MONITORING COMPUTER-BASED TRAINING OVER COMPUTER NETWORKS

A THESIS SUBMITTED TO

THE UNIVERSITY OF HUDDERSFIELD

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

By

Jason Arthur Watson BEng. (Hons) AMIEE

School of Engineering,

The University of Huddersfield

December 1999

Abstract

As time is becoming an ever more precious commodity in today's workplace, effective training is also taking on an increasingly important role, but finding the time to train today's workforce is becoming increasingly difficult. With employees in diverse locations across the country and across the world and some working from home, on the road or "hot-desking" we have to take a new approach to training.
Fortunately computer-based training can solve many of the traditional problems such as the need to bring all trainees together in the same location at the same time. With today's sophisticated computer-based training applications motivated employees can train where they want, at home or at work, and when they want, at lunchtime or after work.
However, there is also a basic legal and pedagogical requirement to record who has been trained and in what. This is very easy in a traditional training scenario, but much more difficult in today's training environments. This problem is currently the major obstacle to the widespread adoption of computer-based training, and looking for a solution to these problems was the aim of this research.
This research began by investigating the processes used by multimedia developers when creating Computer Based Training (CBT) applications, identifying the current methodologies, techniques and tools that they use. Very quickly it was easy to see that developers use a whole range of development tools and that their expertise is primarily in the design of training applications, not in programming. Similarly the students want credit for the training that they undergo but do not want to be distracted by an intrusive monitoring system. The role of the Training Manager is equally important. He or she needs to be able to quickly assess the situation of an individual or a group of students and take remedial action where necessary. Balancing all of these needs in a single generic solution to the monitored training problem was the single biggest challenge.
This research has addressed these important problems and has developed a solution that permits the monitoring of student training progress in any location and at any time in a way that is totally transparent to the user. The author integrates this additional functionality into a new or existing training through a drag-and-drop interface which is very easy to use, creating a monitoring experience which is totally transparent to the trainee and the Training Manager receives a summary database of student progress.
Specifically the system uses a library of C++ functions that interface to Authorware, Director, Toolbook or a C++ application. The functions permit an author to open a monitoring database at the start of a training session and close it at the end. Whilst the database is open we can record any data that we require regarding student progress and performance. On closing the session the resulting database is sent to a central collation point using FTP. Students are identified automatically through their IP address, from their network login or ask them to logon to the training session manually. The system can write any database format that is required and if the network is unavailable when the session ends the database will be saved locally until the next training session. At the central collation point a specially written application takes the many databases created by individual training sessions and collates them into one large database that can be queried by the training manager.
Small trials were initially performed with a prototype system at the collaborating company, CBL Technology Ltd, which in turn led to larger trials at both Cable and Wireless Communication PLC and the University of Huddersfield. In these trials authors of CBT applications found the system extremely easy to integrate into their applications and the training managers and course leaders responsible for training outcomes, found the feedback on student performance, that the system provided, invaluable. This research had demonstrated that it is possible to create a generic monitored training solution that balances the needs of the trainee, the author and the Training Manager. Trainees can train at any time, anywhere in the world, over the Internet or from CDROM and a training manager can monitor their progress provided that at some time they connect to a computer network.

Acknowledgements

I would like to thank Professor. D. Taylor and Mr. P. Smith for their considerable support, guidance and expertise throughout the duration of the project. I would also like to thank my family, especially my grandmother and more recently Catja, for their support and encouragement.

This work is dedicated to Tony Jason Askew.

Contents

List of figures

List of tables

Glossary of terms

AAF	-	Advanced Authoring Format
AI	-	Artificial Intelligence
AICC	-	Aviation Industry Computer-Based Training Committee
API	-	Application Programming Interface
ASF	-	Advanced Streaming Format
AVI	-	Audio Video Interleaved
CAA	-	Computer Aided Assessment
CAI	-	Computer Aided Instruction
CAL	-	Computer Aided Learning
CBL	-	Computer Based Learning
CBT	-	Computer Based Training
COM	-	Component Object Model
DBM	-	Database Management Systems
DLL	-	Dynamic Link Library
DMS	-	Database Management System
DOS	-	Disk Operating System
DPMI	-	DOS Protected Mode Interface
DSE	-	Display Screen Equipment
DTP	-	Desk Top Publishing
E-Mail	-	Electronic mail
FTP	-	File Transfer Protocol
GUI	-	Graphics User Interface

