Public Access to Anatomic Images

George R. Thoma
Lister Hill National Center for Biomedical Communications
National Library of Medicine, Bethesda, Maryland

Public Access to Anatomic Images

1 Introduction

Ever since the Internet and the World Wide Web became ubiquitous, the lay public, as much as the scientific community, has taken for granted easy and reliable access to information of all kinds. This expectation continues to be met by commercial database providers, and increasingly by national institutions such as the Library of Congress and the U.S. National Library of Medicine (NLM). Expressly stated as a goal in NLM's long range plan formulated in 2000 is to "Encourage use of high quality information by health professionals and *the public*." In addition, among the high priority new initiatives identified by NLM's Board of Regents in 2001 is Health Information for the Public.[1] In implementing this vision, NLM has created such services as MedlinePlus®, ClinicalTrials.gov and NIHSeniorHealth, all primarily for the public rather than for its more traditional constituencies, the biomedical clinical and research communities[2].

It is in this same spirit that the AnatQuest project discussed here has been organized. Our focus is to provide the lay public images of the human anatomy, specifically high resolution color cryosections from NLM's Visible Human Project,[3,4] and 3D images of anatomic structures created from these cryosections. By enabling public access to these images, we contribute to the increasingly important mission of the NLM to "universalize" access to biomedical information.

This effort, however, requires investigation into advanced techniques and technologies, e.g., multi-tier system architectures, database design, design of suitable image viewers, image compression and others. Research in these areas inform our overall goal which is to explore and implement new and visually compelling ways to bring anatomic images from the Visible Human (VH) dataset to the general public. Specific objectives are to:

a. Develop a system, also called AnatQuest, to let users query the VH image database via visual and textual navigation, and retrieve and display high resolution images, assuming minimal bandwidth requirements.

b. Investigate techniques to extend text-based information services for the lay public (e.g., MedlinePlus) to include access to anatomic images.

c. Explore options to segment and label the *high resolution* VH cross-section images beyond the thorax, the only region currently segmented and labeled, to enable the creation of images of 3D structures from all anatomic regions of the Visible Male and Female datasets.

Since its availability, the VH image set has inspired many projects and applications worldwide. Of these many applications there are a few that meet three conditions we consider important for public access: Internet accessibility via browsers; the provision of at least some labels for anatomic structures in each cryosection slice; and acceptable user interaction in a low bandwidth environment. One is the Workshop Anatomy for the

Internet (WAI) from the Johannes Gutenberg University in Mainz, Germany. WAI contains both labeled and unlabeled cryosections, correlated CT and MRI images (also part of the VH image set), animations, and a vocabulary of gross anatomy. Another is the Visible Human Web Server from Ecole Polytechnique Federale de Lausanne in Switzerland. This application offers services for extracting labeled slices, surfaces and animations; real-time navigation through the body; constructing 3D anatomic structures; and creating teaching modules. A third application, Net Anatomy, is a multimodal teaching tool for anatomy from Scholar Educational Systems, Inc.

While these are effective means for the public to view and use VH images, none of them offer the *high resolution* version of the image set. The high resolution set was created by digitizing 70mm film frames captured during cryosectioning to a resolution of 4K x 6K, and cropped to 4K x 2.7K. The “standard” resolution set was directly captured by a charge coupled device (CCD) camera at 2048 x 1216 pixels. The latter is universally used by application developers, but we use the high resolution images with the expectation that fine, subtle structures possibly missed in the CCD-captured images would be evident in these (with four times the number of pixels).

This chapter is organized as follows. In section 2 we briefly describe earlier inhouse projects as background. In section 3, we present the design tradeoffs that underlie the development of the online AnatQuest system, and the design considerations in creating a kiosk version of AnatQuest for onsite exhibits. In section 4, we describe ongoing work that should provide the lay public greater access to anatomic images. All work described here has been done at the Communications Engineering Branch of the Lister Hill National Center for Biomedical Communications, an R&D division of the NLM.

2 Background

2.1 Previous work

Previous inhouse work undergirding the AnatQuest project includes the VHSystems and 3DSystems projects. First, the VHSystems project established a means for the bulk transfer of Visible Human data, mainly the color cross-sections, over the Internet. Both high resolution images as well as the original CCD-captured files were disseminated by an FTP server that continues to deliver an increasing amount of data, as seen in Figure 2.1. Demand for these images is widespread, as evident from the geographic and domain distribution of the recipients (Figure 2.2).

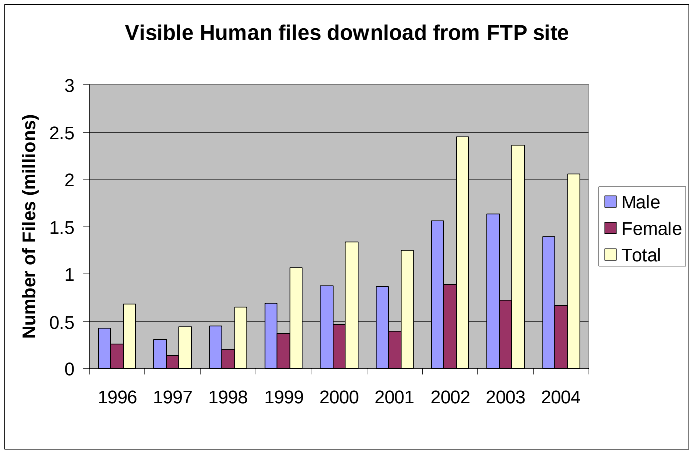

Figure 2.1.

Educational (.edu)	1,942,254	Brazil	51,709	South Africa	11,660
Network (.net)	770,223	Belgium	47,288	Switzerland	10,311
Commercial (.com)	648,003	Italy	45,716	Slovenia	9,021
Japan	623,679	India	45,150	Romania	6,529
Germany	365,791	Hong Kong	41,262	United States (.us)	5,831
Canada	254,616	Mexico	35,542	Colombia	5,514
United Kingdom	228,301	Portugal	32,243	Greece	5,340
Korea (South)	103,289	Sweden	26,366	Malaysia	4,069
Czech Republic	100,405	Spain	25,948	Norway	3,872
Taiwan	91,247	Israel	23,447	Austria	3,739
Netherlands	77,967	Finland	22,607	Denmark	2,710
France	75,735	Ireland	16,384	Military (.mil)	646
US Government (.gov)	75,352	Venezuela	14,787	Thailand	140
Singapore	61,360	Chile	14,188	Egypt	93
Poland	60,750	Hungary	14,011	New Zealand	13
China	53,759	Organization (.org)	13,793		
Australia	52,984	Iceland	13,589		

Figure 2.2. Domain distribution of Visible Human files downloaded

The main goals of the second project, 3DSystems, were to create a database out of the VH dataset, and to create suitable data and file structures so that access would be provided to the original images as well as to derivative products. A non-proprietary file format, VHI[5], was created to accommodate a wide range of image types: color cross-sections, MRI, CT, segment masks (identifying contours and labels on anatomic structures in the cross-sectional images), and volume of interest (VOI) image stacks. The targeted uses for these images were: rendered images for education, e.g., for curriculum development; the segment masks and VOI stacks for product development, e.g., to create surface and volume rendered images of organs; and the color cross-sections for research into the design of algorithms for segmentation, registration and rendering.

To achieve the goals of the 3Dsystems project, a prototype image management system, AnatLine,[6] was developed to import and store the images, and to retrieve and export them. As a means for validating the design and implementation of the database and image management system, example 3D rendered images and VOI stacks were required. To create these, the thorax region of the male (consisting of 411 slices out of the total 1,878) was segmented and labeled, and a tool VHVis[7] was developed to use these labeled segments to surface-render selected anatomic objects.

Also developed were tools needed to use AnatLine: VHParser and VHDisplay. The first is for unpacking the VHI data files into its individual components (cross-section images, byte masks, coordinate and label tables, etc.). VHDisplay is for displaying both cross-sectional and rendered images. Also, VHDisplay is augmented to audibly voice the names of anatomic structures as the images are displayed. These tools may be downloaded from the AnatQuest Website: anatquest.nlm.nih.gov/anatline/.

