Design and Implementation of Cost Effective MIS for Universities

Sastry MKS
Faculty of Engineering, UWI, St. Augustine, TRINIDAD, W.I.

Abstract
Design, development and implementation of a three layer cost effective Management Information System (MIS) for typical university environment are presented. Several aspects which are specific for any typical university environment like - nature of diversified data items, continuous growth of data, different categories of users, demand for access, user expectations and huge quantities of information etc are considered for the design and development process. The approach suggested in this paper combines the strengths of a commercial database product and open license technologies. The need for using a standard and commercial database product is emphasized. Two client applications are developed using two different data abstraction technologies which work independently on the common database backbone. The developed model of MIS is tested using the data of a joint degree program offered by two different universities and the sample results are presented.

Keywords
Management Information System, Database Design, Pro*C++, PHP, API

1. Introduction
Universities, no matter small or big are composed of various faculties, facilities, departments, students, employees etc. Every department for sure, would have its own data and information in an organized form due to the penetration of information technology. Volumes of information will be generated and transmitted within the departments on a continuous basis. In most cases, information in the form of data requires to be shared, updated and modified for meeting the day to day functional requirements. Hence, there is a great need to automate the university functional activities. The range of tools for managing data is usually ranging from a personal, desktop and network based within a department.

For managing wide-ranging functions, universities are tending to use well-known commercial MIS products. There are several such commercial products are already available which are customized to large universities. However, procurement of such products, training the users, product customizing for specific requirement, maintenance over a longtime etc are proved to be quite expensive and yet, the product is not owned by the organization. The licensing system of known and popular commercial products may appear competitive initially; however it would be expensive over a long time. In such cases, universities have to be at the mercy of the product vendors for operation, maintenance and especially for customization following changes in university systems. Even software companies and suppliers offering such solutions may not be able to support their own older installations due to technological developments, intentions of phasing out older versions, competition in the market, ease of management and their own survival etc.

Keeping in view of these aspects, it is always recommended that universities having their own designed, developed and operated MIS infrastructure. This saves the universities from spending spree in the first place, which is essential especially in case of limited funding conditions. Customization is always possible and the product is owned. Even, if the university decides to use a commercial MIS, the necessary
data is ready in required formats, which makes transition easy. In the process, university will gain rich experience in this area and obviously would be able to help other organizations in capacity building in the use of information technology. Hence, design and development of MIS for own use is justified.

1. Requirement Analysis

The functional environment of any university makes the design of required MIS not only different from a general purpose MIS. If not designed properly, it may lead to complex working conditions, loss of data, time and money. The nature of data items is different from that of others like process, industry or any real-time system. Some of the general design requirements of MIS for an academic environment are

- Large number of active users.
- Hi-availability expectations of the system
- Large active data set
- No requirements of complex, scientific calculations on the data.
- No requirements bulk data transfer
- No special display requirements
- High level system/data security requirements

An exhaustive specification study required to be done for designing any MIS. This depends on the extent of MIS coverage, needs of specific university and also available resources. This is out of scope of this paper and hence not discussed.

3. Technologies and Design Considerations

Client/Server architecture has changed the system design technologies to a great extent. Design and deployment of three layer or even multi layer solutions has become a standard practice. The MIS model developed in this work has three layers which are discussed below.

3.1 Design considerations of Database Layer

The design of database layer is the most important factor in any MIS. The overall performance of MIS depends mainly on the performance of database backbone. A standard database is required to host and to manage huge volumes of data. Though license free/open source databases are available (like Mysql) the advanced features of security and ease of operation may not be available. On the other hand, a commercial database can be very effective and seems an ideal solution, but can be very expensive depending up on the usage licenses. Using a standard and commercial database for hosting the database appears to be an inevitable choice. When a specific commercial database server is the chosen, the best practice is to use the technologies that are supported by the same product in other layers. Most commercial databases can process the information that is normally done by APIs. Hence, it is important to transfer all such processing load to database layer. This not only reduces the burden on the API layer but also increase the overall performance. The commercial database product Oracle is chosen in this work.