HTML	-	Hyper Text Mark-up Language
HTTP	-	Hyper Text Transfer Protocol
ICMS	-	Integrated Customer Management System
IMS	-	Instructional Management System
ISDN	-	Integrated Services Digital Network
LAN	-	Local Area Network
LSI	-	Large Scale Integration
MAN	-	Metropolitan Area Network
MIME	-	Multipurpose Internet Mail Extensions
MIS	-	Management Information System
NOS	-	Network Operating System
ODBC	-	Open Database Connectivity
OLE	-	Object Linking and Embedding
PGP	-	Pretty Good Privacy
SME	-	Subject Matter Experts
SMTP	-	Simple Mail Transfer Protocol
SQL	-	Structured Query Language
TCP/IP	-	Transmission Control Protocol / Internet Protocol
TIS	-	Training Information System
VRML	-	Virtual Reality Modelling Language
WAN	-	Wide Area Network
WWW	-	World Wide Web

1 Introduction

1.1 The Need for Alternatives to Traditional Methods of Training

The human species is undoubtedly the most intelligent and complicated on the Earth. This point is proved by the relative ease at which the human species can affect and plan changes in its surroundings and accommodate to changes in its environment. Human beings are able to achieve this because of their very advanced brain functions that include the storing of information, the processing of new information (derived from internal or external resources) and, most importantly, the ability to analyse information and make informed decisions from it [1, chapter 1]. Similarly, computers also store, process and analyse information and therefore certain functions of the human brain can thus be somewhat compared to that of a computer.

This comparison between a human brain and a computer goes much further in that the success of computers is their ability, through programming, to be versatile and applied to many varied tasks. Human beings are also extremely versatile and very active. This is reflected in the sheer volume of activities, professions, and leisure pursuits in which people participate. An important consequence of the complexity of the human race, and its advancement, is the generation of significant amounts of new information. Much more information must be absorbed (knowledge accumulated) and analysed by an individual if he/she is to continue making informed decisions. This point is confirmed by continuing our comparison with the computer, where storage requirements, processor speed, and the application of Artificial Intelligence (AI) has increased massively to address the demands

of more advanced software. Unfortunately members of the human race are simply unable to increase their capabilities in the same way that computers do. The prospect of human beings occasionally upgrading their brains to address the increasing complexities of their lives is firmly rooted in the realms of science fiction. Instead, members of the human race employ a different computing methodology - parallel processing.

In the same way that computers use parallel processing to increase the speed at which a problem is solved, humans use the same concept, not only to increase speed, but more importantly to increase the overall capabilities of the whole. A team of people consists of individuals that are interested in different things, have acquired different skills and perform different tasks. An important consequence of this is that the individuals within a team of people are likely to require different training requirements from each other and also require retraining more frequently to meet the demand for increased capability. Rapid technological change not only means that the number of courses that students have to undergo is increasing but also their sophistication and complexity So, new cost efficient individualised methods of training have to exploited as alternatives to classroom instruction.

1.2 What is Computer Based Training (CBT)?

In today's cut-throat economic climate companies have to look more than ever for new ways of operating efficiently. When a company looks for ways of reducing its fixed expenditure it is inevitable that long-term investments such as the budget allocated for training will come under scrutiny first. So, as companies become leaner, and budgets

become tighter, the money allocated for training becomes tighter. At the same time there is pressure to maintain a well trained workforce and good on-going training programmes are essential to the recruitment of well qualified staff. As a direct consequence, trainers and managers of companies are looking increasingly towards employing CBT to fulfil the need for alternatives to more traditional methods of instruction.

Computers are by no means new to the training environment, application packages, such as word processors, spreadsheets and databases, have long been used as tools to assist in learning tasks. Computers are also used in a purely computational sense, calculating mathematical models and displaying numerical results. Often the results of such computation are used to assist students to obtain a more concrete understanding of abstract concepts.

The earliest roots of actually using a computer to instruct people goes back to the 1960s where Computer Aided Instruction (CAI) was used primarily to teach facts or for drill and practice sessions [37]. Today's CBT goes a significant step further by actually fulfilling the role of the tutor and undertaking the responsibility of invoking the learning process in the student. There are two important properties of education that allow individuals to improve their ability to solve problems and achieve goals. Definitions for each of these are presented below [1, chapter 1]:

- ***Learning*** a term used to describe those internal mental processes (and external activities) which an entity uses in order to increase its knowledge about some universe of discourse;

- ***Training*** a term used to describe those processes that enable an entity to develop and improve its performance in some task oriented skill.

Computers that provide a mechanism for teaching must address both these student educational requirements. There are many acronyms that are used to describe the application of computers to both learning and training: Computer Aided Instruction (CAI), Computer Aided Learning (CAL), Computer Based Learning(CBL) and Computer Based Training (CBT). Although there are accepted differences between the scope of each of these descriptions (beyond the extents of this thesis [1, chapter 1]), a successful teaching mechanism would involve learning, training, and instruction i.e. all of the above. Therefore, for simplicity any description of computers applied to both learning and training, replacing the role of the tutor, will be referred to as CBT in the remainder of the text.

