ELECTRONIC WORKSHOPS IN COMPUTING

Series edited by Professor C.J. van Rijsbergen

Rainer Manthey and Viacheslav Wolfengagen (Eds)

Advances in Databases and Information Systems 1997

Proceedings of the First East-European Symposium on Advances in Databases and Information Systems, (ADBIS'97), St Petersburg, 2-5 September 1997

WWW Access to a Philological Application

S. Calabretto and B. Rumpler

Published in collaboration with the
British Computer Society

©Copyright in this paper belongs to the author(s)

ISBN: 3-540-76227-2

WWW Access to a Philological Application

S. Calabretto, B. Rumpler

LISI-INSA de LYON

20, avenue Albert Einstein

69621 Villeurbanne Cedex

France

Tel : (33) 4 72 43 88 94 / (33) 4 72 43 83 92

Fax : (33) 4 72 43 85 18

E-mail : cala@if.insa-lyon.fr, rum@if.insa-lyon.fr

Abstract

Our project enhances the accessibility of ancient manuscripts and provide new ways of working with them through a Web server. More precisely, we aim to produce a software tool allowing the historians, and more particularly codicologists and philologists, to read the manuscripts, write annotations, and navigate between the words of the transcription and the matching piece of image in the numerized picture of the manuscript alongside the Internet network. A local version of this application is already available. In this paper, we present the design of the Philological Web Application. It is based on the ActiveX-based approach for database remote access. We describe how the Philological Application is made available through a server interface accessed by remote clients.

1 Introduction and Requirements

The transfer of texts and images onto digital media presents an interesting series of possibilities for those who, in various capacities, are concerned with the conservation of written documents, such as members of library staff, as well as for anyone carrying out studies in the field of philology.

Our project fits into this context and is aimed at two categories of users: the first is represented by the general users of a library who wish to examine manuscript sources ; the second category of users envisaged by the project is made up of professional students of texts. In other words, to use the term in its broadest sense, philologists or, to be more specific, critical editors of classical or medieval works that are hand-written on material supports of various types (paper, papyrus, stone). It includes, therefore, students of ancient texts as papyrologists, epigraphists, palaeographers, and codicologists: all those, in short, who are interested in studying, annotating, or transcribing the text contained in digitized and accessible manuscript documents.

Thus, our project aims to provide a *Philological Workstation* (or Philological Application) for European Libraries. Its kernel is therefore a software tool which allows historians to read the manuscripts, enter transcription, enter information (or critical apparatus or notes) on manuscripts linked to images and transcriptions of individual pages of those manuscripts, with especially close linkage of image and transcription. A first project has produced an effective PC-based philological workstation (using Visual Basic 4.0, Access 7.0 and the international standards SGML [1] and HyTime [2]).

It is obvious that a higher feature of this software should enable philologists and librarians to use the new philological workstation alongside an INTERNET server. Consequently, the new project should produce a "Philological Web Server" including the following fonctionnalités for the philologists:

- send requests on manuscript contents ;
- access and visualize manuscript transcriptions and images ;
- make works on manuscripts : transcriptions, annotations, hypermedia navigation ;
- links between part of image and the corresponding part of text.

After an introduction on the local philological workstation, a review of the conventional CGI-based and Java-based approaches for database access and application access is given. Next, the ActiveX solution is presented with experimental design of ActiveX controls (ex OLE controls) for database application with Visual Basic 5.0. Finally, the final architecture of our Remote Philological Application is presented and directions of future development are highlighted.

2 The Philological Workstation

The project entitled BAMBI (*Better Access to Manuscripts and Browsing of Images*) is part of LIBRARIES programs (1995-1997) and aims at designing a modern interactive tool for consulting and working on manuscripts [5], [6]. The different european partners of the BAMBI project are ACTA S.p.a (Florence), CNR (Consiglio Nazionale della Ricerche - Istituto di Linguistiche Computazionale di Pisa), BNR (Biblioteca Nazionale Centrale V.E.II di Roma), MPI (Max Planck Institut für Rechtsgeschichte (Frankfurt)), CPR (Consorzio Pisa Ricerche), and LISI.

