Developing a Web-based Object-Oriented Multimedia Medical System

Miguel Aluen*, Hector Arrechedera*, Alfredo Matteo**, Christiane Metzner**
(maluen@megaserv.com, harreche@reacciun.ve, amatteo@isys.ciens.ucv.ve, cmetzner@isys.ciens.ucv.ve)

1Instituto de Medicina Tropical, Facultad de Medicina, Universidad Central de Venezuela
**Escuela de Computación, Facultad de Ciencias, Universidad Central de Venezuela

Abstract
The goal of this work is to present an experience in the challenge of using Internet technologies for the development of a multimedia web-based medical application, using both multimedia computing technologies and object-oriented technologies. This medical application is focused on the development, diffusion and use of the technology in response to specific domain needs of medical experts in the area of tropical diseases.

Keywords authoring tool, multimedia, object-oriented methods.

1. Introduction
An ever increasing supply of information for analysis and understanding is generated as a product of scientific research, however there are many potential users who can not access and use adequately this information, specially in the countries of the third world. The “information highway” and the use of internet, web browsers and multimedia computers has opened easy access to a huge amount of information of distributed nature. People exploiting computer power can simultaneously transmit and receive text, sound and image via coaxial and fiber optic cables or radio and satellite systems, being able to organize and process information when and wherever they want. Multimedia is becoming more and more central as the network bandwidths and computer processing power grow. The World Wide Web considered as the embodiment of the infrastructure of today’s information age is currently unable to handle documents containing media such as audio and video in a user-friendly way.

However, any technology requires sound underlying methodologies in order to be effective. Object-Oriented (O-O) methods should enable the developer of hypermedia information systems to simplify and shorten the time needed for the development of this kind of systems, particularly due to the reuse of software components.

This work describes our experience in adopting a process that combines proven and well known O-O methods for the development of a hypermedia system with a Web-based User-Interfaces (WUI) which will be used in medical science education and dissemination of expert knowledge with particular emphasis in tropical diseases. The expert knowledge is extracted and stored persistently using a framework [4] tailored to the problem domain. The user interface to the application is through a web browser talking directly to the application that plays the role of a Web Server. This means that the user interface is exactly like the World Wide Web (www) on the Internet. End users must be involved in designing the electronic services and applications, which must be easy to use and in-line with real life needs.

2. The CAIBCO1 System
2.1 General issues
An important application of hypermedia systems is distance information providing and learning. Multimedia is more than multiple media; it adds interactivity to the combination of text, graphics, images, audio and video. In hypermedia, the user may select one of several objects on the screen in order to receive information that he wants, expressed in a media [2]. As Internet links together teachers and learners, these kind of systems are a great potential for global cooperation. Particularly, we need to provide resources to medical staff and students spread over wide geographical areas. Hypermedia documents are digitalized and stored in a multimedia database and they can be loaded, prepared and annotated by means of a hypermedia document editor and distributed fast over computer networks. Our objective is twofold. First, we want to develop, using the O-O approach, hypermedia tools capable of integrating existing and, later on, add new software for digitalization and edition of images. The advantage of this approach from the engineering point of view is that we use an O-O process that combines proven O-O methods to develop the tools and from the end user point of view, we provide a familiar environment allowing users to start creating their hypermedia documents, without the time gap necessary to learn how to work with new tools.

1Centro de Análisis de Imágenes Biomédicas COmputarizadas, http://caibco.ucv.ve
A second objective is to develop a modeless Editor supporting authoring and browsing of hypermedia documents avoiding the frequent gap between a small group of hypermedia authors and a large group of hypermedia consumers. To accomplish this, we require the following group of tools:

- Digitalizator and editor: necessary to create the multimedia objects that will be used in the hypermedia documents.
- Hypermedia Editor or Authoring Tool: used to create and annotate hypermedia documents.
- Web Browser: which is a standard market browser used to navigate through hypermedia documents.
- Administration Tool: used to register information about who is accessing the system and control how the objects in the system are used.

2.2 Design issues

**Requirement:** we need to build an Intranet capability with an intuitive interface, which allows viewing of medical assets by non-developers through Web browsers. Edit or mark-up facilities for distributed review are also desired.