Let us next look at the definition of CBT from the students perspective "A typical CBT course enables you to acquire the knowledge and skills that comprise competence in task performance. The course will often consist of two or more sections each of which may typically take approximately 20 minutes to complete" [2, chapter 1]. It is also highly likely that each section would be immediately followed by a test, and the end of the course would most certainly be preceded by a test to establish how much the student has learnt. The bulk of a CBT course designed to teach skills and knowledge comprises of a series of interactive learning experiences.

Figure 1 shows part of a sequence teaching about bad working postures/environments. The application interacts with the student by asking him/her to click on each of the diagrams. It can be confirmed that the screen is part of a sequence by the presence of a

task-bar at the bottom which allows the student to navigate between sequences using buttons such as 'next' or 'back'. The student can also observe his/her progress through the training by examining the progress bar.

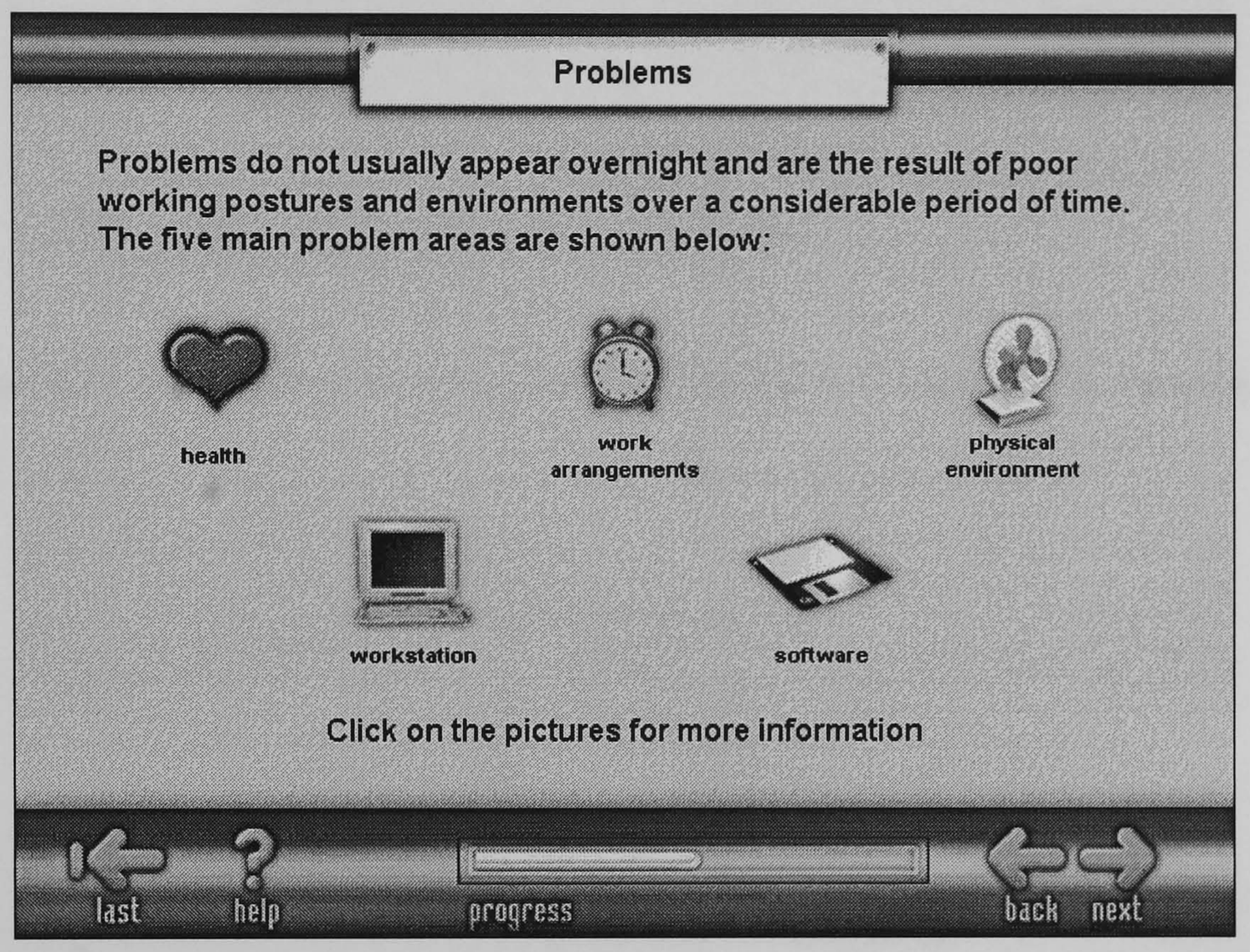


Figure 1: Part of a sequence teaching about bad working postures/environments [7]

Figure 2 shows an example of CBT administering appropriate feedback to a students response. These two figures illustrate the important principles of CBT, and how it promotes learning and training by presenting information in a logical, understandable sequence. Students should at all times know where they are, know where they have came from, and know where they are going (and why).

