Paul Maglio and Rob Barrett

Intermediaries Personalize Information Streams

A software-based middleman transforms data between client and server for faster delivery and discovery.

Intermediaries are software programs or agents that meaningfully transform information as it flows from one computer to another. Just as brick-and-mortar retailers buy products from wholesalers and add value by providing them to consumers in attractive ways, intermediaries located in the information stream between producer and consumer can personalize raw information for individuals, devices, and situations.

In fact, many contemporary Web services are intermediaries, performing such functions as transforming news feeds and airline reservation databases into categorized and personalized Web pages. Because Web information can be easily customized, intermediaries can easily tune it for many different uses and reuse it for many different situations. Likewise, because information providers can no longer anticipate all possible uses of their information, third-party intermediary services will play an increasingly significant role in adapting and personalizing information on the Web.

Consider how news is commonly delivered on the Web today: First, a news service (such as the Associated Press or Reuters) creates a story along with its headline and category. Then, a news portal (such as Yahoo! News or Excite News) acts as an intermediary by deciding how to present the stories to the user (for example, Yahoo! News puts three top headlines on its first page, and produces separate pages for each category, such as business, technology, health). The same raw information may be processed by multiple intermediaries to present it in different ways. One intermediary may produce a page for each complete

JEAN-FRANCOIS PODEVIN

story along with links to associated information, such as stock prices and movie times. Another may add value to the information presentation by displaying headlines on a scrolling ticker or sending users news stories by email.

Many current Web services are based on intermediary processing models, but few would describe them this way. We believe intermediaries are a powerful new way to transform information flowing between computers. One result of describing Web services as intermediary services is that more productive methods for building and deploying intermediary applications can be developed as transformations are separated from the data they operate on and can be dynamically composed in novel ways. Intermediaries can transform information at the server, at the client, or anywhere in between (see Figure 1). Web Intermediaries (WBI) is an implemented framework for adding intermediary functions to the Web [1–3]. By providing a programming model for intermediaries, WBI enables the quick development and deployment of systems that add value by transforming information that flows between Web server and Web client.

Intermediaries are Everywhere

Beyond our news service example, intermediaries are pervasive across the Web. Consider travel resources such as Travelocity and Expedia. These consolidate services and information from a variety of sources, providing consumers of travel services a central point for browsing and reserving airline tickets and hotel rooms. They also tailor travel information to individuals based on user profiles and history of use. As intermediaries, these sites function much like human travel agents, scouring several sources of information, taking account of customer preferences and desires, applying rules and heuristics to

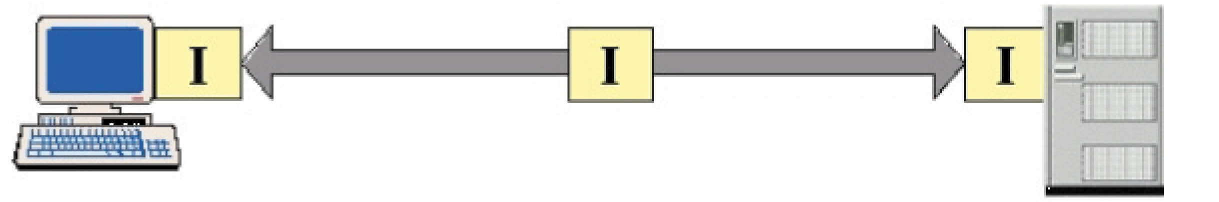


Figure 1. Intermediaries can be placed in the information stream at the client end, at the server, or between the two.

try to satisfy constraints, and engaging in a kind of dialogue with users to create appropriate plans.

As another example, AltaVista has recently added a language translation intermediary service to their search engine. It reads its input from the original Web page and performs machine translation to produce a version of the page in another language. This service adds value for readers who cannot understand the page's original language.