2.2 Prologue: Database design

Since the AnatQuest system inherits the image and data repository developed for the AnatLine system as part of the earlier 3Dsystems project, here we present the principal design considerations leading to AnatLine. We focus on: the conceptual data model, the choice of the object oriented framework for the design of the database, and the selection of a specific object oriented DBMS. Each of these is discussed next.

a. Data
In addition to the raw color cryosection images and the CT and MRI images from the data collection process, the data to be stored also includes annotated rendered images, as well as segmented and labeled images from the Visible Male's thorax region. Concepts and relations in the Metathesaurus of NLM's Unified Medical Language System (UMLS) were used to assign labels to the segments, and anatomical relationships among the segmented structures were identified. The x-y-z coordinates of the segmented structures were used to compute their spatial relationships. The anatomical and spatial relationships transform the database of structures to a form essential for navigating through the body. These entities and their relationships are described next in the data model.

b. Data model
The data model, as shown in the object relationship diagram in Figure 2.3, is organized around a number of objects representing: the body (i.e., the male or female cadaver), body regions (e.g., thorax, abdomen, etc.), the anatomical structures (e.g., organs and their parts), and the images. Note that the data model is general enough to accommodate future data from other cadavers or their parts.

The relationships among these objects may be either hierarchical or not. Hierarchical relationships allow for inheritance of shared properties. For example, the male and female body objects inherit generic descriptions of a heart and cardiovascular system, but have different reproductive systems. Non-hierarchical relationships (that do not exhibit inheritance properties) represent the anatomical relationships among anatomical objects. These are modeled in terms of part-of and contains (or has-part) relationships as shown in the conceptual model in Figure 2.3.

Each of the objects is described in terms of its attributes. Simple attributes are brief, such as a name. More complex are the relational ones that contain instances of other objects which define a part-of /contains relationship between two objects.

As seen in the figure, the anatomical structure object consists of attributes for its name, superstructure, the region of which it is a part, and the physiologic systems to which it belongs. The "superstructure" attribute of an anatomical structure object refers to a second anatomical structure object which contains the first object. For example the *right-ventricle*'s superstructure is *heart,* which means that the *right ventricle* structure is part-of the *heart* structure. The "body region" attribute points to a body-region object which contains the anatomical structure object. For example, *heart* belongs to the *thoracic* region. A physiologic system is a function of the body in which a structure is a component. For example, *heart* is a component of the *cardiovascular* system.

In addition, each of these database objects points to image metadata objects in the database representing data descriptive of an available image (whether MRI, CT, a color slice**,** a rendered image, a segment mask, etc.), including its size and the name of the image file in the Visible Human file server where the images actually reside.

The male and female body objects are represented in terms of their basic characteristics such as age, gender and race.

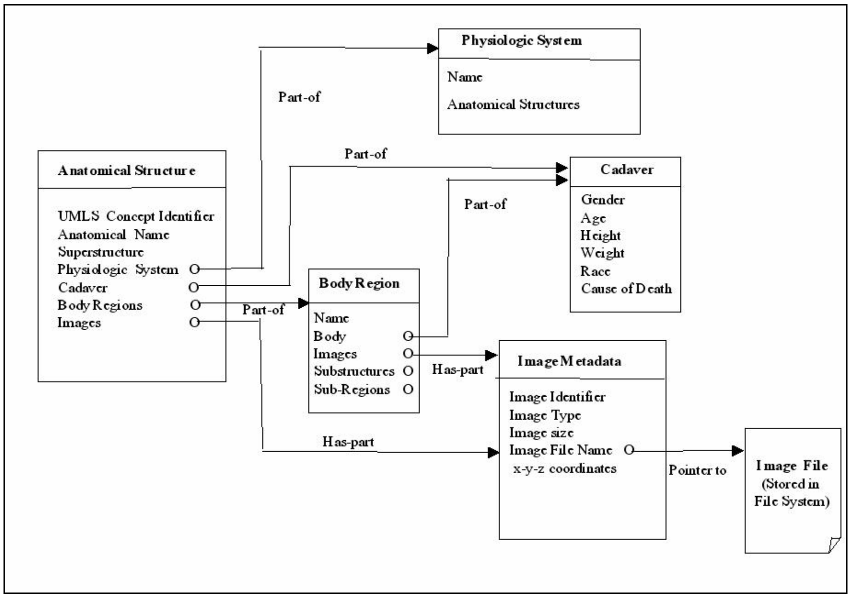

Figure 2.3. Database conceptual model

c. Object oriented framework

The choice of the object oriented framework for programming and data modeling was influenced by a number of factors including the structure of the data, and the requirement for efficient data representation and retrieval, as discussed below.

- ✍ Data structure
  Anatomical data consists of both spatial and structural information, which in turn are represented by other complex objects. For example, a *heart* object consists of four other objects, each describing one of its chambers in terms of other substructures, e.g., *right atrium* containing the *right auricular appendage*. Such nested structures form graphs of objects that are represented more readily in object oriented environments.

- ✍ Efficient data representation
  An object oriented programming framework provides efficient data representation through instantiation and inheritance. Male and female hearts, for example, are instances of the same structure which can be described once but *instantiated* twice. To give another example, a degenerative heart *is a* normal heart with extra conditions associated with it. This is-a relationship forms a class hierarchy which allows specialized objects to *inherit* characteristics of their exemplar objects, thereby providing a more efficient representation for reuse and maintenance**.** Since the "healthy" heart object is described by the common characteristics of heart, the "degenerative" heart only needs to add the extra conditions, because it inherits the generic information from its exemplar object (the healthy heart). These examples demonstrate the benefits of the instantiation and inheritance of objects in the object oriented framework allowing efficient data representation, modeling, and maintenance.

- ✍ Efficient data and image retrieval
  A goal of the 3Dsystems project was to enable the development of human atlases in which users may explore the body by navigating through structures, substructures, and sub-sub-structures by using the spatial and structural relationships among anatomical parts. This requirement is best supported with the graph navigation property of the object oriented framework.

d. Object oriented database

The advantages of the object oriented framework listed above determined the selection of an object oriented database over a relational alternative. Object oriented databases offer the following advantages over their relational counterparts in the object oriented programming environment used in developing AnatLine:

- ✍ Transparent persistence: The objects within the programming environment are automatically saved in, and retrieved from, the database. This makes the database an extension of the computer memory and hence transparent to the programmer. For example when navigating through a *heart* object, its *right ventricle* is retrieved from the database transparently when needed, with no additional programming. For relational databases additional modules would be required to provide the necessary mapping between the database and the object in memory.

- ✍ Unified model: A single model for representation of data in the object oriented programming language as well as the object oriented database is preferable from the point of view of reduced effort in development and maintenance, and eliminating mapping between two disparate models.

- ✍ Ease of navigation: The navigation through objects corresponds to a graph or tree, thereby fitting the requirement for a navigable human image atlas.

In light of these factors, ObjectStore® was selected as the database management system (DBMS). It was found to provide efficient data management, good performance, concurrency and multithreading. Unlike relational or object-relational DBMS which retrieve related rows of data by executing joins at runtime, ObjectStore stores and manages data components and objects with their relationships intact. Also, as in any DBMS, ObjectStore offers concurrency (allowing multiple users and applications to simultaneously access and update the database) and a multithreading feature (allowing the use of kernel threads, asynchronous I/O and shared memory.)

3 The AnatQuest system

3.1 Need for public access

In this section we give reasons for developing the AnatQuest system followed by a discussion of its design. Basically, AnatQuest became necessary because AnatLine was not suitable for the lay public. To begin with, the principal users of AnatLine were expected to be scientists interested in testing algorithms for image processing (e.g., registration, segmentation, feature extraction) or constructing 3D anatomical objects. For such expert users it was considered reasonable to have them download software for disassembling the retrieved files and displaying them. However, this approach posed significant barriers for the lay public. First, the requirement to download and install VHParser and VHDisplay was neither desired nor easily done by novice users. The second barrier was the lack of immediate visual feedback, a consequence of the large size of the VHI files and the difficulty of unbundling them quickly. Some files are on the order of 1 GB when segmented slice images and bitmap overlays for some of the larger organs are packaged. But lossy compression, which would have yielded significant file size reduction, was not considered: to preserve the complete content of these files. Furthermore, to transfer such large files to a user site in a reasonable time requires connections that not only are high speed links, but need to be reliable to avoid resending because of intermittent connections. Such robust high speed links are often not available to the lay user.

