

Web-based Access to an Online Atlas of Anatomy: The Digital Anatomist Common Gateway Interface

S.W. Bradley, C. Rosse, J.F. Brinkley

Digital Anatomist Program, Dept. Biological Structure and
Integrated Academic Information Management Systems (IAIMS) Program
University of Washington, Seattle, WA 98195

ABSTRACT

A World Wide Web Common Gateway Interface package is described for accessing existing on-line interactive atlases of anatomy. The Web interface accesses the same 2-D and 3-D images of human neuroanatomy, knee anatomy and thoracic viscera that are currently accessed by a custom interactive atlas in distance learning courses. Although the Web interface is too slow to replace the existing atlas, it provides a parallel access path that has much broader potential for development of a distributed distance learning network in anatomy. By maintaining both access methods to the same information sources we continue to satisfy the fast interactivity needs for our local courses, while at the same time providing a migration path to the Web as the capabilities of Web browsers evolve.

INTRODUCTION

The recent explosion in use of the World Wide Web [1] has led to several applications for distance learning in medicine, including radiology, blood morphology, and many others. The University of Washington HealthLinks Web site (<http://www.hslib.washington.edu>) has pointers to some of these applications. However, for highly interactive applications current Web browsers are very slow because the connection to the server must be re-established for each interaction, and because all processing must be done at the server.

In a separate presentation we describe a non-Web based client-server framework for the Digital Anatomist interactive atlas of anatomy [2]. In this system the client-server connection is maintained throughout the session, and all information related to a single image is downloaded in a single chunk, after which the client performs the computations necessary for interaction with the image.

This mechanism has proven to be a viable approach to distance learning in our medical school, and is used extensively in several neuroanatomy and gross anatomy classes, both at the University of Washington and at remote sites on the Internet.

The success of our online anatomy atlas has led us to consider ways to make it more widely usable as a building block for University of Washington IAIMS (Integrated Academic Information Management Systems) [3] educational tools. However, the custom framework is a one-of-a-kind system that runs only on the Macintosh computer. In addition, it is not directly compatible with the newer Web based learning programs, nor is the information easily distributed among many sites as it is on the Web. Thus, there is a tradeoff between the increased interactivity of the custom approach versus the wider applicability of the Web-based approach, and we have begun a series of experiments to evaluate these tradeoffs.

In this paper, we describe one such experiment, a Web-based interface to the same images that are accessed by the custom client. Although the Web client is slower than the custom client, it is inherently portable across multiple platforms, and has the potential to link distributed sources of information. By providing both a Web-based and a custom interface to the same data we ensure that the atlas remains fast enough for use in our local courses, while providing a pathway for much wider availability as Web-based browsers evolve.

Web-Based Extensions to the Digital Anatomist Framework

The Digital Anatomist Interactive atlas is an image-based drill and practice program. Images depicting 2-D or 3-D views of anatomic objects are downloaded to a local client from a server. Once the image is available the student clicks the mouse

on various regions and the atlas displays the name of the structure, along with other attributes such as definition. The structure name is also linked to available Quicktime movies. The interactive atlas is one component of the Digital Anatomist distributed framework [4], a portion of which is shown in figure 1. The components above the network line are clients, whereas the components below the line are servers and information resources. The original custom client is the Macintosh Interactive Atlas; the original custom servers are the structural database server, the structural knowledge base server, the log server and the dispatch server.

The interactive atlas client accesses images, stored as files in the database, by means of the structural database server, which acts somewhat like an ftp server. The Mac client can also access anatomic terminology and relationships stored in our knowledge base, via the structural knowledge server.

Each database request from the client retrieves two files: a PICT file showing the anatomic region of interest, and an associated *frame* file. The frame file contains the name of the PICT file, as well as a list of polygonal regions denoting structures of interest on the image. When the user clicks on the image the client determines which polygon has been indicated, and displays the name of the associated structure. The name also acts as an index for retrieving associated information from the knowledge base. A second kind of frame file associates commands with the polygonal regions; the primary command is "OPEN-FRAME", for retrieving a new frame file.

The bottom-left side of figure 1 shows the World Wide Web extensions to the existing Digital Anatomist servers: a standard Web Server (i.e.: NCSA httpd 1.3), and the Digital Anatomist CGI (Common Gateway Interface) Package. Anatomy atlas images are retrieved from the database by means of the CGI package, which is in turn accessed by the Web server. The images are displayed on a standard Web browser such as Mosaic or Netscape. Unlike the custom framework, the Web-based interface to the databases is structured such that almost all of the processing is done inside the server's CGI package, and only minimal amount of data is sent to the client at any one time.

