

DOCUMENT RESUME

ED 453 802

IR 020 648

AUTHOR Strunz, Bob; Waters, Mairead
TITLE The ELISE II Project: A Digital Image Library for Europe.
PUB DATE 1999-00-00
NOTE 6p.; In EUNIS '99: Information Technology Shaping European Universities. Proceedings of the European University Information Systems International Conference (5th, Espoo, Finland, June 7-9, 1999); see IR 020 639.
AVAILABLE FROM For full text:
<http://www.hut.fi/events/eunis99/Csession/C31.htm>; Track C:
<http://www.hut.fi/events/eunis99/TrackC.html>.
PUB TYPE Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer System Design; *Electronic Libraries; Foreign Countries; *Information Networks; *Information Services; *Information Systems; Library Networks; Library Services; Online Systems
IDENTIFIERS Europe; Java Programming Language

ABSTRACT

This paper describes the progress made under the ELISE II electronic image library project from a technical standpoint. The ELISE II project is a European-wide initiative that aims to provide a comprehensive electronic image library service for Europe. It is funded under the European Commission, DG XIII-E, Telematics for Libraries Initiative. The ELISE system is composed of a JAVA client which runs on the user's Web browser and allows access to the system; a broker, based on JAVA servlet technology; and a set of distributed, non-compatible image databases located all over Europe. This paper discusses the major design emphasis in the development of the ELISE system; the overall architecture of the ELISE II system; the use of the Z39.50 protocol and Dublin Core metadata standard; HUNT and RTE databases mappings; implementation of the GUI client; and implementation of the broker. (AEF)

The ELISE II Project, A Digital Image Library for Europe

Bob Strunz, Mairead Waters

University of Limerick, Ireland

Abstract

This paper describes the progress made under the ELISE II electronic image library project from a technical standpoint. The Elise II project is a European-wide initiative which aims to provide a comprehensive electronic image library service for Europe. It is funded under the European Commission, DG XIII-E, Telematics for Libraries' initiative.

Introduction

The ELISE project has developed a comprehensive JAVA based demonstration system which provides controlled access to images and associated textual information. Eight of the partners are providing a variety of image data to the imagebank including still images, and streaming video. At the time of writing, there are over 30,000 images available on the set of demonstration databases.

The system is composed of four component parts :

- A JAVA client which runs on the users web-browser and allows access to the system.
- A Broker, based on JAVA servlet technology
- A set of distributed, non-compatible image databases located all over Europe.

The major design emphasis in the development of the ELISE system was on flexibility with the ability to easily add additional databases to the system being of paramount importance. It was decided to utilise a combination of tried and tested library technology to integrate the databases so that the design team could concentrate their efforts on designing the user interface and broker architecture which relies on less mature technology.

Overall System Architecture

The architecture of the ELISE II system is as shown in figure 1, the JAVA GUI client communicates with the broker via the web, the broker, which is a JAVA servlet, communicates with the image databases. Two separate protocols are used for this communication, the searching is done using the Z39.50 protocol which allows us to perform simultaneous searches on multiple databases. The images themselves are returned using the HTTPS protocol via a standard web server such as Apache™. This communication is at present, in clear, however experiments have been done to utilise SSL to encrypt the data.

The architecture of the system is such that the component parts of it may be widely distributed, this is in fact the case, in the current version of the system, the Broker servlet runs on a machine in the UK and the Database Image Server (DBIS) is located in Tilburg in the Netherlands. With the benefit of hindsight, the development of the system in this fashion, while it made perfect sense at the time, has resulted in performance difficulties. The network links between the institutions are insufficiently fast to provide the user with the performance one usually expects from a web based system.