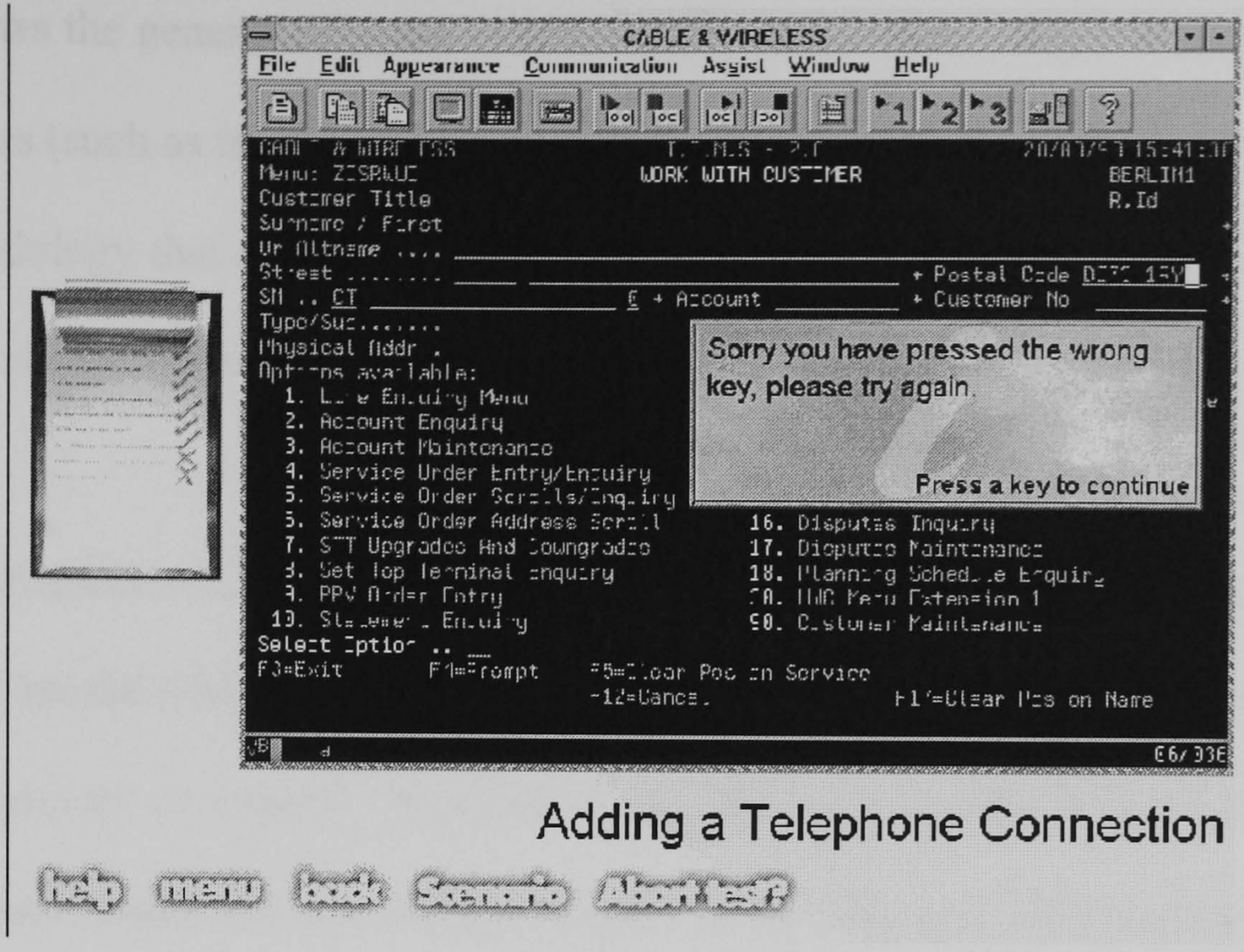


Figure 2: Example of appropriate feedback to a students response [8]