Figure 2 shows a typical page of the Digital Anatomist Interactive Atlas on the World Wide Web. Once the image has been retrieved the stu-

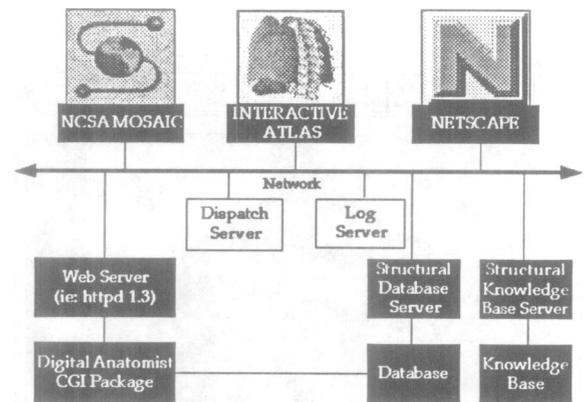


Figure 1: Web extensions to the Digital Anatomist framework

dent clicks on a region of the image, causing the name of the structure, in this case *descending thoracic aorta*, to appear at the top of the image. If the "Show Outlines" button is selected the screen shows the same image with all structures outlined, (figure 3).

The Digital Anatomist Common Gateway Interface

The heart of the Web extension is the CGI package, which consists of four main CGI programs and a server cache directory for saving converted images. The CGI programs access files in the database and knowledge base directly, bypassing the custom servers.

Figure 4 is an expansion of the CGI box in figure 1, and shows how these programs work together. All communication between these programs and the client is facilitated through the Web server. The arrows denote information flow from program to program. The following sections describe these programs in more detail.

The Controller: Imageform

Every atlas page on the client is actually an HTML *form* [1], sent by the Web server. There are several specific items each form contains. Every user interface control, such as the "Show All Outlines" checkbox, is one of these items. The actual anatomical image shown on the page is also a form item. In addition, there may be one or more invisible text fields for use by the server. Each form item has a name and a value. For instance, the

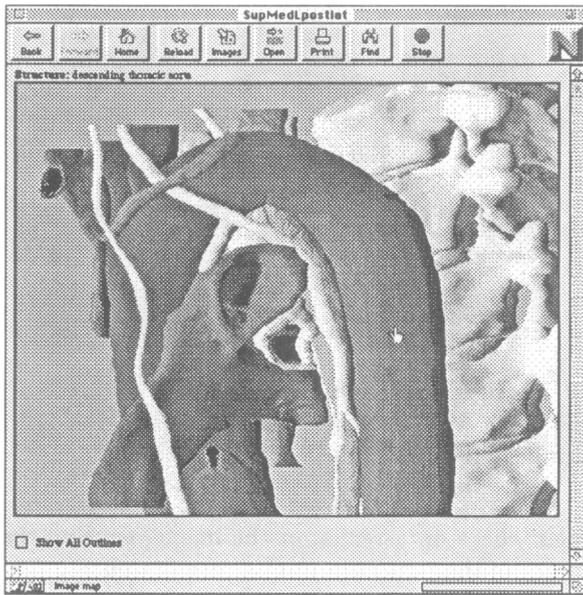


Figure 2: A typical Atlas page on the World Wide Web.

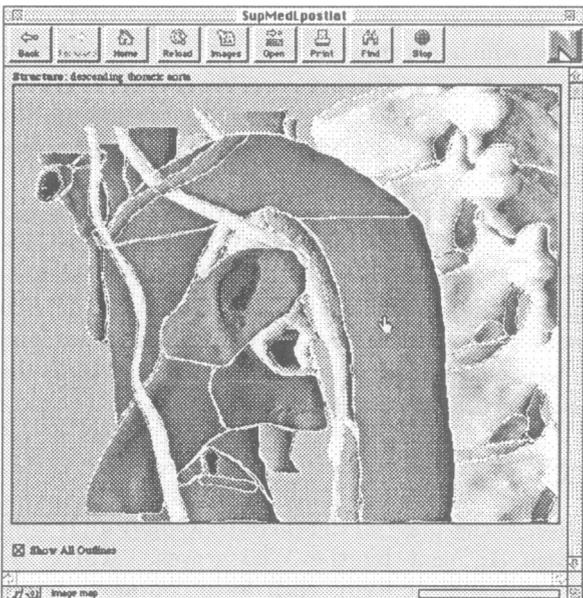


Figure 3: A typical Atlas page with the structural outlines turned on.