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

ED 453 802

IR020648

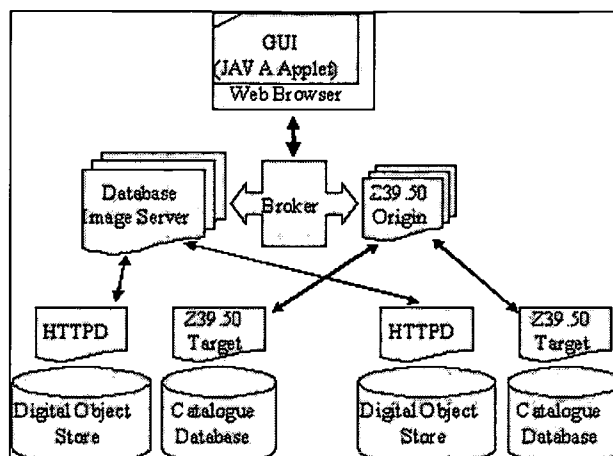


Figure 1 Overall ELISE System Architecture

The reason for this is that all the data delivered to the user must pass through the broker for security reasons. By doing this, it is possible to secure the online databases from external interference as they can only be accessed via the broker, despite the fact that they are available over the web.

If a user in Ireland, retrieves an image from Ireland, the image travels from Ireland to the UK to The Netherlands, back to the UK and then back to the user in Ireland. The simple solution to this is to implement local brokers at the main sites in each country, this has proved effective in the UK and will be implemented in Ireland in the near future.

Z39.50 and Dublin Core

The use of the Z39.50 protocol and Dublin core was a pragmatic decision based on the need for a straightforward means of adding multiple databases at some time in the future.

The Dublin Core metadata standard is a simple yet effective element set for describing a wide range of networked resources. The Dublin Core standard comprises fifteen elements, the semantics of which have been established by an international group of professionals from librarianship, computer science and the museum community.

The Dublin core specifies (but does not dictate) a standard set of database fields onto which new databases may be mapped. For example, in the case of the RTE database the RTE field "Photographer" is mapped onto the Dublin core field "Creator". Using the Dublin core in this way, allows us to provide a potential database provider with a standard set of fields which he can map to his own database. Table 1 shows an example of how this was accomplished for the HUNT and RTE databases.

Dublin Core Field	HUNT Field	RTE Field
Title	Title	Title
Creator		Photographer
Subject	Associations	Keywords AND Personality
Description	Brief Description	Description
Publisher	"HUNT"	"RTE"
Date	Date	Date
Type	"IMAGE"	"IMAGE"
Format	"JPEG" OR "GIF"	"JPEG" OR "GIF"
Identifier	CD_Num/Image_Num	REF_Num/Image_Num
Source		Source
Coverage		Place
Rights		Copyright Owner

Table 1 Dublin Core to HUNT and RTE Database Mappings

In order to format search queries in a standard fashion, the Z39.50 library database search protocol was used, this is layered on top of a more generic search language which is similar to SQL. There are therefore various possibilities for searching a new database. If a standard Z39.50 target is available for the database, then it is automatically searchable. If not, then it is possible to write a thin layer which translates the generic search language into a form which is understood by the new database.

The advantage of Z39.50 over standard web search protocols is that it is stateful, that is to say that it is capable of maintaining a record of previous sessions, and therefore allows a user to take up where he left off. It is also unique in that it supports parallel searching of multiple databases, this is of particular interest in the case of ELISE as there are potentially large numbers of databases accessible to the system and the search protocol is capable of searching all of them, or a subset of them simultaneously.

The architectural model that Z39.50 uses is as follows; a server communicates with one or more databases containing records. Associated with each database are a set of indices that can be used for searching. Thus Z39.50 provides an abstract view of the databases and allows us to deal with logical entities based on the kind of information that is stored in them, while ignoring the minutiae of specific database implementations.

The Z39.50 origin standard allows the user to transmit a search to the server (SEARCH request) this produces a result set which is maintained on the server; the result of a search which is transmitted to the client is a report of the number of records comprising the result set, result sets can be manipulated by subsequent searches.

Records from the result set can be subsequently retrieved by the client using PRESENT requests. The PRESENT request offers elaborate options for controlling the contents and format of the records that are returned. The PRESENT request indicates specifically which records from the result set are to be retrieved.