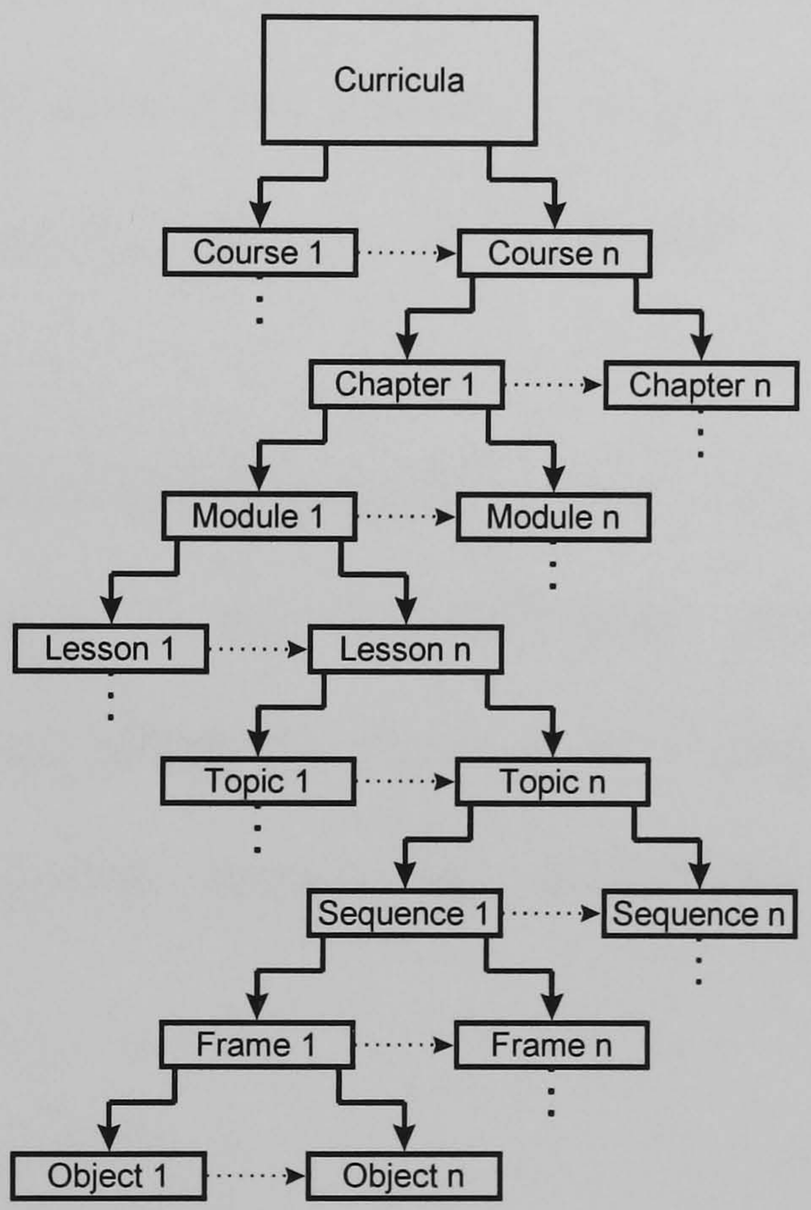


Figure 3: General structure of CBT curricula

Figure 3 shows the general structure of CBT curricula and how curricula are broken down into sequences (such as the example in Figure 1) and eventually into frames and objects. It is this methodology that allows material which was previously taught in a classroom to be successfully taught by a computer.

One main attraction of CBT is its flexibility. CBT applications can be undertaken by the student when he/she wishes, so study need no longer be dependent upon the availability of a tutor, supervisor or expert. In addition, students may choose which route they take through the curriculum and effectively navigate through the curricula hierarchy (Figure 3) at will, in their own time, and fulfilling their own particular individualised needs. An important outcome of this is that the value and efficiency of the time spent training is much improved by allowing the student to concentrate on areas of curricula that will provide the most benefit. The choice of these areas will either be left up to the student, if he/she is capable of making this choice, or to the trainer or automated.

The path a student is allowed to take through the curricula can also be determined by other factors, such as their solutions to problems they face, their previous paths, or their particular pre-defined learning objectives. This concept is termed *dynamic routing* where the CBT tailors the curriculum automatically to suit the students requirements or objectives.

CBT has now been proved, over several years and in many diverse applications to be a cost effective way of addressing diverse training requirements. This factor, coupled with the decreasing costs of PC hardware and software, both of which are advancing in power,

capabilities and ease of use, all serve to make CBT look increasingly attractive. In general, the advantages of using CBT are that costs are reduced, training effectiveness is increased, and new forms of training/simulation scenarios can be presented that would otherwise be impracticable in the conventional classroom. These points are confirmed in Table 1 which shows the general advantages of CBT compared to more conventional methodologies [3].

Category	Benefits
Reduced costs:	• Reducing training time • Reducing reliance on trained staff • Reducing the need for expensive real equipment
Increased training effectiveness:	• Consistent high-quality instruction on a large scale • High-quality training on remote sites • Hands-on, performance-oriented instruction • Individualisation of instruction
Training that would otherwise be prohibited:	• Cannot normally be observed • Happen too quickly for the student to assimilate • Occur infrequently or unpredictably • Are potentially hazardous

Table 1: The Advantages of CBT [3]

1.3 Why we need to monitor CBT

In today's challenging complex commercial world it is very important for companies to maintain exact training records for each of its employees. It is important to determine which training an employee has undertaken, when they took it, and how well they performed in any preceding assessments. Keeping up-to-date training records allow companies to introduce invaluable feedback to training managers and students and determine training effectiveness and their employee proficiency. This information is also

useful in obtaining a realistic snapshot of a companies overall capabilities and facilitates the search for experts in particular knowledge domains.

