailable for some reason, such as a hardware or software failure, the following message is displayed, stating that the referenced hyperlink is not accessible:

ERROR

Requested document (URL http://bsh3185.ssr.hp.com/) could not be accessed.

The information server either is not accessible or is refusing to serve the document to you.

Since there is no replicate of the information, there is no recourse other than trying again later, when perhaps the information may then be accessible.

Third, no data encryption capabilities are explicitly part of Mosaic/WWW. As such, in some applications, Mosaic/WWW may not be a completely trustworthy system, unless additional trust features are manually implemented. For example, a commercial Mosaic service that allows customers to electronically place orders with a credit card number may not be completely trusted enough to be used by potential customers, since the credit card number would be transmitted as plaintext across a wide-area network without encryption. It would therefore be possible to capture this information and use it fraudulently. An upcoming version of the Mosaic/WWW software will be capable of using public key encryption for transactions between client and server.

In the area of control, there is nothing inherent in the Mosaic/WWW software to track and recover from errors, monitor performance, or automatically startup or shutdown the information management service. There is a new capability in the latest version of HTTPD from CERN to reconfigure the service while it is running. Further, other performance monitoring products and startup/shutdown features of the host operating system could be used in tandem with Mosaic/WWW to accomplish this functionality.

2.3 Interface Services

Interface services provides linkages between the computing systems and the users of the systems by presenting information to and acquiring information from the users. Examples of functionality when presenting information include graphical user interfaces (GUIs), desktop management, font management, audio and video output, text markup, native language support, page description languages, and integration with utility services in the enterprise architecture (e.g., printing, faxing, mailing). This module of service is most closely aligned with the end-user role of the overall architecture.

Trust issues here relate to authenticated access, performance and performance control, hacker detection, and uniform appearance and behavior across interfaces and across multiple platforms.

Control issues relate to configuration, customization, and extensibility of the interface appearance, behavior, and content.

2.3.1 Mosaic as an Interface Service

Mosaic acts as the primary client, hypermedia, browsing interface to the WWW of HTTPD servers. It exists (1) in X-Windows format for almost all Unix workstations and servers, (2) in Microsoft Windows format for Windows-based PCs, and (3) in Macintosh format for Apple desktop systems. The Mosaic interface (see Figures 1-4) displays documents in scrollable windows comprised of text, rich text, hyperlinked text, icons, and images. Rich text is comprised of varying sized fonts and font styles. Hyperlinked text appears in a different color with an anchor underlining the hyperlink. Once the hyperlink has been referenced, it appears in yet another color, with a dashed anchor underlining the hyperlink. In addition, audio and video output is commonly associated with many of the WWW documents. When documents are retrieved, their text, rich text, icons, and images are displayed inside these scrollable windows. Associated audio, video, and other media and document types are played using an external application that is started automatically by Mosaic. The user has complete control in associating different local "players" with different media and document types that are retrieved.

Further, using Mosaic's CGI, allows it to also provide application launcher functionality. If an XWindows client is being used, this application launching can be performed either locally or remotely. A typical CGI script to launch a client/server XWindows-based company phonebook would look like:

```
#!/bin/sh
echo Content-type: text/html

 if [ -x /usr/local/bin/X11/xphone ]; then
     echo \<PRE\>
     cat <<#####
       Starting up Phonebook Utility ... please wait ...

       Note:  X11 access control will need to be disabled to complete
              this hyperlink.  If not already disabled, you will need to:

              /usr/bin/X11/xhost +
     #####
     echo \<PRE\>
     eval "/usr/local/bin/X11/xphone -display ${REMOTE_HOST}:0.0 &"
else
     echo Error:  Cannot find Phonebook Utility anywhere on this system!
 fi
```

In this case, the phonebook X interface runs locally on the same machine but in a different window from where Mosaic is running, the client part of the phonebook application runs on the same machine as the HTTPD service, and the phonebook server runs on yet another remote machine where the company phonebook resides. In this way, not only is Mosaic capable of browsing, retrieving, and displaying different document and media types, but it also provides access to diverse applications and application services. Further, these applications and documents can be linked together in a hypermedia fashion if there is some relation between them. With respect to integration with utility services of the overall architecture, Mosaic is very complete. Any Mosaic document can be saved to the local disk, printed, or mailed in a variety of formats -- either text, formatted text, HTML, or postscript. Further, there are versions of Mosaic that have a fax capability implemented. These features implement a complete, integrated utilization of common core services of the architecture. These features allow and encourage plagiarism -- not in the negative sense, but as a positive benefit that helps share a wealth of information and spreads the wealth of Mosaic/WWW authoring expertise throughout the corporation. For example, a novice information builder just getting started using Mosaic/WWW can browse and borrow from already existing HTML documents from around the enterprise that implement similar features to be implemented by saving them to the local disk as HTML, and then editing the HTML as appropriate. This can be viewed as a built-in, low-cost form of apprenticeship that applies the copy and experiment strategy and whereby the impact on the expert's time is minimized[8].