BAMBI is an electronic desktop on which it is possible to deal contextually with the original handwritten document and its transcription (see figure 1). This contextuality means that students always have all the data they need to examine at their fingertips: the document together with its interpretation.

Figure 1: The BAMBI Workstation

The manuscript and its transcription are *linked* in BAMBI at the level of the page or, more frequently, the word, through an automatic system for segmentation of the image and searching for matches: a double click on

the transcribed word is all that is required to find the original word on the manuscript. And vice versa. We recall that the *transcription* of a manuscript is a process which leads to noting the full transcript of a given language by means of the system of signs of a conversion language. In the era under consideration, the early and high Middle Ages, abbreviations can be divided into the following types : syllabic abbreviation (omission and elision of letters), abbreviation by suspension (an example is provided by the names of jurists : ac. = Accurcius, bul. = Bulgarus, ...), abbreviation by contraction (which have endings written on the line), and the use of special signs.

The electronic form of the data processed by BAMBI offers many other advantages: the digitized image of the manuscript can be enlarged or reduced to the desired level of detail; it is possible to arrange documents on the screen in a flexible way; the transcription made does not remain an inert text of mere output but pilots the generation of indexes of forms that can be examined on line and directly in the open document.

Thus the data acquired can be examined in detail in the ambit of the manuscript or evaluated through other manuscripts by means of the sophisticated relational database on which BAMBI relies. The same database also drives the search and cataloguing of manuscripts on the basis of various criteria, saves and classifies the annotations that the user adds to the transcription, supports *browsing* between pages and between documents, manages rights of access to the application, etc.

The BAMBI project is part of a system devised by Andrea Bozzi of the Institute of Computational Linguistics at the CNR in Pisa and realized in prototype form under the name *Multimodular Philological Workstation*. The aim of the Philological Workstation is to automate certain phases in the work of editing handwritten texts [4]. BAMBI pursues the same objective and makes this software available for reading digital images of manuscript documents, transcribing the text contained in them and establishing automatic links between the text of the transcription and the images themselves.

The aim of our work is to allow philologists to work with this workstation alongside the Internet network.

3 Approaches for Database Remote Access

3.1 CGI-based Approach for Database Access

A typical database application on the Web today consists of three components:

- a Web Server,
- a HTTP Server with a CGI program,
- a Database Server.

A database query is initiated by sending an user request (using HTML forms) from the Web client to the HTTP server. Upon receiving the user request, the HTTP server invokes the CGI (Common Gateway Interface) program to assemble user input data into database-specific SQL statements, and send them to the database server for processing. The query results will be returned by the database server to the CGI program, and then passed to the Web client through the HTTP server. Since CGI is the de facto standard for interfacing HTTP server with external applications, this database access scheme becomes the most commonly used method in today's Web world.

Despite of its simplicity and wide acceptance, there are some common problems associated with the CGI-based approach for our Philological Application :

- the communication between a client and the database application must always go through the HTTP server in the middle, which likely becomes a bottleneck if there is a large number of users accessing the HTTP simultaneously,
- the lack of efficiency and transaction support in a CGI-based access scheme,
- the lack of user access control (the philological application access has to be reduced to specialized groups of users),
- the lack of presentation graphics, due to the limitation of HTML. The presentation graphics is very important for manuscript images.

3.2 Java-based Approach for Database Access

In general, there are two methods to construct a Java applet [8] as a client for accessing a remote database server. One is to use a two-tier architecture in which all client functions are implemented entirely in Java [10]. The other one is to use a three-tier architecture, in which a standalone Java server is used as a gateway, passing the request and response messages between the applet and the remote database server.

The former requires extensive programming effort because a Java client has to be implemented at the protocol level for vendor-specific databases. The latter is easier to implement, because the gateway server can be created as a standalone Java application with native client library functions wrapped in Java classes. It communicates as a server with the Java applet at one end in user-defined protocols, while accessing the database server at the other end in native client/server protocols.

Despite of its safety and portability, Java is currently slow and this is a disaster for manuscript image manipulation. Moreover, there isn't existing applets and therefore the use of Java requires professional programmers. In some cases, this approach may also restrict the freedom of developers because no other APIs are given for database access.

As there are many ActiveX controls [7] now available for Windows 95/NT and Visual Basic 5.0 offers a very easily support for creating ActiveX objects, we have chosen this approach for developing the remote philological application.

