15

Distributed ITEM for the future: moving towards client-server systems

Bill Davey
Department of Business Computing, RMIT University, Australia
124 LaTrobe Street, Melbourne, Victoria, Australia 3000
Phone: (613)-96605975 Fax: (613)-96605850
e-mail: Billd@RMIT.edu.au

Arthur Tatnall
Department of Business Information Systems, Victoria University of Technology, Melbourne, Australia
e-mail: arthur@cougar.vut.edu.au

Abstract

The pressures on ITEM systems include legacy systems being distributed among a diverse database. This diversity includes different hardware and software platforms in a variety of geographical situations. Client-server technologies offer a solution for the integration of information hardware and software sources into a system wide information resource. This paper relates experience of client-server technologies in the general literature, both positive and negative, to the needs of ITEM developers.

Keywords

Client-server, information technology, educational management

1 INTRODUCTION

Educational institutions typically have developed a number of information and communication systems that have been developed in isolation for specific purposes. Educational management is seeing the need for information from these sources to be gathered so as to aid in informed decision making. This scenario is typical of those that have been effectively solved using client-server technologies. Client-server architecture and applications can solve distributed data source problems, but introduce difficulties of

A. C. W. Fung et al. (eds.), *Information Technology in Educational Management for the Schools of the Future*

their own. The ITEM developer can gain from knowledge of problems that have arisen in other domains before choosing a client-server solution.

2 THE RISE OF CLIENT-SERVER COMPUTING

There have been negative reports in the computing press of some particular client-server implementations. These point out the disappointment of expectations raised by theoretical gains to be made from downsizing and distributing through client-server architectures. These reports seem to have made little impact on the move toward client-server (CS) when measured in terms of expenditure and number of new applications.

There are two trends that point out the importance of client-server architectures and the need for more training in the area:

- the rapid increase in products from solution developers, and
- the extent to which the growth in the use of client-server technologies in applications is not being met by trained graduates.

As evidence that industry is moving towards client-server, Marc Myers (1995) reports that "IBM is currently pouring 75 percent of its development efforts into client-server related projects." Myers also reports on a survey showing a rapid increase in other vendor products in the area. Julia King (1995), reports on a survey of 400 IS executives carried out by Deloitte Touche of New York. This showed that "50 percent of IS executives who expected benefits from client-server tools actually realised them. On average the IS budget is 16 percent higher in the companies where 25 percent of applications run on advanced client-server architectures. In 1994 43 percent of all applications ran under client-server compared with 27 percent in 1993." This type of result shows that client-server technology is expanding despite not meeting the expected cost and time savings. In fact the rapid expansion of CS caused the Deloitte survey to find "finding experienced people to design client-server systems remains a problem" and "last year 79 percent of IS managers had difficulty recruiting technical architects and distributed database experts." Clearly there is an established trend towards CS technologies in the general IS field. The reason for this trend and the gap between perceived benefits and practice is a little more difficult to explain.

3 WHAT IS CLIENT-SERVER COMPUTING

Client-server computing is based on the notion that some functions are best handled on a local basis while others are better done centrally. It is thus a blend of the central, timesharing approach, with distributed processing which emphasises local use. Client-server computing involves the use of a mixture of pcs and mainframes, typically connected by a wide area network. Application processing is shared between the pc clients and one or more servers. To make full use of client-server computing, application software must be designed specifically to run in such an environment.

Unfortunately there are many views of what constitutes client-server. A definition provided by Jeff Schulman (1993) of the US based consultancy Gartner Group says that "client-server is the splitting of an application into tasks that are performed on separate computers, one of which is a programmable workstation." A similar popular definition is "a client-server architecture is one where the application is split up into a back end, running on a powerful server that stores, retrieves and processes data, and a front end, running on a local client workstation that handles user interaction." Tatnall and Davey's (1994a) definition introduces the idea of a two tier client-server architecture. Of course we now have three tier architecture involving a "middle" step or tier. Robertson-Dunn (1993) also provides a more technical definition of client-server computing "A model of computing in which a system is partitioned into modules where commands and responses pass between modules such that one module acts as a request agent (the client) and another acts as the service provider (the server). The server keeps no requester specific information or states. This means that each command is complete and is not dependant on previous or future commands. It is possible for a server to also act as a client by requesting services of another server."

4 CLIENT-SERVER TOOLS FOR ITEM

The literature points to five important areas in which client-server architecture may be a useful concept in defining better ITEM systems:

- designing client applications;
- designing client interfaces;
- designing server applications;
- communicating from client to server;
- the concept of integrated information systems.

In Victoria, Australia a number of attempts have been made to integrate available ITEM applications at the school level. The attempts make use of the data transfer abilities of Visual Basic to incorporate information sources from across the school. As a client-server tool, VB has the advantage of being very widespread in the community, lying second in installed base to all environments. The tool is excellent for designing user interfaces and third party components make access to almost all data sources relatively easy. In addition to VB there are many tools. These include Delphi, Visual Age, SQL Windows, and Personal Oracle. All these tools are available to classical ITEM developers, both in terms of cost and learning curve.

Problems have emerged with the use of client-server tools in the local situation. These are typical of first tries in client-server technology and the limitations are those of two tier client-server systems. A three tier system incorporating a business services layer would overcome these and all the tools mentioned above allow the production of three tier systems.

4.1 Problems with client-server ITEM systems

Three tier systems provide answers to these problems. The addition of an extra tier allows us to develop systems where the user interface, business rules (such as who can view certain data, who can update and data validations) and the original data capture applications can be separated in a similar manner to object oriented systems. Products like R/3 by SAP and TP monitors like CICS and Tuxedo have allowed three tier systems, but have been too complex and expensive for traditional school based ITEM systems. The new enterprise edition of VB allows true three tier development.

5 CLIENT-SERVER EXAMPLES IN ITEM

In a typical school in Victoria, data will exist in the schools administrative package which is provided centrally and runs under the Dataflex product. This package provides student academic records, personal details, and some transaction processing ability for financial transactions of the school. The package was originally designed to provide some reporting to the school but emphasised a standardised administrative system throughout schools that enabled easy access to school data from the central government department. Much of the data on the system was useful to school administrators but was unavailable to them in any form relevant to decision making. Each school also had a wealth of additional data: library systems, student reporting, and local, teacher designed systems to do anything from timetables to sporting teams. The data redundancy of and integrity problems produced by this plethora of independent systems have not been addresses in the past due to the enormity of replacing all systems with a monolithic solution. Client-server approaches to this scenario would see a set of possibilities:

- meeting strategic planning objectives;
- meeting tactical aims;
- providing feedback on strategic measures such as attendances, performance, budgets, staff development, environmental management;
- operational control such as staffing, resource allocation;
- removing redundancy problems such as lost students;
- removing integrity problems such as exhaustive checks;
- providing information in a relevant format;
- improving communication through centralised retrieval;
- improving productivity by removing redundancy.

These possibilities have recently been addressed by a VB front end supplied to schools that provides some DSS support using the centralised database package.

Research (Tatnall and Davey 1994a) has shown that most information systems are produced to handle transaction processing. These systems can be improved to provide support for higher level management decision making. In the Victorian system an interface to the classical transaction processing system provides summary data. This can be tailored to the mission statements and their measures for each school.

Research by Davey and Tatnall (1994b) has shown that good Graphical User Interface front- end writing tools are available to the classical ITEM developer both at system and school levels. This research points to a number of conceptual problems that arise in the move to client-server but hints that these difficulties can be overcome.

6 CONCLUSION

The amount of data in an ITEM system far exceeds that which is used for educational decision making. This is often because of the diverse nature of the databases and physical distribution of the data throughout the educational system. Client-server technologies have matured, and tools are available, to such an extent that client-server solutions are available in most ITEM situations. These tools provide a mechanism for moving from disparate sources of data to an integrated information system. The lessons of the introduction of client-server systems in the broader IT world show us that client-server systems have a start up cost which is only offset by the quality of the information available from the system.

Experiences in Australian schools have shown that a client-server approach can produce an integrated information system. The wide variety of data sources and hardware in the Australian system has not proved an insurmountable problem when using modern client-server tools: However the ITEM designer must be aware of the potential weaknesses of the two tier approach.

7 REFERENCES

Davey, B. & Tatnall, A. (1994a) Problem solving and management information systems, in *Reflections on problem based learning*, Sydney, Australian PBL Network.

Davey, B. & Tatnall, A. (1994b) *Introducing object environments: cognitive difficulties.* SIGRET'94, Dunedin.

King, J. (1995) IS undeterred by the complexities of client/server. *Computerworld,* **29(17)**, 8.

Myers, M. (1995) Examining IBMs client/server blueprint for the future. *Network World,* **12(11)**, 16-19.

Roberston-Dunn, B. (1993) Cautionary tale of client-server. *Informatics,* October 1993.

Schulman, J. (1993) Quoted by G. Philipson, In search of client-server. *Informatics,* August 1993.

Tatnall, A. & Davey, B. (1994c) *Information systems: design and implementation.* Data Publishing, Melbourne.

Tatnall, A. & Davey, B. (1995) *A guide to Visual Basic.* Data Publishing, Melbourne.

8 BIOGRAPHIES

Bill Davey is a Senior Lecturer in Business Computing from RMIT, one of Australia's largest Universities. He is interested in management education, particularly the implications of client-server technologies to executive information systems. Bill lectures in systems implementation, client-server systems, end user computing and various computer languages.

Arthur Tatnall is a currently Senior Lecturer in Information Systems at Victoria University of Technology, Melbourne, Australia. Before moving to tertiary education, Arthur spent a number of years as a secondary school teacher and administrator, then a consultant and systems analyst with the Victorian State Computer Education Centre. He has written a number of school and university texts on Computer Science and Information Systems. His research interests centre around the areas of computing curriculum and management information systems in schools and business.