As a consequence of these factors, AnatLine found relatively low use overall, and almost none at all by the general public. Hence our development of the AnatQuest system.

3.2 AnatQuest: Design considerations

The design of AnatQuest retained AnatLine's database system, but its goal was to provide widespread access to the VH images for lay users with special attention to those with low speed connections as well. AnatQuest offers users thumbnails of the cross-sectional, sagittal and coronal images of the Visible Male, from which detailed (full-resolution) views may be accessed. Low bandwidth connections are accommodated by a combination of user-adjustable viewing areas and image compression done on the fly as images are requested. Users may zoom and navigate through the images. As shown in Figure 3.1, the number of hits for AnatQuest far exceeds that for AnatLine.

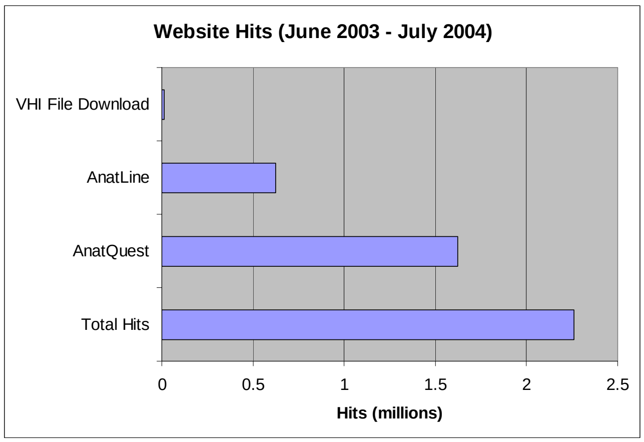

Figure 3.1.

In addition to its main purpose, AnatQuest serves as an entry point for both the FTP server for bulk downloading of VH files as well as all the functions of AnatLine. Through AnatQuest the user may also retrieve more than 400 surface-rendered objects created at the Lister Hill Center as well as a few samples from outside sources.

Since a key goal is to accommodate low bandwidth users, we had to address the file size problem. The large size of the VH images, both in totality and as individual files (7.5 MB for the CCD files; 33.2 MB for the scanned 70mm photographs), motivated parallel inhouse research in the compression and transmission of these images.[8-13] Studies were conducted in both lossy and lossless compression techniques. Among the lossy techniques investigated were JPEG and Digital Wavelet Transform (DWT) followed by scalar and vector quantization. For equivalent compression ratios (CR), DWT was found to yield better quality, artifact-free, decompressed images, but was computationally too expensive for real-time operation. Moreover, DWT would require client-side plug-ins, while JPEG is accommodated by most Web browsers.

Lossless techniques (Unix compress and its variations) yield low CR, on the order of 2 to 3). We combined background removal with a lossless method (arithmetic coding) and achieved a CR of about 9. Though a considerable improvement, this figure was deemed too low for practical use in transmitting VH images to the AnatQuest user. Our final choice was to remove the background (i.e., convert the blue background of all the slice images to uniform black), followed by JPEG compression. Other details on improving transmission rate are given later in this section.

The main effort in developing AnatQuest focused on creating suitable image viewers and server-side image processing modules, each dictating different sets of development tools.

Two image viewers are provided in the AnatQuest GUI: A Rendered Image Viewer (RIV) to display rendered images in 2D projection, and a Cut-away Viewer (CAV) to display thumbnail as well as detailed-view images of two-dimensional slices of the front (coronal), side (sagittal), and top (axial) views of the body. Both image viewers are viewed through a standard Web browser and do not require the installation of any plug-ins.

The image viewers were designed to serve as the clients in a three-tier client-server architecture (Figure 3.2), in which a tier is a logical partitioning of an application across client and server. The image viewers have only the graphical user interface, while the middle tier contains the application logic, and the third tier consists of the image server.

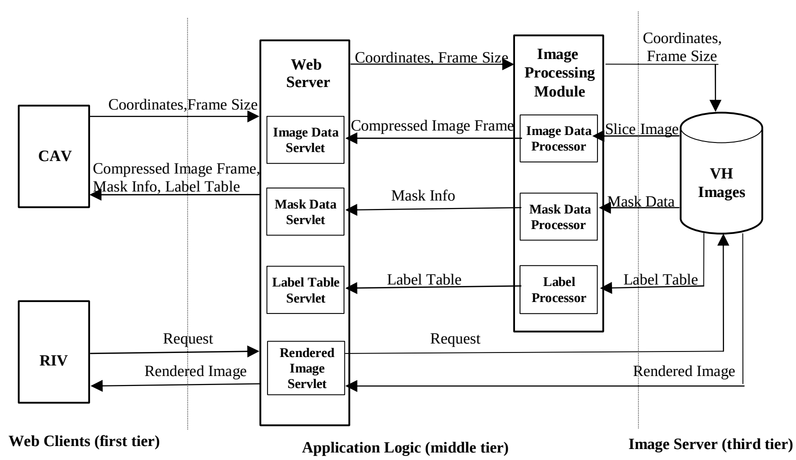

Figure 3.2. AnatQuest's 3-tier client-server architecture

The three-tier client-server architecture, also called “thin-client architecture,” was chosen over a two-tier approach for better scalability and maintainability. The three-tier approach has the advantage that any changes made to the application logic (middle tier) do not require changes to the client (first tier). By contrast, in a two-tier architecture, the client typically contains the application logic that sends requests to the server or database, and processes the returned results sent on to the user. This architecture is commonly referred to as “fat-client” because most of the application logic resides in the client. Although this is easier to build than the three-tier architecture, the graphical user interface (here the image viewers) would be closely tied to the application, and any changes to the application would require modification of the client as well, thus making two-tier applications less scalable and maintainable.

We chose to write both image viewers as Java applets rather than servlets (programs that reside in a Web server's servlet engine), though either would meet our objective of allowing user access to the system via a browser, such as Microsoft's Internet Explorer or Netscape Navigator, without having to download and install plug-ins in the browser. In other words, since the standard browsers have the built-in Java Virtual Machine (JVM) necessary to run any Java program, the image viewers could be either applets or servlets. Our choice was dictated by empirical testing that proved that applets were superior to servlets in allowing users to scale and manipulate the horizontal and vertical hairlines and field-of-view controls that float over each of the three view ports in an intuitive and pleasing way.

At the middle tier of the system, however, the application logic was written as servlets. These servlets transfer the burden of computing the size of, and retrieving, an image to the server side. They process the request for a new image on the server, retrieve the image, compress it on the fly, and return it to the applet for display. This interaction between applets at the first tier and servlets at the middle tier provides the fast response necessary for a wide user community equipped with low speed connections. (Running the Web server and the servlet engine in this middle tier are Apache and Java Jakarta-Tomcat, respectively).

The Cut-away Viewer allows the user to dynamically navigate the Visible Human body via a Web browser along three dimensions: along the x, y, and z axes. As shown in Figure 3.3, the user sees three side-by-side view ports that occupy the left part of the Web page, each view port containing a thumbnail representation of a sagittal, coronal, or axial slice. A user-movable hairline controls the location of the three axes (planes) in each view port. To dynamically display a different cut-away view of the body in three directions, the user moves a hairline vertically or horizontally across any one of the view ports, resulting in updating the corresponding spatially related one. In this way, the first view port (control source) and the second view port (control target) form a coordinated pair.

Two parameters determine the amount of data transferred to the client interface, thereby accommodating users with varying bandwidth connections: the size of the field-of-view which is user-adjustable, and the degree of image compression, also selectable by the user.

When the user clicks on the rectangular field-of-view box in any of the view ports, a detailed view of the section of the image encompassed by the box is displayed. The user may select the size of this detailed view (and therefore the number of bytes received from the server) from a pull-down menu in the GUI. The three field-of-view sizes are: 448 x 320 pixels (for low bandwidth connections); 640 x 448 pixels and 640 x 640 pixels, for higher bandwidth connections. At the user's click, the coordinates and size of the selected field-of-view are sent to ImageDataServlet on the server.