In addition, the local philological application has been developed in Visual Basic 4.0 and the ActiveX approach associated with Visual Basic 5.0 allow us to reuse the previous code developed. Component software, building application from reusable parts, is an attractive idea. The fundamental notion is to create an application by plugging software components into some kinds of container. Those components may be specifically written for this application or, better yet, reused from other project. The goal is to create more reliable applications more quickly, and to spend less money doing it. Today, Visual Basic is perhaps the most common example of a container, and the components it uses are commonly loaded when needed from a machine's local disk or a file server.

4 ActiveX-based Approach for Application Remote Access

With ActiveX controls, we use four components :

- A web Server ;
- A Client Server ;
- ActiveX Server ;

- A Database Server or Information System.

ActiveX Servers (ex OLE Servers) are used for running application through INTERNET. The applications are executed on the ActiveX Server, playing the role of relay between Web server (HTTP) and the Database or the Information System (that is the Philological Application). ActiveX Server allows to create non visual management objects which will integrate the management rules and the treatments associated to data. The ActiveX Server access can now be performed asynchronously (despite of OLE servers).

ActiveX Server created will use DCOM model. DCOM ensures the communication between ActiveX Servers in a transparent manner for the developer (figure 2).

Figure 2: Application integration via ActiveX controls

Like Java applets, ActiveX controls are self-contained pieces of functionality that run inside some kind of container. And as with Java applets, a web browser is a good choice for that container, allowing ActiveX controls to be embedded in web pages and downloaded on demand.

5 The Philological Web Application

5.1 Open Architecture

For designing the Philological Web Application, we propose an open architecture (figure 3) composed of one or several servers (the Philological server (Web server and Database server) contains the Philological application) and client workstations, each having specific access according to the user profile (philologist, librarian, scholar, ordinary user). We have identified three kinds of access:

- *Philological Application information access*: is limited to the presentation of the philological application functionalities available on the Web. This access will be performed with the creation of a simple HTML page.
- *Partial access*: permits only the consulting of manuscripts stored in the manuscript database (images or transcriptions).
- *Full access*: allows clients to use all the functionalities of the Philological Application.

For the two previous access, users have to subscribe to philological user group according to their function and needs. It requires the creation of user management and security access management.

In the following paragraph, we describe how to perform the full access to the Philological Application and how to allow remote users, that is philologists, to use this application. The Philological Server and Web Client is based on the architecture presented in paragraph 4 (c.f. figure 2).

Figure 3: Philological Web Application Architecture

5.2 Design of the Philological Web Application

In this section, we demonstrate the methodology for developing our Web application.

- *Application design:* Our work consists in preserving the maximum of elements of an existing application when exporting it to the Web. For designing this web application, we propose to use MethodF, a method based on OMT [11] and OOSE (Object Oriented Software Engineering). OMT is used for its formalism and OOSE for its techniques of " Utilisation Case " (scripts) which allows to describe the processes in order to identify the different components of the application. The purpose of the analysis phase is to construct an analysis model giving the type of identified objects and an interface model giving the interactions between these objects and the different actors.
- *Construction of ActiveX controls:* The design of the Philological Web application is based on the reuse of the code of the local version. Reusing this code has consisted in transforming the main forms of the Visual Basic application in ActiveX controls which will be directly loaded in a HTML page. Visual Basic 5.0 (Control Creation Edition) allows ActiveX controls to be produced easily. Visual Basic 5.0 proposes a type of project integrating directly an object of User Control type (the basis of ActiveX control) and an assistant for preparing the creation of ActiveX control. The first step consists in choosing the attributes and standard methods available through the User Control. After, we can define the members (methods, attributes or events) which will be implemented later.

For example, the control related to the visualization of a manuscript page with its informations (transcription, annotations, links, index verborum) (presented in figure 1 of paragraph 2) is the more important one and uses specialized DLL. It is composed of seven zones: the manuscript image, the transcription, the links, the Verborum, the annotations and the icon zones. The associated ActiveX control is `ctlallmanu.ctl`. This control uses the elements of all modules. They are then incorporated in the project (figure 4).