Intermediaries are common in many other kinds of information streams as well. For instance, email depends on intermediaries to hold messages after they have been sent and before they have been received (for example, the POP3 protocol). Other email intermediaries might provide local replicas of remote email repositories so that email can be handled while offline, automatic summarization of long email messages, intelligent email routing to correct email addressing errors, and services that log and index email for later retrieval.

Table 1. Intermediaries perform five basic functions that can be distinguished by the kind of information they take into account when performing their transformations.

Function	Purpose	Additional Information
customize	tailor for particular users	user and user's environment
filter	remove information	information from world
annotate	add information	information from world
aggregate	combine from multiple sources	other information streams
transcode	change to different form	desired output format
cache	store for later use	timeliness of information

Intermediaries can also be used directly by consumers for removing links to sites inappropriate for children (for example, Edmark's KidDesk Internet Safe), enhancing Internet privacy through anonymization (for example, Anonymizer), suggesting interesting links to follow (for example, Alexa), adding links based on a user's history of Web use [3], and so on. As the quantity and variety of online information increases along with the number and variety of users, devices, and situations, intermediaries become valuable tools for both consumers and producers of information.

Fundamentals of Intermediaries

Intermediaries work in several different ways depending on how they are implemented and what functions they perform.

Implementation issues. Intermediaries can be classified according to where they run and who controls them. They can run on client machines under the control of an individual user. They can run on server machines as part of the information provider's system. They can also run on the network's edge, such as at the ISP or corporate firewall as a service provided to the end user (see Figure 1). For example, AltaVista provides a language translation service for Web pages at its Web site, but one could imagine translation being performed at a corporate or university firewall (for instance, so all English-language pages viewed at a university in France appear in French) or even on the end-user's machine. Of course, individual sites can provide translations of their own content as well by inserting a language-translation intermediary internal to their sites.

Intermediary functions. Intermediaries can also be classified according to their function. Intermediaries process information by *customizing, filtering, annotating, transcoding, aggregating,* and *caching* (see Table 1). These functions take into account different kinds of information when doing their processing. For example, customization takes into account information about the user or the user's environment when doing its work. For instance, Amazon.com adds personalized book recommendations to Web pages based on the user's previous purchases. Transcoding takes into account information about the data's input and desired output format, for instance, transcoding hypertext markup language (HTML) documents authored for the Web into wireless markup language (WML) documents displayed on a wireless phone. Aggregation takes into account a second data stream, for instance, merging results from several search engines into a single page of search results. Caching

Figure 2. Intermediary functions can be implemented with three basic operations. Generation produces raw information for further processing. Editing modifies the raw information. Monitoring observes information, tracking it as it is processed.

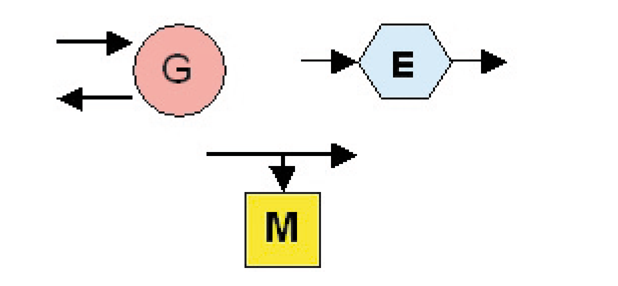
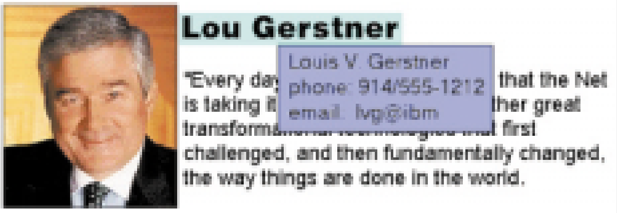


Figure 3. An example intermediary application. This intermediary edits Web pages to add links for people's names. It also generates information from the corporate directory about the person. Finally, it monitors the names the user has seen so they can be added to an address book.