There are several stages in design and successful deployment process of any MIS (Sastry 2005). The first stage of design process is identification of a team with necessary skill sets. The team members should be assisted by an experienced academician, who also has the knowledge of administrative functions of university systems. The next stage is finding the functional specifications and requirements of MIS, which is critical for survival of whole process. Designing the individual databases based on the specifications (Sastry, M.K.S. and Manohar, A, 2003) is the next stage. Usually, the most common databases in a university environment include: Student Information System, Program Information System, Employee Information System, Accounting Information System, Asset and Facility Management System, and User Information System. Once the core components are identified and designed properly then the
next stage is identification of specifications for data and web servers, operating systems, database products, design and development tools, platforms and possible technologies for this project. A sample of data connection and relational strategy that is used in the database layer of this work is shown in Figure 1.

The data is kept in different clusters and layers to make it not only robust but also faster. Each data item (like student) will have obvious relations with the other data (like program). Oracle 10g is used for hosting this database. The design automatically ensures such relations and efficiency. Some of the standard industry practices that are used in this work are:

- Entire data is divided into master data and transaction data. Master data contains high level definitions, rules and references. Hierarchical relations between data items are specified in the design itself.
- Security rules and governing policies on computational procedures are also part of the design schema.
- Triggers are kept for vital data checks and updates in a separate data layer.
- Partition techniques are used at table level.

The details of the data resulting out of MIS, summarized data and the concepts of data warehousing are out of scope of this paper and hence not discussed (Shi, D, Lee Y. et al, 2001). Core and common information systems, functional requirements of information, decision making process, changing program structures, course numbers and contents, typical growth of data over years, different types of users, types of access, costs of technologies, size of deployment etc are some of few aspects which influenced the proposed design. On top of the database layer, two client models have been developed which are discussed in sections 4 and 5.
3.2 Design Considerations of Data Abstraction Layer

Different technologies are now available for implementing the data abstraction layer of the system. Though many server-side scripting technologies are available, the Personal Home Pages (PHP) is considered for the present discussion. Other technologies like, Active Server Pages (ASP) and Java Server Pages (JSP) are not favored as ASP is Microsoft specific and JSP is a heavy layer. Open Database Connectivity (ODBC) would be a better choice if database product is to be changed later. In this paper, two different APIs are developed using different technologies which are built on Oracle Call Interface (OCI). OCI is specifically chosen due to the following reasons.

a) OCI is Oracle specific technology and works directly with the core shell of the database server.
b) OCI is considered as the most efficient technology for interfacing the APIs with Oracle database
c) OCI supports different programming environments

3.3 Design Considerations of Application Layer

The visual interface to the user is provided by this layer. Ideally the client application should be

a) having the required performance
b) user friendly
c) using minimum resources on client machine
d) compatible with standard environments

Two different APIs are developed in this work, one using C++ and the other one with PHP. Both technologies are supported by major operating systems and offer good level of flexibility for developers. For the client side application design, there are numerous number of solutions are available. C++/ C#/ Java/PL-SQL are some of the frequently used solutions when compared to others due to compatibility issues. Ideally, user must be able to download the client application quickly and install it in few steps.

4. Designing MIS with a C++ API

A client application is developed based on above design to store, manipulate and to display the data items according to specifications. This application is designed using C++ language.

![Figure 2. Screen shots of windows based Client Application](image-url)
A pre-compiler known as Pro*C++ is used which actually uses Oracle Call Interface (OCI) technology. Pro*C++ is a part of Oracle application development suit. There are several distinct advantages of using Pro*C++ pre-compiler are
a) The database scripts can be mixed with C++ code and then pre-compiled.
b) The application can be developed in a modular fashion using several library components which can be dynamically linked depending up on the requirement.
c) It is relatively easier to maintain the code using this technology
d) The OCI layer is a very light layer and hence the application would be faster in execution
e) Pro*C++ is supported on all known operating systems by Oracle.

Some of screen shots of the application are shown in Figure 2. Though the application is very small in size, it still has to be downloaded and installed on every client machine. Once started, it will be extremely fast as data transfer is quick as it is built on OCI technology. This client application requires a client Oracle database to make a connection to the database server, which means this solution requires several user licenses of database. However, this approach is useful if the computational burden is heavy. The total system configuration with all the components including this application is shown in Figure 3.