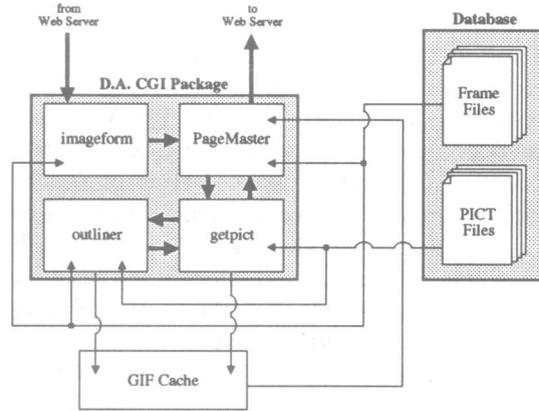


Figure 4: Architecture of the Digital Anatomist Common Gateway Interface

checkbox named "outlines" may have a value of "on" or "off." One of the invisible fields, "ffpath", has its value set to a string denoting the path to the frame file containing the named structure information.

The image acts as two form items, "image.x" and "image.y", which record the value of the X and Y coordinates of the mouse click, respectively. All forms must have an action that causes them to be sent to the server. Many forms have a "Submit" button. The Digital Anatomist forms use the click inside of the image as a "Submit" button. When the user clicks inside the image, the "image.x" and "image.y" items are assigned their appropriate values, and all of the form items, names and values, are sent to the server for interpretation.

Imageform is the program responsible for reading the form data from the Web client. It does not process the value of the form's user interface controls, but passes them on to the other programs that do. Imageform uses the path stored in the "ffpath" invisible field to read the structural outline coordinates and perform a point-in-polygon comparison with the mouse-click coordinates.

Once Imageform determines which region the click was in, it looks at the name of the region to determine whether it is a control button that links to another frame with a different image, or an anatomical structure that needs to be identified. If it is an anatomical structure, imageform transfers control to PageMaster, passing it the values of all the form items (one of which is the path to the current frame file) and the name of the anatomical structure.

If the region is determined to be a control button, imageform reads the path name of the new frame to be linked to. It then substitutes the new frame file's path name for the form item that was specifying the current frame file. Imageform then calls PageMaster with no structure name parameter, and all of the form items.

If the mouse-click is not found to intersect any of the named regions, Imageform transfers to PageMaster, passing the structure name "unlabeled area" and all of the current form items, without changing frames.

Screen Design: PageMaster

PageMaster is the program responsible for processing most of the form data, and designing the next form page that the client will see. First, it names the client window after the frame file. Then, at the top of the page, it displays the name of the last structure clicked, as passed from Imageform. From the directive in the frame file, PageMaster finds out the relative path to the image file. It then determines, from a form item, whether the outlines should be shown. PageMaster then looks into the server's cache of GIF files, to see if the necessary image file exists in GIF format. If the GIF was found, PageMaster outputs a pointer directly to that file, for the Web browser to display, probably from its own local memory cache. Otherwise, PageMaster outputs an IMG SRC path to another CGI program: Getpict. This URL directs the Web client to retrieve the GIF image that getpict redirects it to. Once the IMG SRC path has been resolved PageMaster outputs the "Show All Outlines" button, with its previous state remembered, just below the image. The last thing PageMaster outputs is the invisible form item containing the path of the frame file for this image, as it was passed from Imageform.

Image File Handler: Getpict

Since all of the image files are stored as Macintosh PICT files, and the standard image file format for the World Wide Web is GIF, image file conversion must be done. To keep everything compatible with the original Macintosh version of the Digital Anatomist, our current approach is to not touch the original PICT files, but to convert them to GIF as they are needed at run time. This conversion process can take an unacceptable length of time, especially if the process is run on a slow computer. This is the reason for the server-side GIF cache.

This cache is a directory, with the hierarchy duplicated from the original PICT directory. When a PICT file is converted to GIF format, it is saved in the cache directory, exactly where it would appear in the PICT directory. Having a separate cache directory gives us the ability to monitor the cache size, and remove the least-recently-used image file if the cache gets too big.

This image conversion is the main function of Getpict. As input, Getpict takes the relative path to a PICT file and a flag denoting whether outlines should be on or not. If no outlines are needed, the appropriate path prefix is prepended, then the PICT file is converted to GIF with Unix *netpbm* utilities, and saved in the cache directory. Getpict then exits, outputting redirection commands to the Web client to get the new GIF file. If outlines are needed Getpict transfers control to Outliner, passing the full paths to both the PICT file and the frame file, and then echo's Outliner's output back to the Web client via PageMaster.