An ActiveX control uses the services supplied by various files. These files are gathered in an other file named Cabinet file (`Pjword2.CAB`).

- *Generation of HTML pages:*

Figure 4: The ActiveX control for manuscript page

After the design of ActiveX controls, we have to incorporate them in a HTML page. As we have generated a Cabinet file Pjword2.CAB, we obtain a file "projectame.HTM". This file contains a call model of all ActiveX controls of the project. The following file contains the ActiveX control *ctlallmanu*:

```
<OBJECT
  classid="clsid:CA8ACAEA-C55E-11D0-9F35-0020AF430337"
  id=ctlallmanu
  codebase="Pjword2.CAB#version=1,0,0,0">
</OBJECT>
```

- *Remote database access*: The database access is performed using ODBC (Microsoft Open Database Connectivity). ODBC is a standard programming interface for application developers and database systems providers. The advantage of ODBC is that the application can be written to use the same set of function calls to interface with any data source. The source code of the application doesn't change whether it talks to Oracle or SQL Server. Our method to perform remote database access consists in replacing local database access by database access using ODBC mechanisms.

The access functions to the database are not envisaged systematically, and especially automatically. We wish to extract all the database accesses actually present in each module. The preservation of these database accesses without any modification allows a complete reuse of the code, but this preservation induces a major drawback. The clients have to be equipped with ODBC for acceding to the database. This feature represents a major constraint. Moreover, if the components accede to the database, they will be very specialized and reusable with difficulty in other configurations. It will be more interesting to use the CGI interface or the ISAPI interface for performing the database accesses. But, it will increase the development time and the development cost of the ActiveX controls to include in the HTML pages.

6 Conclusion

In this paper, we have demonstrated a method for designing a Philological Application on the Web using ActiveX controls. Compared with the Java-based approach for database application, this approach is more easy to develop, fast and allows us to reuse the code of the Local Philological Application. Following the proposed methodology, future development will be conducted in order to allow remote clients to make cooperative work.

In particular, we can quote:

- asynchronous cooperative work (annotations, private news, manuscript exchange,...) ;
- synchronous cooperative work (voice conference, information exchange, ...).

References

- [1] AFNOR. *Langage normalis de balisage gnralis : SGML*. Norme NF EN 28879, Systmes bureautiques, Traitement de l'information, Dcember 1993, Afnor
- [2] AFNOR. *Langage de structuration hypermdia/vnementiel : HyTime*. Norme NF ISO/CEI 10744, Technologies de l'information, Dcembre 1994, Afnor
- [3] H. Berghel. *The client's side of World Wide Web*. Communications of the ACM. Vol.39, N1, january 1996, pp.30-40
- [4] A. Bozzi. *Towards a Philological Workstation*. RISSH, 29 (1993), Lige, pp. 33-49
- [5] S. Calabretto, A. Sappupo and F. Tariffi. *Modules and functions in the BAMBI software system*. European Libraries Project Lib-3114 Report. Deliverables 4-5. September 10th 1996. 56p.
- [6] S. Calabretto and J.M. Pinon. *Modelling of a medieval manuscript database with HyTime*. Proceedings of the ICC/IFIP Conference on Electronic Publishing'97. New Models and Opportunities. 14-16 April 1997 - The University of Kent at Canterbury. To appear in Chapman and Hall Eds
- [7] T. Coombs, J. Coombs and D. Brewer. *ActiveX source book. Build an ActiveX-Based Web Site*. Ed. John Wiley and Sons, INC. 1996
- [8] N. N. Duan and B. Atlantic. *Distributed Database Access in a Corporate Environnement Using Java*. Fifth International World Wide Web Conference, May 6-10, 1996, Paris, France
- [9] S. Hadjiefthymiades and D. Martakos. *A generic Framework for the Deployment of Structured Databases on the WWW*. Fifth International World Wide Web Conference, May 6-10, 1996, Paris, France
- [10] T. Ritchey. *Programming in Java-Beta 2*. Indianapolis, New Riders Publishing, 1995
- [11] J. Rumbaugh. *Modlisation et conception orientes objet*. Edition MASSON. 1994. 515 p.
- [12] S. Zimmerman and C.L.T. Brown. *Web Site Construction Kit for Windows 95*. Sams Net Editor. 1996