**Rationale:** avoid dual maintenance of assets (HTML and tool internal).

2.3 System architecture

The architecture of the Caibco system [1] is based on the understanding that one hypermedia document consists of a set of pages. A page (HTML) can be filled with multimedia objects (MMO) and subsequently linked to other pages. A multimedia object is any piece of information such as sound, images, text or video. The images can be of different kinds: magnetic resonance (MR), computerized tomography (CT), X-ray, electrocardiograms (ECG) among others. Associated with the MMO’s is the design of mechanisms for user authentication and the assignment of read and/or write privileges. While the pages are not published even the read privileges may be restricted.

The Caibco system will be used in different medical areas, each with its own sets of documents. Central storage, as databases installed on servers, provides the capacity needed to store the information that can be referred in order to place a hypermedia document in a broader context. They should be stored in such a way, that effective search strategies are allowed and can be implemented either as a file system, a database or a hypermedia network.

To obtain independence of different possible implementations, a “home-made” Object Request Broker (ORB) is introduced which simplifies the communications and provides all the services related to storage and retrieval of hypermedia / multimedia.

The authoring tool, through a WUI, will establish a link to instantiate a framework in the medical domain. A framework is an abstract subsystem, consisting of a set of interconnected classes designed to be reused and extended. The pathology framework is defined by software artifacts that can be represented as a pathology graph. When the authoring tool is used to create a hypermedia document the framework is instantiated transparently for the user and the pathology graph is expanded.

The system is a client-server system and its architecture is shown in Figure 1. It involves an Intranet, comprised of Web technology and our organizations internal network. Every desk in the organization has a workstation with access to the Web browser software on the server. The client portion is platform independent but requires the use of a Java-enabled www browser. Anyone can launch the Web browser and specify the Uniform Resource Locator (URL) of the Web site of the group. Caibco's environment is a mixture of midrange Unix machines and PC’s from different vendors, and a TCP/IP backbone. All the tools of the Caibco System are developed with Sun/Soft Java.

2.4 The Development Process

The approach used for the development of the tools is an object-oriented process is defined in [7], where a combination of proven object-oriented methods is introduced. In this process the OOSE [5] and OMT [9] methods are combined, but the identity and flavor of each of the methods is maintained, taking advantages of the strong points of each one.
In Tables 1a and 1b the inputs and the deliverables or software artifacts obtained in each process step are shown.

<table>
<thead>
<tr>
<th>Step</th>
<th>Using</th>
<th>What to do</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Definition of the problem Use-case Model</td>
<td>Extract and enumerate use-cases, describe use-cases, build GUI prototypes</td>
</tr>
<tr>
<td>2</td>
<td>Requirement Model</td>
<td>Analyze use-cases: derive classes</td>
</tr>
<tr>
<td>3</td>
<td>Requirement Model</td>
<td>Apply strategy</td>
</tr>
<tr>
<td>4</td>
<td>Analysis Model, Use-case Descriptions</td>
<td>Analysis</td>
</tr>
<tr>
<td>5</td>
<td>Analysis Model, OID</td>
<td>Translate to OMT</td>
</tr>
<tr>
<td>6</td>
<td>Object Model, Dynamic Model</td>
<td>Decompose</td>
</tr>
<tr>
<td>7</td>
<td>Object Model, Dynamic Model, Functional Model</td>
<td>Object design</td>
</tr>
</tbody>
</table>

Table 1a. Inputs of each process step

<table>
<thead>
<tr>
<th>Step</th>
<th>Deliverable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Actors, Requirement Model: Use-case and GUI descriptions, Use-case Model</td>
</tr>
<tr>
<td>2</td>
<td>Requirement Model with Domain Object Model and Data Dictionary</td>
</tr>
<tr>
<td>3</td>
<td>Analysis Model</td>
</tr>
<tr>
<td>4</td>
<td>Object Interaction Diagram (OID)</td>
</tr>
<tr>
<td>5</td>
<td>Object Model, Dynamic Model, Functional Model</td>
</tr>
<tr>
<td>6</td>
<td>Subsystems specification</td>
</tr>
<tr>
<td>7</td>
<td>Object Model, Dynamic Model, Functional Model</td>
</tr>
</tbody>
</table>

Table 1b. Outputs of each process step

The process is centered on the analysis and design phases and is outlined below.