5. Designing the API using PHP

The present problem does not require any special/dedicated client application in reality as computational burden is not much. A simple Internet browser can itself be used as a client to get the data and also for data purposes. This eliminates the installation and related cost and burden on client side. Also, client does not have to download any special application. In this case, PHP technology is used for server side scripting. With this design, the client can use any standard web browser such as Internet Explorer or Netscape Navigator which uses the common protocol known Hyper Text transfer Protocol (HTTP). Output layer is designed using Hyper Text Markup Language (HTML). The working mechanism of this approach is shown in Figure 4.
PHP technology does not require any license. It is quite stabilized and is used by many. Also, it offers very good flexibility to the programmers and the language is similar to C++. The major advantage is that it does not require any additional client software applications to be installed like in the previous case. Also, popular operating systems and databases support PHP technology. The user interface is provided in simple and friendly web pages. The home page of the MIS is shown in Figure 5.

![Home Page of the designed MIS using PHP](image_url)
This technology is recommended for handling large scale systems and huge volumes of data like university information systems. This solution is not only cheaper, but also web ready in the design philosophy itself. The total system configuration with all the components with this design strategy is shown in Figure 6.

![System configuration for MIS based on PHP](image)

**Figure 6. System configuration for MIS based on PHP**

From the above discussion, it can be seen that the design based on PHP is preferred (Odewahn, A. 1999) for the present problem due to the following reasons:

i) No special hardware or software requirements on client side

ii) No need of client database on client side

iii) No client program installation

Both APIs have been built on common database to study various aspects of data access technologies. To test this developed model, an existing data belonging to a joint degree program is used. The objective is to test the model with a typical situation, which is different from the regular university system. This joint degree program offers good challenge to the many known (commercial and popular) software packages. The student of this program is expected to have two IDs from two universities. Few courses will be done according to one university system and rest of the courses will be done according to partnering university. Hence, marking and grading procedures will be different. Calculations relating to GPA and CGPA are specific to this program. In addition, there are several special rules governing the student progression from one level to the other. The two client applications ported on to the common database model described in section 3. These applications work independently but are expected to have the same functionality. Both APIs have given the same output with satisfactory performance. The output of the MIS for a specific query is shown in Table 1.

**Table 1. Report generated corresponding to a specific query (Partially shown)**

<table>
<thead>
<tr>
<th>UTT-ID</th>
<th>UWI-ID</th>
<th>Last Name</th>
<th>First Name</th>
<th>Courses completed /Max</th>
<th>Credits Secured/Max</th>
<th>ENGL1001 (ENGL1303)</th>
<th>ENGL1002 (ITEC3372)</th>
</tr>
</thead>
<tbody>
<tr>
<td>104001540</td>
<td>5741171</td>
<td>Aaron</td>
<td>Roberts</td>
<td>7 (47)</td>
<td>15 (137)</td>
<td>B</td>
<td>F</td>
</tr>
<tr>
<td>105000412</td>
<td>2800159</td>
<td>Edward</td>
<td>Lisaene</td>
<td>18 (47)</td>
<td>37 (137)</td>
<td>C</td>
<td>F, B+</td>
</tr>
<tr>
<td>106211587</td>
<td>1800617</td>
<td>Lewis</td>
<td>Clifford</td>
<td>15 (47)</td>
<td>44 (137)</td>
<td>I</td>
<td>D</td>
</tr>
</tbody>
</table>

The actual student IDs and names have been changed to protect their identities. The output can be obtained in HTML/ MS-Word/ MS-Excel depending on the user settings. It is shown that only partially here and columns for other courses in the program are not shown. The complete listing includes all the courses. From the above, it can be understood that the students of this program have two student IDs
(both UWI and UTT). Also, courses have both old and new numbers. Also, the MIS has the information of all the grades obtained by a student in every course (see row 2 of Table 1). The proposed MIS is thus designed keeping in view of a general academic environment which is subjected to frequent changes.

6. Conclusion

The design strategies have been discussed for developing an MIS for the automation of university activities. A three layer MIS has been designed and developed. Two different APIs have been developed and tested on a common database and the results have been shown. It can be concluded that the combined use of open license technologies (PHP) and the commercial products (Oracle) will greatly reduce the overall cost of MIS without compromising the performance.

References:


Sastry, M.K.S. (2005), “Virtual information system for contingency planning and disaster recovery”, Proceedings of 18th APETT’s Annual Technical Conference on Engineering Infrastructure, Planning and Development for Disaster Preparedness and Mitigation, The University of the West Indies, Trinidad and Tobago, April, pp.249-256


Odewahn, A. (1999), Oracle Web Applications, O'Reilly & Associates, Sebastopol CA