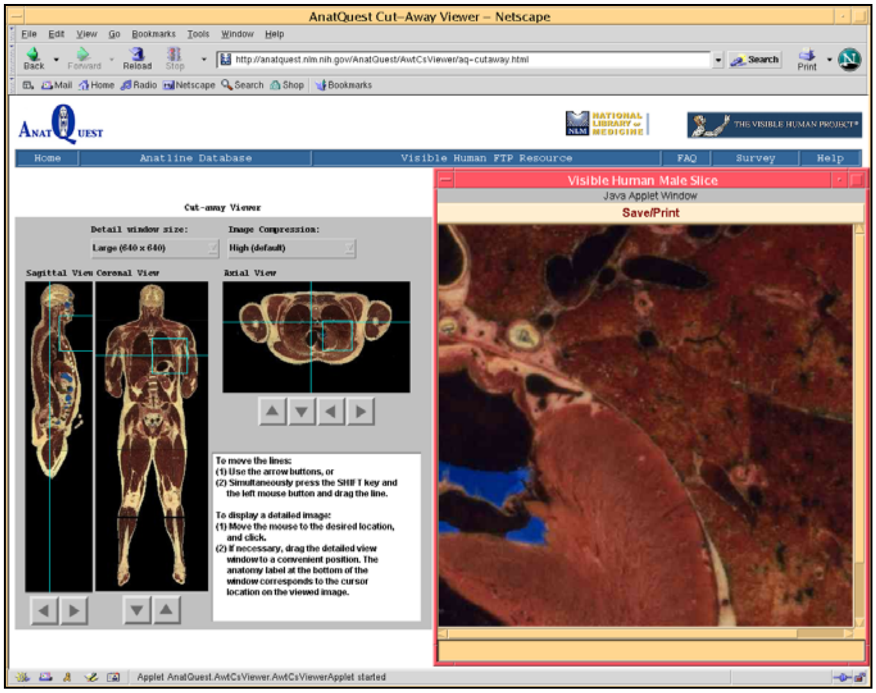

Figure 3.3. AnatQuest Cut-away Viewer

This servlet validates the parameters, passes the information to another server-side program called ImageDataProcessor which extracts the cryosection image file that contains the requested detailed image frame according to the specified position and dimensions. It then uses the JPEG compression scheme in Java Advanced Imaging to compress the image data at a user-specified compression ratio. The generated image data is then returned to the ImageDataServlet, which sends the data to the applet at the client to be displayed. Similarly, mask information is retrieved through MaskDataServlet and MaskDataProcessor, and label table data (a table of anatomical terms) through the LabelTableServlet and LabelTableProcessor. The label table and mask information enable the detailed view window to display the label (name) of an anatomic structure under the mouse cursor.

In order to have the region of interest efficiently encompass the detailed image, i.e., at their boundaries, we picked view port sizes that are divisible by 64. Also, since the files are stored as tiled TIFF images, and the average user's screen resolution is expected to be 1024 x 768, we set the maximum window size to 640 x 640 pixels to enable users to display an image portion that is large enough to include a significant part of the image, without taking up the entire screen.

To increase the transmission speed of the detailed image portions from the AnatQuest server to the user's browser, the images are compressed on the fly using the JPEG compression scheme, as mentioned earlier. The user may select one of five compression levels from a pull-down menu in the GUI: high (the default value), medium/high, medium, low/medium, or low. The approximate compression ratios (sample averages) range from 60 at the high level to 11 at the low. To achieve a reasonable response time for users with low speed modems, we set the default window size to 448 x 320 pixels and the default image compression option to high. For example, assuming an average transfer speed of 33.2 kbps commonly available through such modems, these default values allow a JPEG image of approximately 7,168 bytes to be retrieved and displayed in the user's browser in 1.7 seconds.

The GUI of the Rendered Image Viewer applet consists of a user-selectable list of anatomic structures in available rendered images. Selecting an item in this list results in the display of its thumbnail image together with an indication of the full image size. When the user clicks on the thumbnail, the full-size rendered image is displayed in a separate window. Figure 3.4 shows the components of the RIV.

As noted earlier and summarized in the following, several aspects of the AnatQuest design exploited the functionality offered by Java Advanced Imaging. First, we exploited JAI's file handling capabilities to locally store very large image files in the tiled TIFF format that would be too resource intensive for most users to download. JAI was also used to extract only those tiles required by the user (as defined by the coordinates of the selected region of interest), and to compress the image data to a JPEG file, as mentioned earlier.

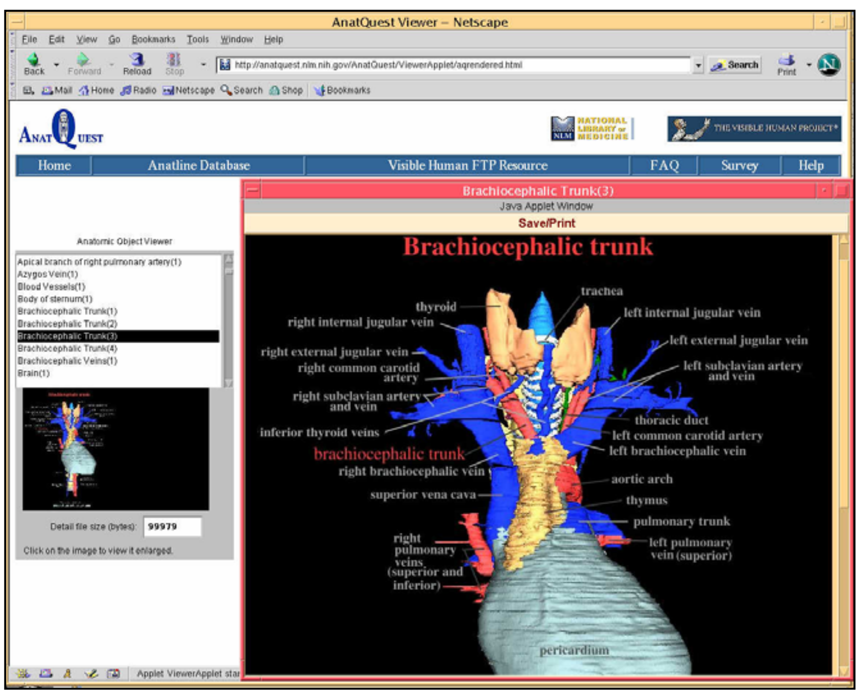

Figure 3.4. AnatQuest Rendered Image Viewer

As a class library, JAI supports generalized image processing functionality built as an extension to the Java programming language. In a simple programming model, JAI provides a rich set of imaging capabilities that can be readily used in applications without undue programming overhead. JAI encapsulates image data formats and remote method invocations within reusable image data objects, so that an image file would be processed the same way, whether in local storage or across networks. JAI also provides cross-platform imaging APIs, allows distributed imaging, and comes as an object-oriented API that is flexible and extensible. In addition, it is device independent and provides high quality performance on various platforms.

3.3 AnatQuest for onsite visitors

While reliable and rapid online access promotes the use of anatomic images by the public, onsite displays in exhibits are further opportunities to reach another public constituency: visitors to the library. One such exhibit was Dream Anatomy[14] installed at NLM for a year spanning 2002-2003. To serve as an onsite display in this exhibit, AnatQuest was modified to take advantage of the particular characteristics of this environment. Different issues come into play in the design of this modified system we call AnatQuestKiosk.

First, as an on*site* system, AnatQuestKiosk did not need to rely on a Web browser, and could therefore be designed as a standalone application. Second, an exhibit visitor in close proximity to the screen is inclined to navigate by touch; hence, a touchscreen monitor is provided (Figure 3.5). The design implications of these factors are discussed below.

Figure 3.5. AnatQuestKiosk: Touchscreen version of AnatQuest

Eliminating the need for a Web browser allowed both the Cut-away Viewer and the Rendered Image Viewer in AnatQuestKiosk to be designed as Java applications rather than as applets. Compared to applets, Java applications allow the user interface to be built with a greater variety of Java visual control class libraries (as in Java Swing), giving the interface a more polished look than possible with the Abstract Windowing Toolkit (AWT) libraries used in AnatQuest. Using AWT in the online AnatQuest system is necessary since these libraries are built into Web browsers, while Swing libraries are not. Swing libraries have to be downloaded into the Web browser as plug-ins, adding to the download time for an online system. Since this is not a problem with a standalone onsite system, the use of Swing in AnatQuestKiosk is an advantage.