Lou Gerstner

Louis V. Gerstner
phone: 914/555-1212
email: lvg@ibm

"Every day that the Net
is taking it ther great
transforma first
challenged, and then fundamentally changed,
the way things are done in the world.

takes into account when data was last stored or last changed so that data might be retrieved from a local repository. These high-level intermediary functions can be implemented by combining just three lower-level operations: *data generation, data editing,* and *data monitoring* (see Figure 2).

Generation produces the source form of the information and passes it along to the intermediary for processing. Generation may retrieve the raw information from a Web server or local cache, or it may compute it dynamically. Conventional Web servers, along with their programming interfaces (for example, Common Gateway Interface (CGI) scripts and Java Servlets) are examples of data generation. Clearly, any function can be built as a single, complex data generator. But intermediary-based systems are more flexible and modular because they separate information generation from subsequent transformations.

After it has been generated, an intermediary may edit the raw data. It may be transcoded into a form compatible with the user's viewing device, for example, the Palm Pilot or Wireless Application Protocol (WAP) phone. Or as another example, it may personalize the information for the particular user. Similarly, as more data is structured in eXtensible Markup Language (XML), intermediaries will edit the data with eXtensible Style Sheets (XSL) to transform it appropriately for the particular situation.

Data monitoring allows the intermediary to track information as it passes through, either in raw form or after processing. Monitoring is commonly used for caching Web resources for faster delivery, logging usage patterns, billing for the use of raw information, and building user models for Web browsing activities.

Figure 3 illustrates the way an intermediary application can be built from these three basic components. In this case, Web pages are edited to add links to people's names from a corporate directory. When one of these links is clicked, the intermediary generates a page with that person's directory information. Monitoring is used to keep a record of the people this user has read about on the Web.

WBI Functions

As mentioned, WBI (pronounced, WEB-ee) is a development kit and server software for adding intermediary functions to the Web. More precisely, WBI is a programmable hypertext transfer protocol (HTTP) proxy server designed for easy development and deployment of intermediary applications [1, 2].

With WBI, intermediary applications or plugins are constructed from the three basic operations: monitors, editors, and generators (known collectively as MEGs). Plugins are made up of multiple MEGs that work together. Multiple plugins can be installed in WBI to combine applications, such as transcoding and caching. WBI then dynamically routes Web transactions through the MEGs by matching the information about the HTTP transactions against Boolean rules associated with the MEGs. For example, a monitor might be configured to track only information of type "text/html" that comes from the "yahoo.com" domain.

Plugins are defined by their MEGs and the rules that control when the MEGs are involved in a transaction. For instance, a caching proxy is easy to implement as a WBI plugin that monitors HTTP transactions to store documents, and generates documents from the local store when they are available (see Figure 4). Document transcoding is also straightforward to implement as a WBI plugin.[1] Recall that transcoding is the process of converting a document formatted one way into a document formatted a different way. For example, it might be convenient to convert HTML documents to WML or convert images containing millions of colors to images contain-

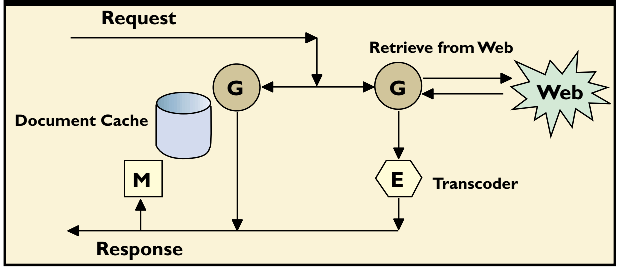
[1] In fact, IBM's Websphere Transcoding Publisher product is based on WBI.