Industries such as financial services, aviation, retail, telecommunications, nuclear power and manufacturing all require documented compliance with government training regulations. Beyond these requirements, it is also a recognised problem that inadequate feedback to the learner is a substantial failing in much of the CBT solutions implemented today [32 page 34]. Administering feedback is very easy in a traditional training scenario, but much more difficult in today's CBT environments where students seldom receive individualised advice.

To address this issue, reliable and robust systems need to be implemented that allow the tracking of employee's progress through curricula. Such systems need to highlight non-compliance and allow companies to avoid exposure to litigation. The general reasons that we think support and justify the requirement to monitor and track performance in training are summarised in Table 2.

Rapid technological change has increased the number, and complexity, of courses employees have to undertake. As a consequence there is a tangible requirement to automate the process for gathering and analysing the monitoring and performance records produced by training. The advent of CBT, driven by its success at reducing costs and increasing training effectiveness, offers an excellent opportunity to completely automate performance monitoring, and analysis of training. This opportunity is further enhanced by the application of network technology. CBT and network technologies (such as the

Internet) improve the distribution of courses, provides centralised record-keeping and facilitates distance learning.

Reason	Description
To assess training effectiveness	As budgets allocated to training are scrutinised, training managers are required to provide documentary evidence that the training deployed is effective, beneficial to the student, and good value for money.
To highlight non-compliance	Many companies are forced by law to be legally responsible for the training of their staff. As a consequence it is imperative for them to be able to provide supporting evidence when challenged.
To provide rapid feedback	Training managers need to establish whether a student is struggling so that remedial/intervention strategies can be initiated. Students have a more satisfying experience during their training session when constantly aware of their progress.
To assess group/individual proficiency	It is imperative for a company to know the proficiency of its employees and departments in order to strategize and initiate changes to increase efficiency.
To focus in on domain experts	A company informed about how its skills are distributed within its workforce can tailor the way tasks are assigned to increase efficiency.
To record continuing professional development	Individuals wish to gain credit for the training they have undergone as part of their professional development.

Table 2: Reasons supporting monitoring of training

Training costs can be split into two areas: (1) the cost of the training asset itself, and (2) the lost opportunity of using that student when they are undergoing the training. Networked CBT would reduce staff downtime by allowing geographically separated students to train at any time and place they want. In addition, the implementation of such a centralised system would introduce invaluable rapid feedback to training managers making it

extremely easy to evaluate training effectiveness and employee proficiency and enable them to hunt down subject matter experts when required. In addition, the costs of physically delivering and installing CBT is dramatically less if done via a computer network when compared to CD-ROM delivery.

1.4 The impact and Issues of Networking

The proliferation of computer networks has made networked CBT a viable solution. Organisations wish to meet the training requirements of their staff in the most efficient manner possible and, therefore, are starting to reap the benefits of network connectivity. Network connectivity combined with CBT offers many rewards. Firstly, it gives the opportunity for the training itself (the content) to be transmitted to a large number of students (recipients) in real time. This means that organisations can update content frequently without incurring the expense of publishing to CD-ROM and the expense of distribution. In addition, because content is transmitted directly to each recipient as opposed to being broadcast, and also because information can be transmitted back, there is an opportunity to dynamically transmit specific training in order to meet the needs of particular students.

Secondly the connectivity provided by computer networks allows the opportunity to transmit monitoring information from the student back to the training manager for analysis. This is in fact the main theme of this thesis and is discussed in depth in the preceding sections.

Thirdly, data can be transmitted between individual students in order to develop a collaborative environment. Students who are training by distance learning can use this to enhance their training experience. Collaborative environments, their benefits and how they can be implemented, are discussed as further work in section 9.5.

The issues to do with networking CBT are very similar to those of networking any type of application. Networks have bandwidth limitations that restrict the speed at which data can be transported. Also, there are several different types of networks that are physically different from each other and many different software protocols. However, high level protocols which have been established by the popularity of Internet is removing much of this complexity. In the context of transmitting CBT content over computer networks the bandwidth limitation is of particular relevancy. CBT content normally includes interactive multimedia and, therefore, large file sizes and a higher bandwidth requirement.

Computer networks and their implications on CBT delivery and CBT monitoring are examined in depth in sections 2.2.1 and 2.3.