While touchscreen monitors are attractive for onsite applications, in designing applications to run on them one must take into account the size of the GUI controls. For example, buttons and sliders must be large enough to respond to the touch of a finger. To support this, we replaced the thin vertical and horizontal user-movable hairlines that control the thumbnail images in AnatQuest with large sliders. This required the modification of the Java library slider controls to achieve a tailor-made look-and-feel.

To ensure that AnatQuestKiosk runs continuously, and covers the entire screen, it is necessary to prevent users from inadvertently stopping or starting the application, or from coming in contact with the operating system. We met these requirements by using Java's Fullscreen Exclusive Mode API, a new feature in JDK 1.4. This API supports high-performance graphics by suspending the operating system's windowing function so that the application takes full control of the contents of video memory and draws directly to the screen.

Finally, as an exhibit display, it is desirable for the AnatQuestKiosk application to be unaffected by network outages. To guarantee uninterrupted use, the image files are locally stored in the system to avoid continuous fetching from a remote site. However, the application is designed to fetch images remotely as well by enabling the parameters passed to its startup script to specify the image site as a URL for the image database server. This feature is useful in the event that new images, particularly newly rendered 3D structures, are added to the database.

3 Next steps

There are a number of interesting directions that the AnatQuest project is taking to further serve the lay public. Among these are: (1) Increasing the contents of the image database; and (2) Extending information systems designed for the public, e.g., MedlinePlus, to provide anatomic images. The full realization of the second goal depends on adequately addressing the first, as discussed below.

4.1 Increasing content

We believe the lay public would be well served with more image content in the database, and in different forms, e.g., 3D volume and surface renderings, and animated sequences. To our knowledge, existing images from third-party sources have been created from the CCD-captured cross-sections and not from the high resolution version, thereby possibly precluding the most detailed structures.

To date, we have produced a limited number of 3D surface-rendered structures, mainly as exemplar images to evaluate the design of the database and the processing tools. These images were created from the segmented and labeled *high resolution* cross-sections of the Visible Male thorax region, with the assumption that this dataset would allow rendering of more detailed structures than possible with the lower resolution CCD-captured data.

Currently, AnatQuest provides access to these images, numbering about 200 (showing about 400 anatomic structures), as well as some produced by other organizations.

However, the fact remains that other regions of the high resolution Visible Male, and all of the Visible Female, have neither been segmented nor labeled. If one is to make structures in all regions of the human anatomy available, this would be a necessary task.

User feedback suggests a demand for rendered images of structures in all regions, but particularly in the head and abdomen. That these regions are of highest interest is evident not only from anecdotal evidence, but also from the cumulative demand statistics shown in Figure 4.1.

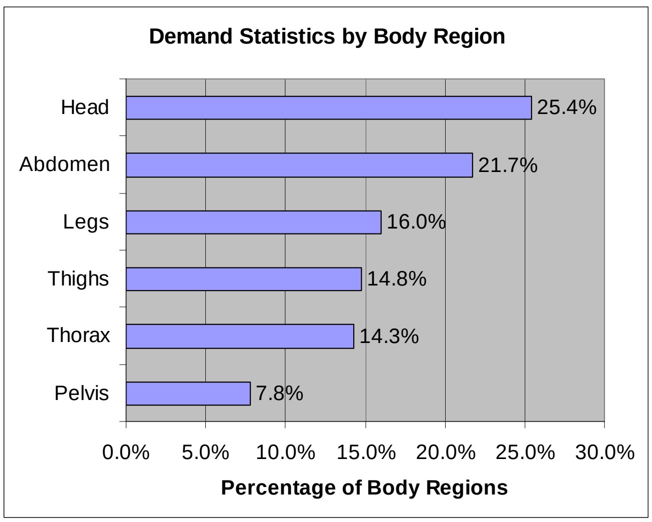

Figure 4.1.

Meeting this demand would require these regions to be segmented and labeled. The approach taken in segmenting and labeling the high resolution slice images of the thorax region, though computer-assisted, had a manual component and was therefore labor intensive. Since there appears to be work done in segmenting the low resolution (CCD-captured) images (e.g., Gold Standard), it is possible to conceive of more efficient techniques based on pixel statistics and interpolation to use the low resolution data to segment the high resolution images. The availability of a complete set of segmented and labeled high resolution slice images for both the Visible Male and Female would offer developers material for creating and testing tools for surface and volume rendering, and would consequently supply the lay user with rendered images of objects anywhere in the human body.

We additionally note that successful rendering requires the correct registration (alignment) of adjacent slice images. The high resolution images are found to be misaligned to some degree, whereas the low resolution slices have been shown to be correctly aligned. It is possible to conceive of techniques to use the low resolution set as a reference to correcting the registration of the high resolution slices.

4.2 Linking text resources to image database

Here we come to one of our most important objectives. A long term goal of the Visible Human Project is to transparently link the print library of *functional-physiological* knowledge with the image library of *structural-anatomic* knowledge into a single, unified resource for health information. Indeed this has been echoed several times in the past, including the NLM's Board of Regents Planning Panel whose recommendations as far back as 1989 stated: "The NLM should encourage... research into methods for representing and linking spatial and textual information..."

In this section we present our early research in this area. We explore the steps required to link text from a biomedical document to the relevant images, and apply the concepts to the design of a prototype linking a search of MedlinePlus, a text-based document source popular with the lay public, to anatomic images in our database (described in section 4.3). We define the following four functions to implement such a linkage:

1 Document Analyzer: Identifying biomedical terms in a document.
2 Term Mapper: Identifying the relevant anatomical terms.
3 Image Locator: Identifying the images in the image database.
4 Link Assembler: Linking the identified terms to the images.

Figure 4.2 shows the top level architecture of the system and its components. Actual implementations might utilize the components differently. For example, the document analyzer and the term mapper can be provided as independent services for indexing documents outside of AnatQuest. For simplicity, all the above text and image services can be made available through a single API interface.

The AnatQuest API is a URL-based parametric API in which a desired image can be described in terms of its metadata in the URL statement. In addition, the user can specify: (a) whether mapping the terms through a vocabulary (e.g., UMLS) should be utilized, and (b) the form of the output desired (e.g., XML file, image file). The following is an example of the API for the metadata in XML for a heart image:

http://image1.nlm.nih.gov/pm/servlet/ImageLogic?name=heart&resultFormat=xml

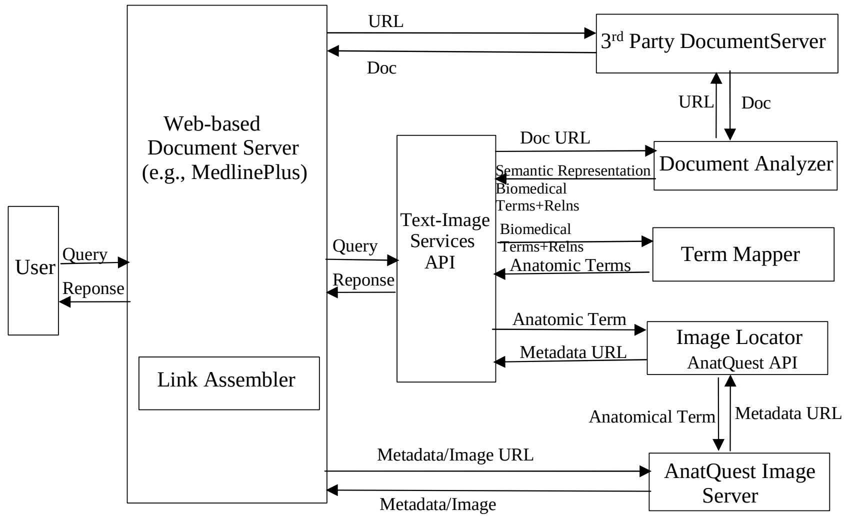

Figure 4.2. Architectural components for linking text and images

Document Analyzer: Identifying biomedical terms in a document

The document analyzer (parser) creates a representation of document content, in the form of a list of keywords, or in a more complex way as a semantic network representation of the contents.