BECAUSE INFORMATION PROVIDERS CAN NO LONGER ANTICIPATE ALL POSSIBLE USES OF THEIR INFORMATION, THIRD-PARTY INTERMEDIARY SERVICES WILL PLAY AN INCREASINGLY SIGNIFICANT ROLE IN ADAPTING AND PERSONALIZING INFORMATION ON THE WEB.

ing four bits of grayscale so that wireless phones can display them. To convert HTML to WML, an editor would trigger on a rule that matches either the content-type of the response ("text/html") or the file extension of the URL ("*.html"). This editor can then use whatever software is necessary to effect the conversion from HTML to WML. WBI handles all the plumbing, allowing the application writer to concentrate on application details rather than on HTTP (see also [5]).

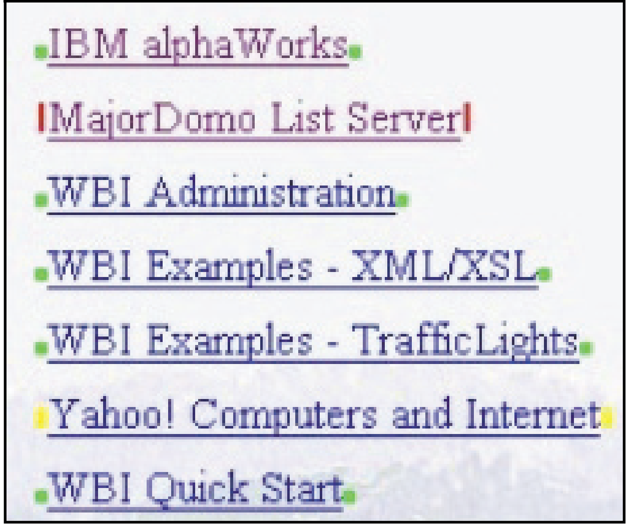
Obviously, the plugins for transcoding and caching can be chained one after the other so that transcoded documents are automatically cached, eliminating expensive format conversions already completed (see Figure 4).

Figure 4. Caching can be easily implemented as a WBI plugin. A request for a document is generated by a Web browser and flows either to a generator that retrieves it from the cache or retrieves it from the Web. If the generator retrieves the document from the Web, transformations, such as transcoding from one document type to another, may then be done by an editor. On its way back to the browser, the response is monitored and possibly stored in the cache.



We have developed many WBI plugins that personalize and customize information for individual users or groups of users [3]. For instance, the Personal History plugin uses a monitor to record the sequence of pages visited by a user along with the content of each page. The user's personal history is accessible through keyword queries or path browsing. Keyword queries retrieve a list of pages the user viewed sometime in the past containing the given keywords. Path browsing allows a user to view paths taken through a particular page. If the user knows only that some relevant page was seen shortly after going through the IBM home page, then by browsing only the paths taken through the IBM home page, he or she is likely to quickly find the sought-after page.

Figure 5. The WBI Traffic Lights plugin annotates links with colored dots that indicate the network delay for reaching the linked page.



🟩IBM alphaWorks🟩

🟥MajorDomo List Server🟥

🟩WBI Administration🟩

🟩WBI Examples - XML/XSL🟩

🟩WBI Examples - TrafficLights🟩

🟨Yahoo! Computers and Internet🟨

🟩WBI Quick Start🟩

Using the contents of pages monitored and stored by the Personal History plugin, WBI can customize Web pages by adding links to pages frequently visited by a certain user. Because Web users rely on routines and standard behaviors to access information [3, 7], WBI's Short Cuts plugin adds links to pages a user routinely visits. This plugin uses a document editor to add links when there are pages in the database that the user habitually visits within some radius (say, five clicks) from the current page.

Another way WBI can tailor Web pages is by adding extra information about the hyperlinks on a page, such as the speed of downloading information from a particular site. WBI's Traffic Lights plugin uses a document editor to add green, yellow, and red dots next to hyperlinks to inform the user that responses from the server are fast, slower, or slowest (see [3, 4], Figure 5).

Other sorts of annotations to Web pages are also possible. For instance, the Dictionary plugin turns ordinary words on Web pages into hyperlinks that point to their definitions [6]. In this case, a document editor analyzes the text of Web pages for words or phrases whose definitions are contained in some Web-based dictionary. If a word or phrase definition is found, it is transformed from ordinary text to a hyperlink that points to the definition. In this way, pages containing highly technical medical or legal information can be made accessible to nonspecialists.