In the area of trust, Mosaic currently does not perform any authentication, performance control, or hacker detection. Mosaic does provide, however, the capability to interrupt large document (e.g., postscript, audios, videos) transfers by clicking on the familiar spinning globe of the world in the Mosaic interface. Detection of these large document transfers is provided by Mosaic informing the user of the current and total byte counts during the actual data transfer. With respect to interface consistency, there are no additional interstitial costs to users deploying Mosaic on multiple platforms, since Unix, X-Windows, Windows, and Macintosh platforms are all supported and since functionality, appearance, behavior is identical on each. Further, if non-graphical access to the WWW repositories is required (e.g., for remote dial-up access, or when using a terminal) two other clients (i.e., WWW and Lynx) are available to browse and access the textual portions of the hypermedia documents. With these browsers, inline images are omitted, rich text is converted to ordinary text, and hyperlinks are highlighted or have a number associated with them, while out-of-line viewing of other media and document types is preserved.

In the area of control, customizations to the appearance of the interface can be applied to the Mosaic documents by modifying the respective Motif widget characteristics such as fonts, colors, and sizes. Individual documents can be privately annotated with text at the user's discretion. In the future, workgroup and public annotation capabilities will be possible increasing the collaborative functionality of Mosaic. Hotlists to documents of most interest to the user can be created, thus providing a shortcut mechanism to document retrieval.

3 Conclusions

An architectural model referenced in this paper has helped describe the characteristics of the various components of Mosaic/WWW and has positioned them relative to other components and products of an enterprise's overall architecture. Specifically, the

Information Management, Distribution, and Interface Services of Mosaic/WWW have been formally analyzed for functionality, interoperability, completeness, usability, synergy, interstitial transparency, openness, and standards compliance.

It has also been shown how different products in an enterprise's architecture can provide complementary information sharing services, that span a spectrum of functionality ranging from personal browsing to workgroup synchronous to enterprise-wide asynchronous messaging to enterprise-wide asynchronous threaded discussions to enterprise-wide asynchronous hypermedia browsing. Thus, the data that these products use, manage, and provide access to needs to be easily exchangable from one to another. Here, Mosaic/WWW provides a unique openness, extensibility, and integration, not previously seen in other products. No specific proprietary formats are mandated and yet many widely implemented formats and standards are understood and accepted. For formats and products not natively understood, there are open standards-based linkages and gateways.

The complementary, integrating elements of Mosaic/WWW have been discussed in that it is shown how its components fit in with and extend an enterprise's overall architecture and co-exist with other elements of the architecture. Mosaic/WWW becomes a linking technology between otherwise disparate elements of the architecture, interoperating with related products in the architecture, piecing together documents that have a natural relation, acting as a change agent in the area of people and processes, and thereby adding to the overall cohesiveness of the architecture.

As Mosaic is deployed within an enterprise, it becomes more than an information management product to that enterprise. It helps further the enterprise's culture to more quickly shift towards a culture that is more sharing, collaborative, and team-oriented by nature. It does this by providing an example of how a true, open architecture might function with various products and technologies integrated and providing synergy with each other. It also brings information where it is needed on demand, using a hypermedia metaphor which works well for browsing related multimedia information that come from multiple sources across the workgroup, across the enterprise, and across the world.

References:

1 HTTPD, HyperText Transfer Protocol Daemon, Version 1.0, developed by theNational Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign.

2 Mosaic, Version 2.0, developed by the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign.

3 Andreesen, M.: NCSA Mosaic Technical Summary, Revision 2.1 (1993).

4 Kincl, N., Armstrong, J.: Open Distributed Computing: The Overall Architecture, NSA-93-019, Hewlett-Packard Company (1993).

5 Borenstein, N., Freed, N.: IETF Request for Comments 1341: MIME (Multipurpose Internet Mail Extensions)Mechanisms for Specifying and Describing the Format of Internet Message Bodies (1992).

6 A Beginner's Guide to HTML, NCSA (1993).

7 Berners-Lee, T.: IETF Request for Comments Draft: Uniform Resource Locators (URL), A Unifying Syntax for the Expression of Names and addresses of Objects on the Network (1993).

8 Berlin, L., Jeffries, R.: Apprenticeship Rediscovered: Understanding and Supporting an Overlooked Productivity Tool, HP Labs Technical Report HPL-92-143 (1992).