Alternative approaches to implementing this function are:

- Word frequency vector
- Pre-assigned vocabulary terms
- Vocabulary-based term identification
- Vocabulary-based term and relation identification
- Text understanding

These alternatives range from the relatively simple (a word-based analyzer) to the most sophisticated (a full fledged text understanding system). The latter would be able to infer from the context, for example, whether the term "brain" should point to a male or female brain.

We focus on an approach that is reasonably practical at present: vocabulary-based term identification. This approach processes the text to identify the occurrences of the vocabulary terms in it. As implemented in the inhouse MetaMap program,[15] this takes into account such factors as lexical variances, word order variances and synonomy. The accuracy of this approach would depend on the level of sophistication of the analyzer. An alternative would be to use domain-specific semantic processing to identify the relationships as well as the anatomical terms in the document, thereby generating a richer representation of content.[16]

The parsing of documents can be initiated either by the Web-based document servers (e.g., MedlinePlus) or provided as a service by AnatQuest. Note that parsing may not be necessary if the documents already possess vocabulary-based indexing terms (e.g., MeSH terms assigned by an indexer) as part of their metadata.

Term Mapper: Identifying the relevant anatomical terms

Since our objective is to link text resources to *anatomic* images, the question is whether the biomedical terms identified in a document are anatomical. In fact, the chances are that they are not explicitly anatomical terms. However, using the concept relationships in NLM's Unified Medical Language System (UMLS) Metathesaurus we can map a biomedical term in the document to a related anatomical one. For example, the term *pneumonia* (a disease) could be mapped to *lung*, the underlying organ with the disease. Other mappings are also possible, e.g., mapping a particular anatomical term, for which there is no image in the AnatQuest image database, to a more general anatomical term for which there is an image in the database. But to demonstrate the concept we focus on the location-of relationship in our prototype system linking a search of MedlinePlus to images. Figure 4.3 shows the functions that are part of the term mapper.

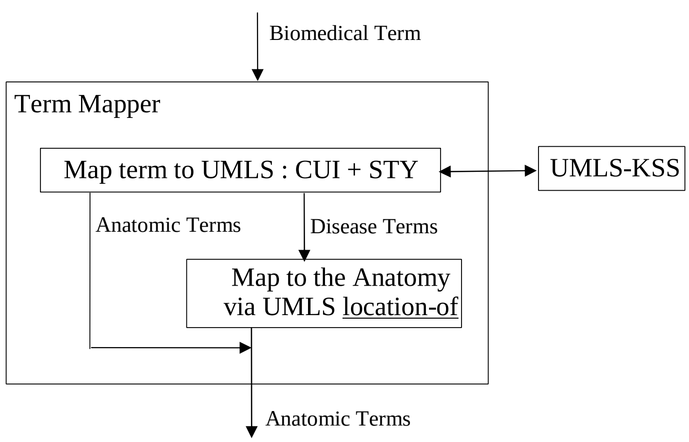

Figure 4.3. Term Mapper components

The following steps are taken by the term mapper:

1 Biomedical terms are mapped to UMLS concepts through the Knowledge Source Server, UMLS-KSS, resulting in their concept unique identifiers (CUI) and semantic types (STY). The mapping takes into consideration lexical variants and synonomy.
2 A term identified as anatomic by its semantic type is returned by the mapper as is.
3 For terms that are not anatomical (e.g., a disease name), the server uses the UMLS Metathesaurus concept relations to obtain the related anatomical term. As shown, a disease term would be mapped via the location-of relationship to the corresponding anatomical terms. This mapping appears to suit MedlinePlus since most health topic pages to which it points possess MeSH designators that are disease terms.

While our current prototype implementation of the mapper uses the location-of relationship in UMLS for mapping mostly disease terms to their underlying anatomical structure, we note that this relationship is only one among 88 available in Metathesaurus. Of these we have identified 22 that could potentially lead to relevant anatomical structures and hence images. Some of these relationships are: Is-a, part-of, has-part, branch-of, has-branch, has-tributary, tributary-of, manifestation-of, has-manifestation, broader, diagnosed-by.

While a chain of such relationships can link a biomedical term to a possible underlying anatomic structure, some types of relationship chains will not be successful, as shown in examples below. [Note that since in the UMLS Metathesaurus concept relation table the relation names are from the second concept to the first, we are displaying the relation names in the chains in *reverse* order.] The following are some findings and suggested heuristics for the selection of chains and their ranking:

✍ Combination of mapped-to and mapped-from creates a sibling relationship which might map to unrelated concepts ("nose") in the following path generated for heart attack ("myocardial infarction"):

Nose location-of --> *Necrosis of nose* mapped-to --> *Necrosis* mapped-from --> *Myocardial Infarction*

✍ Combining part-of and has-location relations provide substructures which might not be quite relevant, so they should be ranked lower, or deleted. Example:

Mitral Valve part-of --> *Heart* location-of --> *Heart Diseases* mapped-from --> *Myocardial Infarction*

✍ The mapped-from relation provides a broader concept than mapped-to, suggesting that mapped-from should be ranked higher. Example:

Necrosis of ovary mapped-to --> *Necrosis* mapped-from --> *Myocardial Infarction*

Other heuristics suggested in the course of our investigation are:

✍ Shorter chains should be ranked higher than longer ones.

- ✍ The combination of is-a and broader should be avoided. They create a sibling relationship that may yield unrelated concepts.

- ✍ The mapped-from relation should be ranked higher than the broader relation, because the former is more specific and the latter too general.

On evaluating a number of sample paths for a given input term we have identified some relations and their combinations which generate the most promising mapping of biomedical terms to their related anatomical structure. The following are relations which result in successful identification of anatomic terms (in descending order of effectiveness): location-of, mapped-from, is-a, broader, has-part, has-branch, has-tributary.

Our implementation of the relation-based term mapper consists of a look up of a mapped-table in which each entry contains a biomedical concept and its related anatomical structure. This is followed by ranking the entries, and inserting the highest ranked ones into the document content.

While we have focused on the relation-based mapping strategy as outlined above, we are also exploring two other methods: image-based and model-based. The image-based strategy, like the relation-based method, uses the Metathesaurus to map biomedical terms to anatomical structures, but employs a different ranking approach. To rank the related anatomical concepts and their images, this method relies on the *number* of mapped concepts which are labeled on an image. For example, a heart image labeled with four mapped concepts of *Right-atrium, Left-atrium, Right-ventricle,* and *Left-ventricle* would be assigned a ranking of 4, whereas an isolated mapped concept labeled on an image would be given a lower rank of 1.

The third strategy, model-based mapping, is based on clustering the mapped anatomical concepts using the Metathesaurus concept relation table. The mapped concepts in the most concentrated clusters will be assigned higher rankings. These techniques will be examined in research to follow.

Image Locator: Identifying the images in the image database

There are three possible approaches to identifying and accessing the images, each influenced by the design of the image database:

- ✍ Database-brokered access via a URL-based parametric API
- ✍ XML file-based access with searchable metadata
- ✍ Integrated image and metadata file format

The first approach is based on a private database system which brokers access to the images within a controlled environment, and custom-made interfaces designed for particular systems. The AnatQuest Web browser falls in this category. Here users can access the images through the GUI menus, the database is local to the system, and the database query language is not accessible to the users. In order to relax this constraint, as mentioned earlier, we have defined a URL-based parametric API in which a desired image can be described in terms of its metadata in the URL expression.

The second approach is to eliminate the database and store the metadata for each image in separate files that are made public for search engines to crawl and index. These metadata files, in addition to the information about the image, contain the URL to the actual image file. The metadata encoding can be as simple as unstructured text included in the ALT text of the IMG tag of an HTML document. Alternatively, the metadata can be represented in a more formal XML structure with data elements that can be searched individually. Search engines can index specific XML data elements, in order to provide results with high precision and recall. Alternatively, search engines may ignore the XML structure of the data and index all the metadata page contents. We have defined a detailed XML schema describing the structure of the image files which may be crawled and indexed by search engines. Figure 4.4 shows a fragment of the XML structure for the image metadata file.

In this example the web crawlers may be instructed by their owners to index the contents of the VHI-image.term.name nested field when it comes across an XML file with the root element <VHI-image>. Here the value of the name field is *Heart* allowing the crawler to add the URL to this example XML file to its list of index entries for the term *Heart*. Thereafter, when the search engine is queried for *Heart* the URL for this XML file will be returned. The GUI can then display desired elements of the XML file such as the name and the thumbnail for the image.