1.5 Project Goals

As the popularity of CBT increases, many customers are beginning to realise the advantages of automated monitoring systems. Customers are now beginning to request that CBT developers integrate such systems into the applications they produce. This has led to a proliferation of bespoke monitoring solutions that are designed for specific CBT applications. The sheer quantity of CBT produced has created a definite need for some

interoperability between the monitoring systems they employ. The ideal solution to this would be to implement a single, off-the-shelf, monitoring system that would work both with all CBT applications, irrespective of the authoring tool used to create them, and over a LAN or a WWW network. Developers, who commission CBT systems, would find it much easier if they could use a monitoring solution that they can implement in any CBT product they develop. Likewise, training managers would like one analysing tool (Figure 4) that can extract information from all the different types of CBT products they use.

The need for a consistent way of recording and analysing monitoring information is emphasised by the introduction of new network technologies. Network developments such as the accepted popularity of the Internet and the introduction of intranets mean that computers are no longer isolated stand-alone machines. The significance of this is that distance learning with CBT is now a possibility. Companies that provide training have access to a vast previously untapped market of people with many varied training requirements. It is therefore vital that standard ways of outputting, manipulating, delivering and analysing monitoring information are introduced. There is thus a need for a 'design once' but 'use many times' monitoring system. The objective of this project is to examine the possibilities of just such a system, and its specification is to address the issues presented in Table 3.

Issue	Description
Reusability	The monitoring system must be capable of being incorporated into any CBT application written in any programming language.
Variety of CBT	CBT is so diverse and customer requirements so different that there is difficulty in pinning down one specific methodology that is generic enough to be used to monitor all applications. The system must provide the author with the flexibility to create their own monitoring system; i.e. The definition of 'monitoring within CBT' must be set by the author rather than the system.
Integration into CBT	The system must be as easy as possible to integrate into CBT by the CBT developers. CBT developers are very often Subject Matter Experts (SMEs) with knowledge about content and teaching techniques but with limited programming skills. The system must be able to seamlessly integrate into the CBT with the minimum effort from these people.
Distance learning	The system must be able to service distance learning scenarios, such as training over the Internet. The system must incorporate some mechanism that transmits reported data over the Internet.
Future enhancement	Development in CBT and network technology occurs incredibly fast. An encompassing, generic system must address this problem and be designed to facilitate rather than hinder any foreseeable developments.
Open standards	Industry likes products that use open standards and consider them less likely to put other systems at risk. The design should try to steer clear of using unusual standards, obscure file formats, or communication methods that would limit its compatibility with other systems.

Table 3: Issues of a generic monitoring system

The aims of this research are focussed on developing a flexible and generic monitoring methodology. This will involve consideration of the entire CBT development, implementation and courseware management cycle. This is because a successful monitoring system will need to take into account users needs from all perspectives. Users have different requirements due their quite different roles as authors, students, network managers or training managers

2 Design Considerations

2.1 Monitoring Requirements of CBT

In section 1.2 the concept of dynamic routing was introduced. Dynamic routing is the term used to describe how CBT generates the path the student is allowed to take at run-time. Often the dynamic path is determined by the student's requirements previously judged by their earlier performance in the CBT. Consequently, CBT using dynamic routing already performs its own tracking and analysis. For the CBT to address the monitoring requirements listed in Table 2 an additional method is needed to report performance records externally so that they can be analysed by a separate tool under the control of the training manager.

This is illustrated in Figure 4 where the student, in a learning role, interacts with the CBT which records performance/tracking information externally onto temporary storage. The training manager, in an assessing role, is then able to interrogate the information using some form of analysis tool.