The standard supports the transfer of large amounts of data, however it is not a particularly efficient way of doing this and therefore in the case of the ELISE system, the PRESENT request returns a reference to the broker which allows it to retrieve images via a standard web server.

Implementation of the GUI Client

When this project was initially conceived, JAVA technology was emerging as an obvious choice for client side applications designed to run in a web environment. JAVA is intended to provide the user with a programming language which is truly portable.

The JAVA language does this by defining a virtual machine which is, itself a computing engine. In order to run JAVA on a new machine, in theory all one has to do is to write a JAVA virtual machine for the new physical machine. The JAVA environment provides the user with a rich set of tools for developing web-based information services. The idea of JAVA is excellent and it should provide the user with a stable and portable computing platform. The reality is unfortunately somewhat different.

The failure of software manufacturers to provide a fully standardised run-time environment for JAVA makes the development of truly portable JAVA applets something of a challenge. The development team in the ELISE project has had an extremely difficult task writing the client for the system. The decision to use JAVA is regretted by all of the development team, the effort which was required was far in excess of what was budgeted for.

Most of the problems encountered by the developers have been related to instability and incompatibilities in the Netscape and Internet Explorer JAVA platforms, though a secondary consideration is that JAVA programming is a very marketable skill and the project had great difficulty in retaining programmers for any length of time.

The net result of this is that the JAVA code which was developed for the GUI has had to be written at a lower level than was initially expected and large amounts of run-time library code has to be downloaded along with the applet in order to allow it to function in both Netscape and Microsoft browser environments. A partial solution to this is the use of the Java Archive facility JAR which allows us to compress the applet into a smaller form for downloading.

A further solution may be the use of a browser plug-in which will be downloaded only once but will provide the platform dependant functionality to the user. This solution is not desirable as it defeats the purpose of using JAVA to some extent. Using a special plug-in means that the system is no longer truly platform independent and could have been developed more easily as a browser plug-in.

Implementation of the broker

The broker is implemented using JAVA™ servlet technology, this is effectively server-side JAVA™ which allows the user to develop portable applications which are ideal for implementing middleware between databases and GUI clients. It was found that the development of this broker was a much easier task than the development of the GUI.

The reasons for this stability seem to lie in the fact that the broker is not doing any graphics or mouse-event handling, it is simply brokering information from the various component parts of the system. In such an application, the use of servlet technology is an ideal solution and provides the developer with a very powerful, flexible and stable means of gluing a distributed system together.

Conclusions

In conclusion, we believe that the JAVA runtime environment is still not stable enough to easily develop complex applets. The competition between manufacturers has led to a situation where an appearance of cooperation between them is actually concealing efforts to kill off competing products. It is to be hoped that this situation will change as the technology is excellent in principle.

The excellence of the basic technology is demonstrated by the broker side of the system which uses a servlet for its implementation. In this case the lack of graphics and mouse event handling made the design and implementation of the system much more straightforward. In a situation where additional functionality has to be added to a web-server which supports servlets, it is a very credible alternative to using cgi scripts.

Under the current circumstances, educators should carefully consider the implications of using this technology to deliver content to desktops in their institutions.

Acknowledgements

We would like to thank our partners in the ELISE II project for their assistance in producing this paper.

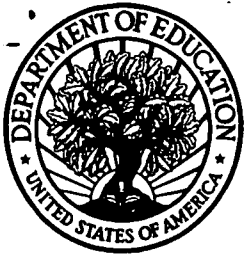
Bibliography

The ELISE Web Page <http://severn.dmu.ac.uk/elise/>

Address

Department of Information Technology,
University of Limerick,
Limerick,
Ireland.

TEL + 353-61-202399
FAX + 353-61-330316



U.S. Department of Education
Office of Educational Research and Improvement (OERI)
National Library of Education (NLE)
Educational Resources Information Center (ERIC)



NOTICE

REPRODUCTION BASIS



This document is covered by a signed "Reproduction Release (Blanket)" form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.



This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket").