A DISTRIBUTED CLIENT-SERVER INFORMATION SYSTEM FOR ARCHITECTURE

CHRIS YEUNG
Department of Electrical and Electronic Engineering,
The University of Hong Kong

AND

JOHN BRADFORD, ERIC SO, GUYYVER CHENG
Department of Architecture,
The University of Hong Kong

Abstract. We describe in this paper the implementation of an internet-based information system for the organization and retrieval of 3-D architectural structures. Each structure may consist of sub-structures down to the level of building blocks like pillars, roof-tops, walls, windows and doors etc.. Along with each structure, a set of characteristics is attached. These characteristics can be physical such as geometry styles, finish, and materials; or environmental such as climate, contour, and region; or historical such as era and religion. An application can thus query the information system by specifying any combination of these characteristics.

1. Introduction

An information system for architecture allows a user to retrieve, store and manipulate data models which are 3-dimensional objects. The representation and the access mechanism of the models are vital for the study of architecture design. To make the system more useful, it must provide every model the ability to associate with attributes, characteristics, and features relevant to architecture design theory.

With the explosive growth of the internet and world-wide-web technologies, it is feasible to allow online communication among people for the exchange of architecture ideas through platform independent web-browsers. Such approach has the advantage over conventional customized CAD environment by encouraging collaborative team works all over the world.

The purpose of this research project is to develop a client-server search engine for accessing architecture information. The engine interfaces with
databases storing the properties and the physical organization of architecture models. These databases are in fact distributed over the internet. While a particular internet site may contain databases for one or more Chinese temples, another may store roman empire artifacts. The engine has the ability to broadcast a query from a client application to all known servers for looking up models satisfying a user’s request.

To illustrate the usefulness of the search engine, we have implemented a site offering several historical Chinese temples. The implementation includes the physical models, their properties, and a sample client application. The way that we associate properties to models is based on references (林會成, 1989; 鄭海超, 何金泉, 李惠玲, 何翠媚, 1983).

2. A Sample Client Application

Before discussing the design of our search engine, it would be better to introduce a sample client application which is essentially a Java-enabled web page. The application allows an end user to search and browse general-purpose 3-D models. All models described in the Virtual Reality Modeling Language (VRML) can be manipulated within the web page. Figure 1 below depicts the initial layout of the page loaded through a web browser.

![Figure 1 A sample web-page application](image)

The page interacts with a user through the lower two panels. The right panel, appearing in blank grey for now, will display a selected model. The left panel is a window-like file manager folder written as a Java applet. Once the page is loaded, it initiates a communication link with the search engine. The engine then immediately seeks for those sites that it knows of and reports to the Java applet. A user, however, has the ability to direct the engine to search for other sites. As
in Figure 1, the three sites first found are indy14, hkarch6, and hku. These sites are
icon next to a site implies that relevant databases likely exist.
Figure 2 shows the web page after expanding the folder hkuarch6, in which two
databases: lopan.Arch and guangsheng.Arch have been retrieved. With the
lopan.Arch folder highlighted and the BROWSE button activated, a 3-dimensional
model of the complete Lo Pan temple (a famous ancient Chinese architect) will
be brought into scene on the right hand side panel.

![Figure 2 Browsing architecture models](image)

The user will be able to examine the model and even virtually take a tour
inside. Indeed, the model can be further analyzed in detail part by part. Figure 3
demonstrates the case when a one-sided brick wall is selected.
The significant issue here is that the user will always be presented with the most updated database information, even within the same session. This would be the case when there is another client application somewhere which has the authority to modify the databases. Such dynamic nature of obtaining online models strengthens collaborative team work despite geographical diversity.

The web-page application also lets a user specify other possible sites which would be included in the search list. In addition, a query can be generated in the form of a combination of properties and the engine will search all known sites to retrieve matching models and parts. For example, the user may be interested in those which are related to feng-shui or yin-yang.

3. ArchServer - The Search Engine

The search engine, codename ArchServer, is a hidden process running in one or more web sites. The overall system is fully distributed in the sense that the engines, the databases, and the actual physical architecture models may all reside
in unrelated sites. It is possible that a database may include only the hierarchical information and localized properties, but not the 3-D geometric models. In other words, the same architecture model can be characterized differently by different people. This principle is illustrated in Figure 4 where site A contains both databases and physical models, site C has only the later, and whereas site B is just a database description referring to models in the other two.

![Diagram](image_url)

**Figure 4** Distributed search engine

The engine communicates with its client applications through a protocol. Thus the underlying database structure is totally transparent to clients. Examples of the protocol which the engine delivers to a client are: tree hierarchies of models; lists of properties; universal resource locators (url) of models and parts; introduction of models; and concurrent client user information and dialogs etc.. A client through the protocol, may request the engine to search new sites, specify search constraints in terms of property keywords, broadcast its browsing status to other clients, and record the user’s comment on a model into the database. In response to a client’s keyword search, the engine will pack the request into a sequence of relational SQL statements and forward it to the databases concerned.

The engine, however, makes very little assumption on the design and the specification of the databases. Through the Java DataBase Connectivity interface, it can effectively access any relational database management system.

To reduce searching at start up time, the engine only looks up those databases with the .Arch suffix as a site may contain databases for other purposes.

Within a database of our interest, a collection of tables of data is defined. The engine assumes that a convention has been applied to the names of the tables. Suppose a table is named *abc*, then another table with the name *abc.xyz* will be considered as a subordinate entity to the former. This convention facilitates the hierarchical specification of an architecture model which essentially comprises of sub-models. For example, the existence of the two database tables *wall.window* and *window.frame* implies that the frame for the window is
attaching to the wall.

4. Database Organization

An architecture model can be hierarchically decomposed into different sub-models to the level of building blocks like pillars, roofs, walls, windows and doors. Figure 5 shows an example of a logical view of the database tables following a particular hierarchy. This example is derived from reference (張英超, 何金泉, 李惠玲, 何翠華, 1983).

![Figure 5 Logical view of database and tables](image)

While Figure 5 presents a logical view of the database organization, the physical structure of the database tables is shown in Figure 6. As discussed earlier, each table corresponds to a model or sub-model and its name implies the level in the design hierarchy. Every table contains at least two fields in pair: property and value. A model is freely assigned or grown with these pairs.

In the table Gable.Part1, which is a part of a gable for a roof of the house, the properties at present include putoonghua phonetic, the attribute feng-shui, the internet address of the site, a VRML-version model, an Autocad version, and the author’s name. As we can see, the 3-D model can be located anywhere unrelated to the database. Moreover, while one database attempts to classify the parts by cultural or historic properties, another database elsewhere may re-classify them.
in climatic, continental or oceanic, and longitudinal or latitudinal views.

5. Conclusion

In this paper, we have proposed and implemented an information system over the internet for accessing and browsing architecture models. The system is highly flexible and extendible as the overall design is platform independent with distributed database support. We have demonstrated the system through a Javatized web-page application communicating to a back-end search engine.

The system could be found useful to designers in exchanging ideas online and to students in learning architecture models with theory. We intend to construct more sites with Chinese artifacts so that people world-wide can appreciate one important ancient architecture style.

Reference

北京市建築設計研究院編寫組：張人琦，鄭麗萍，楊崇敏，趙春青：1992，漢英建築工程詞典，中國建築工業出版社
林會承：1989，（台灣）傳統建築手冊：形式與作法編，藝術家出版社
鄭海超，何金泉，李惠玲，何翠娟：1983，香港島區之華人廟宇，香港大學亞洲研究中心
Fong, G.: 1980, An Introduction to Chinese Geomancy, University of New South Wales, Sydney, Australia
Lip, E.: 1979, Chinese Geomancy, Times Books International