Steps 1 through 5 correspond to the analysis phase: step 1 and 2 describe how to develop the Requirement Model. Step 3 corresponds to the development of the Analysis Model. In step 4 a high-level Object Interaction Diagram is elaborated and in step 5 the transition to OMT is made. Steps 6 and 7 correspond to the design phase using OMT. Each step is further detailed next.

Step 1. Develop a Requirement Model: extract and enumerate the use-cases, i.e. list of use-cases, which is a list of all the facilities the system must provide. Continue the process defining actors and elaborating the use-cases describing the normal use and building the GUI prototype. Then, exceptional or error use-cases are described.

Step 2. A Domain Object Model that shows the domain objects and their relations is built along with a data dictionary.

Step 3. Develop the Analysis Model creating a partial object model for each actor and his elaborated use-cases, identifying interface, entity and control objects. Structure the interface objects according to the Use-case-PAC model [3,6]. In this model, for each human actor a central control object and a central interface object is defined. The central interface object corresponds to a window used by the actor to access all the functionalities (use-cases) which he interacts with. For each one of these use-cases an interface object is defined. Also for each actor a central control object is introduced expressing his control requirements. Notice that the interface and control objects are hierarchically structured. Entity objects communicate with interface objects only via control objects.

Step 4. Develop Object Interaction Diagrams, which define the interactions between the system and the actors but also the event communication between analysis objects.

Step 5. Translate the Analysis Model to OMT. Model changes of object states using state transition diagrams (STD), one STD for each class of objects with interesting behavior. Develop a high-level Functional Model using the schema proposed in [10].

The process during design consists of:

Step 6. Decompose and obtain a subsystem specification.

Step 7. Refine the models incorporating design issues. The process is iterative and incremental. If new requirements are identified or an existing one is changed during any step, we define the use-cases that satisfy these requirements or modify an existing use-case, changing accordingly the developed models. The process is also flexible, as the structuring of interface objects (step 4) can be done with any adequate model. Having defined these models using the OMT notation, organize and refine the classes, their relationships and their implementation as defined in the OMT design process [9].

### 2.5 Applying the process

To develop a Requirement Model using the previously defined process, the following high-level use-cases where identified:

- Digitalize and Edit a MMO: this use-case describes the interactions and options required for digitalizing and enhancing any valid diagnostic element, which must be
stored as a MMO for later retrieval and linking with documents.

- Construct and handle a document: this use-case describes the interactions identified for creating and handling documents that contain expert knowledge.
- Administer and control the resources (MMO’s and HTML) that are stored in the system and the users with their corresponding read and write privileges.
- Access Caibco’s public assets using a Java-enabled browser.

The figures 2 through 7 are part of the deliverables obtained in step 1. Figure 2 shows a very high-level use-case model of the Caibco system. The Caibco system has a WUI, a digitalization GUI (Graphical User Interface) and an Authoring WUI. Through these WUI's and GUI the actors communicate with the system. The digitalization tool is not available on the Web; it may only be used at our installations. The actors identified are: Laboratorist which is the actor in charge of the digitalization and edition of images that can be used with the authoring tool. He feeds the system with the images and their description. He may use any of the installed software for doing his job: Gscan, Capture, Imgwork, Moviemaker, and Movieplayer. New software for scanning and image edition may also be added. The Big Guru Medical Doctor (MD) is the actor whose expert knowledge will be incorporated into the system. This actor has great expertise in a particular medical area and although he may directly use the system to incorporate his knowledge, generally there will be a Little Guru MD that interacts with the Big Guru MD and the system. Both can be considered a specialization of an actor MD. The Browser is a non-human actor that interacts with the Caibco system and with a Web Navigator, a secondary actor that browses through existing and public hypermedia documents, typically students or physicians who want to learn or know something about a particular medical area. The Administrator is the actor responsible of all the administrative tasks related to the system and the ORB is the non-human actor which provides all the services related to storage and retrieval and use of the installed graphic software.

In Figure 3 and 4 incorporation of "extends" and "uses" relationships to the defined use-cases are shown. Figure 3 describes that if the Laboratorist needs using the system in order to Digitalize & edit, he (she) may add, remove, modify or view an element of his choice. Figure 4 describes that if an actor needs using the system in order to Construct & handle documents, he (she) will annotate a document and once finished, the corresponding HTML will be generated. In particular, notice that in Figure 4, the Annotate use-case can be extended with a link to a multimedia object (Link MMO) or with a link to an HTML page (Link Page).

Figure 5, 6 and 7 show the interfaces for the administration, digitalization and authoring tool respectively. All user interfaces are graphical but those to be used with a web browser are defined as WUI and those used locally without a browser are defined as GUI. Every pathology, a problem domain concept, has properties that are common to all diseases such as: common used name, other names that identify the same pathology, description, symptoms, treatment. These properties are defined in the Pathology entity object and inherited by all diseases. This generalization-specialization relationship is shown in figure 8 using OOSE [5] notation. Once the conversion to OMT models [9] is done, the relationship between the Pathology class and other classes is explored and shown in the partial Object Model of figure 9. Every pathology object is an aggregation of text, which describes the pathology, case studies and zero or more MMO’s such as anatomic pieces, electrocardiograms, X-rays, etc. It is the MD Big Guru’s responsibility to decide the MMO’s used to document pathologies and the case studies that he (she) considers relevant. Every pathology can be associated with a normal case against which the case studies may be contrasted. A case is also modeled as an aggregation of text describing the case and zero or more MMO’s.

---

2 Silicon Graphics software
Figure 3. Incorporation of “extends” relationship into the **Digitalize & Edit** use-case.

Figure 4. Incorporation of "uses" relationship into the **Construct & Handle Documents** use-case.

Figure 5. The WUI of the Administration Tool.

Figure 6. The WUI of the Authoring Tool.

Figure 7. The GUI of the Digitalization Tool.
In the context of this work, the strategy applied is the Use-case-PAC Model [3,6] but any other strategy can also be applied for the structuring of objects.

In figure 10 a partial structuring of the analysis objects for the Laboratorist is shown: a central control object (ctrl_digitalization) communicates with a central interface object (IFZ_digitalization) that detects events from the outside and with entity objects to satisfy services requested by the actors. The structuring is obtained by application of step 3 and the partial Use-case PAC model of the system is presented in figure 11.
The development team performed the testing of the system using unit tests at code level and stubs to mimic responses, as well as integration tests, combining units and exercising their interface and functionality. The development team consisted of one project leader, two software architects and three undergraduate students. The development time for the first version of the system was 8 months. This first version was implemented using 16 classes, 117 attributes and 148 methods, without taking into account Java classes. The average number of methods per class was 9 and the average number of attributes per class was 7 with an average number of lines of code per class of 596. Due to the innovative development tools and techniques, members of the team were learning the new technologies while the software was under development.

3. Conclusions and Future Work
This paper describes the development process of a set of tools intended to produce hypermedia documents containing the knowledge acquired during more than 30 years at the Institute of Tropical Diseases, Universidad Central de Venezuela. These hypermedia documents can be created and manipulated by non-expert users through a standard web browser. The computational problem has been solved and the tools have been implemented and proved by our technical staff. But even with a great support from different sponsors and being cataloged by them as a successful project, we have not been able to get the non-expert users, medical doctors, to use it consistently. It is, in our opinion, not enough considering success factors to have a good project manager, a good technical leader and a good development method. It is absolutely necessary a real organizational commitment and support and an education tailored to the audience.

The guidelines for our future computational work can be derived from this experience: it is only a first step in the use of Internet technologies as a resource provider in the area of tropical diseases. Moreover, it is necessary to expand Caibco’s environment incorporating specialized software for distributed systems in order to use the tools in the solution of clinical cases in remote areas and for active dissemination of knowledge. New facilities are needed, particularly for the reconstruction of 3-D images. Currently we are working on the development of a pathology framework and on the incorporation of design patterns [4] in a new version of the software.

Acknowledgments
This work has been supported by project No.96001320 of Conicet and project No.0935383297 of CDCH and it was also sponsored by Sun Microsystems de Venezuela,

---

3 Consejo Nacional de Investigación Científica y Tecnológica
4 Consejo de Desarrollo Científico y Tecnológico, Universidad Central de Venezuela
Digital, and Informix. The development was done by the following students: Milagros Rivero, Ana Maria Valero and Ramon Cruces as their undergraduate projects in Computer Science at the Universidad Central de Venezuela, Venezuela. The student Irene Aluen did the graphical editing.

4. References