Outliner

Figure 3 shows an atlas page containing an image that is the result of Outliner. Outliner is the program that takes care of drawing the structural outlines on top of the image file. It is passed the path to a PICT file and the path to its associated frame file. First, Outliner converts the PICT file to a standard format called *ppm*. Once this has been done, Outliner can easily read information about the image, like height, width, and number of colors. After this information has been read, Outliner reads the coordinates from the frame file, storing them in order in memory. It must also compute the coordinates of the pixels that compose the line segment connecting two adjacent region boundary coordinates. Then Outliner modifies the ppm file by replacing all of the pixels that correspond to the region boundaries, with a pixel of the color specified by the frame file. This is all done in memory. After this process has been completed, the content of memory is converted to GIF format, and saved in the cache directory under a name similar to the original that denotes that it has outlines. Outliner then outputs commands to redirect the Web client to retrieve the new GIF file.

RESULTS AND DISCUSSION

The Digital Anatomist CGI Package provides access to the same image data that is available to the custom client, currently 423 image frames, of

which 290 are neuroanatomy and 133 are thoracic viscera [2]. As the anatomists update the database the changes are immediately visible on the Web via the CGI package. Each image obtained via the Web is associated with the same structural regions as in the custom version, so all structures that are outlined can be seen in a Web image.

Qualitative comparison between the Web based atlas and the custom atlas suggests that, although the Web atlas is slower to return a structure name than the custom atlas, it is not unacceptably slow, especially when run on fast clients and servers. However, since all mouse clicks must be processed on the server, we expect that as network distance increases the Web based atlas will be unacceptably slow when compared to the custom atlas. This expectation, along with the lack of overlays delineating outlined structures, is the main reason we will continue to support the custom atlas.

In addition to the slower speed and lack of overlays there are other features of the custom atlas that are not yet supported in the Web version, including quiz mode, animations, and access to the knowledge base. These features should be relatively easy to incorporate into the CGI package given the availability of the forms interface and our Lisp-based knowledge server organization [4]. We have already demonstrated a text entry Web interface to the knowledge base. All that needs to be done now is to replace the manual text entry with a structure name returned from Imageform. With the help of the anatomic extensions to the UMLS (Unified Medical Language System) [5] currently under development, this will be very powerful for looking up synonyms and root words of structure names that have been retrieved.

The current experiment has shown us the feasibility of constructing a Web based interface to an interactive atlas of anatomy. Although the interface is too slow to replace our current atlas, it provides many exciting opportunities for expansion that could not be easily accomplished with the custom atlas. Among these are automatic platform independence, easy creation of user interface devices, dynamic generation of atlas contents by back-end knowledge bases and graphics server programs (which are already available in our framework and elsewhere[4]), and sharing of content among multiple sites.

The latter idea has great appeal as the basis for a collaborative network for distance learning in structural biology, of the kind envisioned in our IAIMS project [3]. For instance, one school may

have a premier neuroanatomy program while the other may have more strength in its musculoskeletal program. With the use of this type of framework, any group of institutions or departments may collaborate to create a distributed anatomy distance learning system. As some of the current Web limitations are resolved this approach to medical education will become a very useful and powerful learning tool.

Acknowledgments

This work was funded by National Library of Medicine Grants LM05620 and LM04925, and Human Brain Project Grant DC/LM02310. The authors would like to acknowledge the support of the following members of the IAIMS Educational Group at the University of Washington: Jim Barrett, Cliff Solomon, John Bolles, Debbie Ketchell, and Sherrilynne Fuller. The Digital Anatomist on the World Wide Web can be found at the URL: <http://www1.biostr.washington.edu/DigitalAnatomist.html>.

References

- [1] R.J. Vetter, C. Spell, and Ward. C. Mosaic and the World-Wide Web. *Computer*, 27(10):49–57, 1994.
- [2] J.F. Brinkley, K.R. Eno, J.W. Sundsten, D.M. Conley, and C. Rosse. A distributed framework for distance learning in anatomy: The Digital Anatomist Interactive Atlas. 19th Symposium on Computer Applications in Medical Care. In Press, 1995.
- [3] S. Fuller. Creating the integrated information infrastructure for the 21st century at the University of Washington Warren G. Magnuson Health Sciences Center. In *Proceedings, 17th Annual Symposium on Computer Applications in Medical Care*, pages 529–533, Washington, D.C., 1993.
- [4] J.F. Brinkley, K. Eno, and J.W. Sundsten. Knowledge-based client-server approach to structural information retrieval: the Digital Anatomist Browser. *Computer Methods and Programs in Biomedicine*, 40:131–145, 1993.
- [5] C. Rosse, M. Ben Said, K.R. Eno, and J.F. Brinkley. Enhancements of anatomical information in UMLS knowledge sources. 19th Symposium on Computer Applications in Medical Care. In Press, 1995.