The third alternative solution for the image locator is to combine an image with its metadata in a single file. This has the advantage of preventing dangling links between the image and its metadata when they are stored in separate files and moved individually. This approach requires a standard file format that encompasses both pieces, as well as a standard set of metadata. A number of groups are attempting this. For example, the PNG2000[17] and extensions to the Dublin Core-based metadata activities[18] seem to be moving towards this objective. This approach also requires modifications to search engine crawlers to open a compressed image for retrieving and indexing its metadata.
A PNG-enabled search engine should be able to decompress a PNG file format, as well as know about the metadata in order to extract and index the data elements. An example of such an implementation is the PNG-enabled version of HotMeta.[19]

```xml
<?xml version="1.0"?>
<VHI-image xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:VHI="http://anatquest.nlm.nih.gov/vhi">
	<term anatomicalType="Structure">
		<cui>C0018787</cui>
		<name>Heart</name>
	</term>
	<specimen>
		<id>1</id>
		<name>Visible Human Male</name>
		<owner>National Library of Medicine (NLM)</owner>
		<sex>Male</sex>
		<race>Caucasian</race>
		<age>38</age>
	</specimen>
	<image size="59008" format="jpg" modality="70mm" capturedBy="NLM">
		<rendered segmentedBy="EAI" renderedBy="NLM">
			<url>...</url>
			<title>Anterior View of Heart</title>
			<modifier>1</modifier>
			<thumbnail size="5073" format="jpg" dimensions="">
				<url>...</url>
			</thumbnail>
		</rendered>
	</image>
</VHI-image>
```


Figure 4.4. XML structure of image data

Link Assembler: Linking the identified terms to the images

This function is intended to incorporate into the document links that point to the images associated with the biomedical terms found in the document. Ways to present the links depend on the design of the GUI of the Web-based document server. Possible approaches are:

- Hot linking anatomical terms in a document
- Related-Images button or menu option, based on document similarity criteria
- Portal approach

In the hot-linking approach, each term identified in the parsing phase is converted to a hot link that indirectly points to a relevant image in the Visible Human Image database through the AnatQuest API. The degree to which such a link points to a specific image correctly depends on the ability of the document analysis system (parser) to resolve ambiguities. For example, by default the links would point to images of the Visible Male unless the parser can understand from the document's context that the topic is about the female. Needless to say, this is a difficult task with a high probability of error. Figure 4.5 provides the flow diagram for the hot-linking approach, with numbers representing the sequence of actions following a user query.

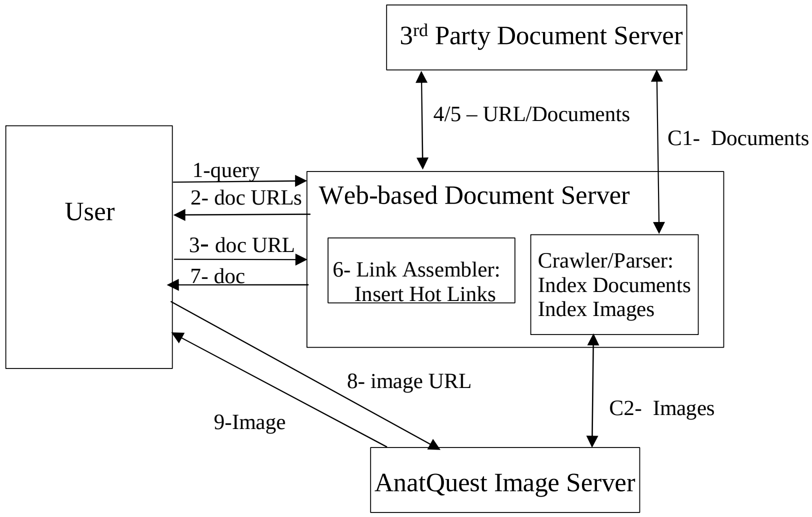

Figure 4.5. Link Assembler – hot linking anatomical terms

As shown in the diagram, steps C1 and C2 require the Web-based document server (e.g., MedlinePlus) to compute and save the offsets of the anatomical terms within the documents at the indexing phase so that hotlinks may be inserted in the third-party documents at time of retrieval. The Web-based document server would not store these documents but would act as a proxy server for them in order to insert the hotlinks.

An advantage of this approach is that the anatomical terms are turned into hotlinks which the user can easily click on while reading the document. A drawback is the possibility of mismatch between the text and the image due to parsing inaccuracies. Further, the server needs to maintain the offsets to all the anatomical terms within the document, requiring additional storage and processing to include the hyperlinks for each retrieved document.

In the Related-Images approach to displaying the links the original text of the document is displayed for viewing by the user. The GUI provides a "related anatomical images" button in proximity to the text. This button, when clicked, will send the set of vocabulary terms, identified earlier in the parsing phase and already associated and stored with the document metadata, as query terms to the image database. Thumbnails of the returning (matched) images will then be displayed on a sidebar next to the main document for the user to view. Selection of each thumbnail will open up a detailed view of the image. Alternatively, the thumbnails of related images may be shown on a sidebar when the document is first displayed instead of providing the Related-Images button.

An advantage of this approach is that it reduces possible confusion caused by a mismatch between terms and images, since the links or thumbnails are offered as "related images" rather than "exact images" suggested by hotlinked terms in the text (as in the previous approach).

In the portal approach to displaying the links, the documents need not be analyzed at all, nor are changes made to them. We simply make the Visible Human image database Website open to the Web crawlers of the Web-based document servers (e.g., MedlinePlus). These crawlers can then build an index to the images in the database, and simply treat the images as individual documents. Through metadata, the image database would provide the anatomical terms and possibly their synonyms to be indexed by the crawlers. Advantages of this approach are that: (a) no changes need be made to existing Web-based document servers; and (b) since the documents do not need to be parsed, parser accuracy would not be an issue. The flow diagram for the portal approach is given in Figure 4.6. The interaction consists of:

1 The user enters a term at the query prompt of the document server.
2 When the document server is MedlinePlus, the browser displays a list of (third party) documents grouped under a number of categories. For each document, a one-line summary is displayed together with the hotlinked URL to access the full document. One of the categories is "Images", under which appears a list of anatomical terms the crawler has found to be relevant to the user query.
3 A document URL points directly to a third party document server.
4 A returned document from this server is displayed for the user.
5 An image URL points to the AnatQuest Image server.
6 An image or its metadata file is returned from this server.

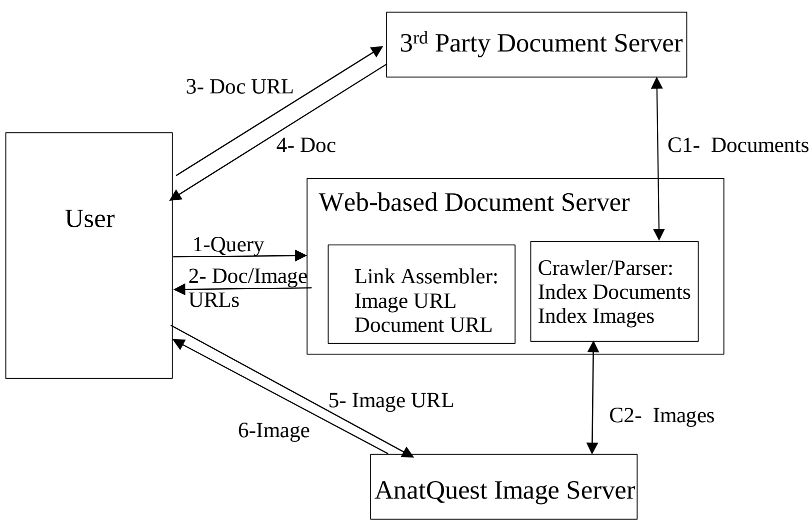

Figure 4.6. Link Assembler – portal approach

As shown in the diagram, steps C1 and C2 consist of crawling documents in the third party document server as well as the image files, and generating indexes for both images and documents. These indexes are used by the link assembler component.

The advantage of this approach is that it requires no changes to Web-based document servers such as MedlinePlus. A drawback is that only images related to the user query are provided, and not those related to the contents (or “meaning”) of the documents.

A comparison of the three alternatives for linking terms to images suggests that the best approach might be to combine the portal and the Related-Images methods. This approach not only provides the images matching the user queries (portal approach) but also allows the user to view the images related to the contents of the selected documents. Further, when the metadata in the relevant documents include MeSH headings, this approach avoids the parsing requirement of the hotlinking approach.

4.3 Implemented prototype: MedlinePlus proxy server

We have implemented a prototype of a text-to-image linking system based on the MedlinePlus health information Web server. As shown in Figure 4.7, a proxy server has been developed to intercept the user request to MedlinePlus. The proxy server first retrieves the MedlinePlus page that satisfies the user query, and in parallel sends the user query to the AnatQuest image server which uses the UMLS Knowledge Source Server and a term mapper module to map the query terms (mostly disease names) to the corresponding anatomical structures. The links to these images are then inserted by the proxy server as hotlinks in the image section of the MedlinePlus page, which is then returned to the user.

In a variation of the prototype, instead of parsing the text, we could use the MeSH terms attached to the document, if these exist, since these terms may be assumed to represent the focus or topic of the document. The document is then linked to the images through the MeSH terms.

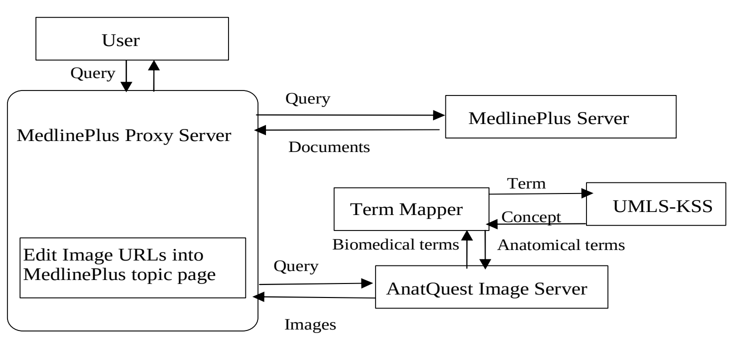

Figure 4.7. Prototype implementation

Our initial prototype serves to demonstrate the feasibility of three of the four functions necessary for text-to-image linkage: viz., the term mapper, image locator and link assembler. It does not address the document analyzer stage. Instead, the user query, rather than the contents of the returned document, is mapped to the anatomic term which then is linked to the image. Even the functions included are not addressed comprehensively, as for example, the term mapper uses only the location-of relationship. Nevertheless, this prototype provides a platform on which a more extensive testbed may be designed to address research issues related to the larger problem of linking text in documents to images. Some of these issues are:

1 Relevance of images to the topics in a document. This is influenced by the types of mappings used by the term mapper. For example, would the location-of relationship used in this prototype result in the display of appropriate images (of a diseased lung rather than those of a healthy lung) for a document about pneumonia? This measure is subjective, but may be quantifiable by user studies.
2 Precision and recall of images. These well-defined and quantifiable measures reflect the performance of the image retriever searching the metadata in order to locate desired images.
3 Accuracy and precision of the links. This is of particular importance to the hotlinking approach for the display of links where document terms are hotlinked to the images. This approach might raise user expectations for an "exact image" rather than a "relevant image". An image may be seen as relevant but not necessarily “accurate”.
4 Term mapper performance. Considering other promising relationship mappings (e.g., is-a, part-of, etc.) should increase recall. In addition, a ranking of the individual mappings should provide a measure for judging the relevance of the resulting anatomical terms.
5 Parser performance. In addition to extracting the anatomical terms, identifying the explicit anatomical relationships in the document enables formulation of more refined search queries against the image database, which should increase relevancy.
6 System performance. Some approaches require more resources than others, which could translate into higher turnaround time. For example, the hotlinking approach requires modifying each document before it is sent to the user. Turnaround time, excluding the network delay, should provide a measure of system performance.

Addressing these research issues should allow us to identify the best combination of the alternative solutions in each of the four functions described here for an optimum solution for linking document text to images.

5 Summary

The goal of providing the public ready access to anatomic images, in particular, “real” human anatomy from the Visible Human Project, is one aspect of universalizing access to biomedical information. This goal is being implemented in the AnatQuest project in which a system has been developed that provides Internet accessibility to *high resolution* Visible Human images via browsers, labels for anatomic structures in each cryosection slice as well as 3D rendered images, and acceptable user interaction in a low bandwidth environment. Besides the AnatQuest system for Web access by resource-limited end users, we have also developed the AnatQuestKiosk system for onsite visitors to the National Library of Medicine, and a prototype system demonstrating the linking of queries by MedlinePlus users to anatomic images. We have also defined an architecture and functional components to further investigate the linking of the text content of documents retrieved by MedlinePlus and other document sources to anatomic images in our database.

6 Acknowledgments

The extensive R&D and implementation activities in the AnatQuest project are the work of a team at the Lister Hill Center's Communications Engineering Branch. In particular, Dr. Amir Razi has made significant contributions in research toward the architecture design and component definition of a system linking textual document sources to anatomic images; Mr. James Seamans is responsible for an automated system for the registration of Visible Human cryosection images; Ms. Annette Strupp-Adams and Ms. Jinghong Gao, as well as Mr. Seamans, are responsible for the successful implementation of the Web and kiosk versions of the AnatQuest system and the prototype system for text to image linkage. Each of them has made important contributions to the research reported in this chapter.

References

1. Programs and Services, Fiscal Year 2001, National Library of Medicine, NIH, Bethesda, MD, p. 1.

2. NLM Website: www.nlm.nih.gov

3. The Visible Human Project: www.nlm.nih.gov/research/visible/visible_human.html

4. Ackerman MJ. The Visible Human Project. Proc. IEEE 1998 Mar; 86(3): 504-11.

5. Henderson E, Seamans J, Strupp-Adams A. VHIF: A prototype file format for anatomical images. Proc. Visible Human Conference, 1998. (CD-ROM from NLM).

6. Strupp-Adams A, Henderson E. Retrieving high resolution images over the Internet from an anatomical image database. Proc. SPIE: Internet Imaging, Jan. 2000, Vol. 3964, 259-65.

7. Zhou R, Henderson E, Seamans J. Visualization of Visible Human anatomic images. Proc. Visible Human Conference, 1998. (CD-ROM from NLM).

8. Thoma GR, Long LR. Compressing and transmitting visible human images. IEEE MultiMedia 1997 April-June; 4(2): 36-45.

9. Meadows S, Thoma GR, Long LR, Mitra S. Entropy encoding of difference images from adjacent Visible Human digital color photographic slices for lossless compression. In: Kim, Yongmin, editor. Medical Imaging 1997: Image Display. SPIE Vol. 3031, 1997; 749-55.

10. Long LR. Transmission of medical images over wide area networks. Proc. Visible Human Conference, 1996. (CD-ROM from NLM).

11. Mitra S, Long LR, Pemmaraju S, Muyshondt R, Thoma GR. Color image coding using wavelet pyramid coders. Proc. SSIAI'96, San Antonio, TX, April 1996; 52-63.

12. Pemmaraju S, Mitra S, Long LR, Shieh Y-Y, Roberson G. An adaptive vector quantization with fuzzy distortion measure for image coding. Proc. SPIE Medical Imaging '96, Newport Beach, CA, February 1996; 112-5.

13. Long LR, Berman LE, Neve L, Thoma GR. An applications-level technique for transmission of large images on the Internet. Proc. SPIE: Multimedia Computing and Networking 1995, Vol. 2417, San Jose, CA, February 1995, 116-29.

14. http://www.nlm.nih.gov/exhibition/dreamanatomy/da_technology.html

15. Aronson AR. Effective mapping of biomedical text to the UMLS Metathesaurus: The MetaMap program. Proc AMIA Symp 2001:17-21.

16. Rindflesch TC, Aronson AR. Semantic processing for enhanced access to biomedical knowledge. Kashyap V, Shklar L (eds.). Real World Semantic Web Applications, 2000, IOS Press, 157-72.

17. Portable Network Graphics: http://www.w3.org/Graphics/PNG

18. Dublin Core Metadata Initiative: http://dublincore.org/

19. HotMeta: http://archive.dstc.edu.au/RDU/HotMeta/png/index.html.