As sketched in the example shown in Figure 3, WBI's Person Directory plugin provides another example of annotating Web pages with externally available information [6]. This plugin uses a document editor to find names contained in a phone or corporate directory (for example, using a Lightweight Directory Access Protocol—LDAP—server), and insert phone numbers or other personal information into the document. This can be done at a corporate firewall, enabling Web pages that flow through the firewall to be automatically customized with corporate directory information.

Conclusion

Intermediaries are programs that operate on information flowing between its producer and its consumer. As we have shown, intermediaries already play a fundamental part in information production and transformation on the Web (for example, portals, proxies, transcoders). The notion of intermediaries can be generalized to incorporate processes that produce personalized information and much more. Intermediaries can be analyzed in terms of MEGs and in terms of their positioning and function. This analysis simplifies the design, development, and deployment of intermediary-based processing. WBI is a framework for developing and deploying such intermediaries on the Web. We believe intermediaries provide an important and useful new model for personalizing Web-based information.

References

1. Barrett, R. and Maglio, P.P. Intermediaries: An approach to manipulating information streams. *IBM Syst. J. 38* (1999).
2. Barrett, R. and Maglio, P.P. Intermediaries: New places for producing and manipulating Web content. *Computer Networks and ISDN Systems 30* (1998); 509–518.
3. Barrett, R, Maglio, P.P., and Kellem, D.C. How to personalize the Web. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI '97).* ACM Press, New York, NY.
4. Campbell, C. S. and Maglio, P.P. Facilitating navigation in information spaces: Road signs on the World Wide Web. *Intern. J. Human-Computer Studies 50* (1999); 309–327.
5. Ihde, S., Maglio, P.P., Meyer, J. and Barrett R. Intermediary-based transcoding framework. In *Poster Proceedings of the Ninth International World Wide Web Conference.* (2000)
6. Maglio, P. P. and Farrell S. LiveInfo: Adapting Web experience by customization and annotation. To appear in *Proceedings of the First International Conference on Adaptive Hypertext and Hypermedia (AH 2000).* Springer–Verlag, Heidelburg.
7. Maglio, P. P. and Barrett, R. How to build modeling agents to support Web searchers. In *Proceedings of the Sixth International Conference on User Modeling.* (1997). Springer Wien, New York, NY.

Paul Maglio (pmaglio@almaden.ibm.com) and **Rob Barrett** (barrett@almaden.ibm.com) are research staff members at IBM Almaden Research Center in San Jose, CA.

URLs for Further Information

Alexa—www.alexa.com
AltaVista—www.altavista.com
AltaVista World—world.altavista.com
Amazon—www.amazon.com
Anonymizer—www.anonymizer.com
Associated Press—wire.ap.org
Common Gateway Interface (CGI)—
hoohoo.ncsa.uiuc.edu/cgi
Edmark KidDesk Internet Safe—
www.edmark.com/prod/kdis
Excite News—news.excite.com
Expedia—expedia.msn.com
eXtensible Markup Language (XML)—
www.w3c.org/XML
eXtensible Style Language (XSL)—
www.w3c.org/Style/XSL
Hypertext Transfer Protocol (HTTP)—
www.w3c.org/Protocols
IBM WebSphere Transcoding Publisher—
www.ibm.com/software/transcoding
Java Servlets—
www.javasoft.com/products/serlet/index.html
Lightweight Directory Access Protocol—
www.faqs.org/rfcs/rfc1777.html
Palm Inc.—www.palm.com
POP3—www.faqs.org/rfcs/rfc1939.html
Reuters—www.reuters.com/news
Travelocity—www.travelocity.com
Wireless Access Protocol (WAP)—
www.wapforum.org
Yahoo! News—news.yahoo.com