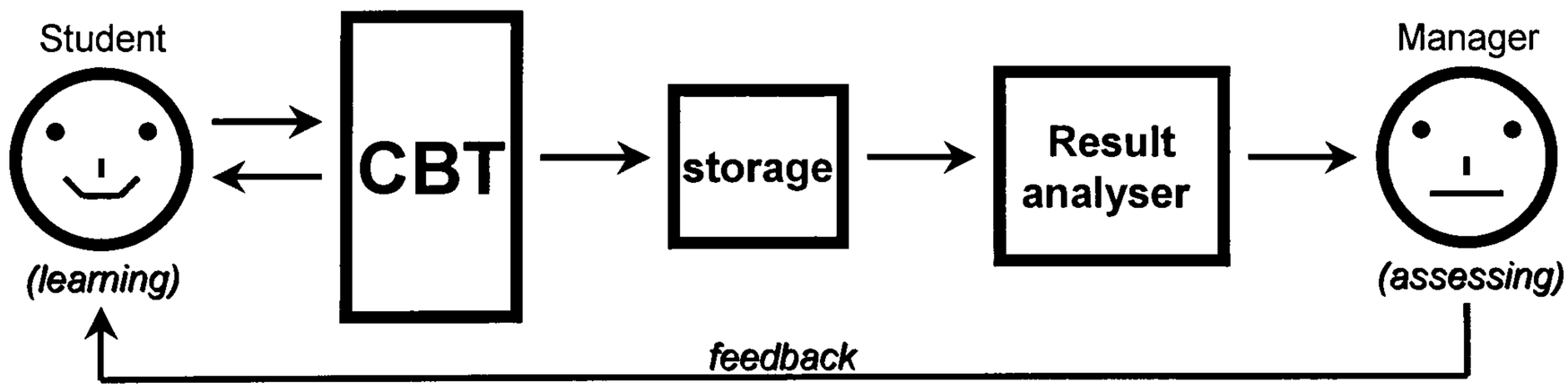


Figure 4: CBT reporting student performance externally

This system would work equally well for appraising many students. The kind of information that the CBT would generally have to record onto the storage media is summarised below:

- CBT name
- CBT unique ID
- student name
- student unique ID
- student location
- training workstation location
- parts of training visited
- parts of assessments attempted
- results of assessments
- time spent training

Using this information the analysis tool would be able to illustrate to the training manager which pages/chapters/sections of the CBT a student has visited, which parts of an assessment the student has attempted and how well they performed [39]. The training manager could use the analysis tool to compare students and get an overall picture of the performance of the students and indeed the performance of the CBT application itself. This functionality is extremely important to the Training Manager, allowing him/her to ascertain if the expenditure on training is justified.

The introduction of computers into training allows an accuracy and depth of monitoring that was previously unreachable in the conventional classroom scenario. As a consequence some people may consider the accountability that such systems afford as an infringement of their human rights. In modern commerce where impetus for all actions are inevitably driven by financial considerations, businesses cannot afford to employ staff who are not willing to be accountable. Monitoring of a person's actions can only be considered as such

an infringement if it is done whilst the person is unaware it is happening. This point is confirmed by British law disallowing taped conversations to be submitted as evidence in court if none of the speakers were aware that they were being recorded at the time.

2.2 Developing CBT using Authoring Systems

This section examines the different authoring tools and authoring methodologies that are used to create today's CBT applications. This is important as the CBT monitoring system will have to successfully integrate into existing authoring tools and authoring methodologies. There are currently numerous ways of developing a CBT application. CBT applications are effectively interactive multimedia applications that invoke learning in the user.

There are two analogies that illustrate the development concept of multimedia applications extremely well [26, chapter 7]. Firstly, publishing where text, graphics and charts are prepared by various means and then finally integrated using a Desk Top Publishing (DTP) application. Secondly, television newscast production where people go into the field recording video, developing graphics, writing scripts and edit audio prior to integrating the components together for broadcast. However, as previously mentioned, CBT, and in fact most multimedia documents, are interactive and differ significantly from these examples. CBT and Multimedia applications do indeed demonstrate the integration of different types of elements in time and space but are also characterised by an inherent temporal dimension. This is a departure from the standard "What you see is what you get" (Wysiwg) paradigm as you cannot immediately see its results when you specify dynamic

routing and dynamic behaviour (via user responses) [27 chapter 8]. Thus, the integration of media components into a CBT application requires some kind of a programming tool, so that so that interactivity can be built in, and the flow of execution controlled.

There is an extremely wide range of programming languages that are in use today for authoring, ranging from low level languages such as C, C++ or Java, up to much higher level languages such as Visual Basic. In authoring, high level languages are more user friendly, and much easier to learn and use, than low level languages but by contrast they offer less flexibility, decreased speed and larger file-sizes. Even though computers have increased in processor speed and storage capacity, there is still a case for using low level code, especially for games and simulations, when developing some types of multimedia. However, more increasingly, multimedia is being developed using higher level languages, and quite often using high level tools that provide an intuitive Graphics User Interface (GUI).

These high level GUI tools are now the mainstay of CBT development. This is because people who develop CBT should know how to train people and invoke learning through interaction but are not necessarily, in fact unlikely, to be computer programmers. Authors of training material are most likely to be subject matter experts and may possess very little programming expertise and only a small amount of computer literacy. Therefore, CBT applications are seldom programmed, but rather *authored*, instead, by an authoring tool. Authoring, in respect of CBT and multimedia development, can be thought of as "programming by non-programmer". Authoring tools, or authoring environments, offer

what is effectively the highest level of abstraction from computer implementation (as demonstrated in Figure 5).

Authoring tools are much easier to learn how to use than lower level tools or languages, and hence allow non-programmers to build multimedia or CBT applications extremely quickly. However, although authoring tools are good for rapid development, and also prototyping, they do not match the wide ranging flexibility and functionality that the lower level programming languages can provide. In addition, these high level authoring tools tend to produce software that has larger file-sizes and operate more slowly than their low level programmed counterparts.

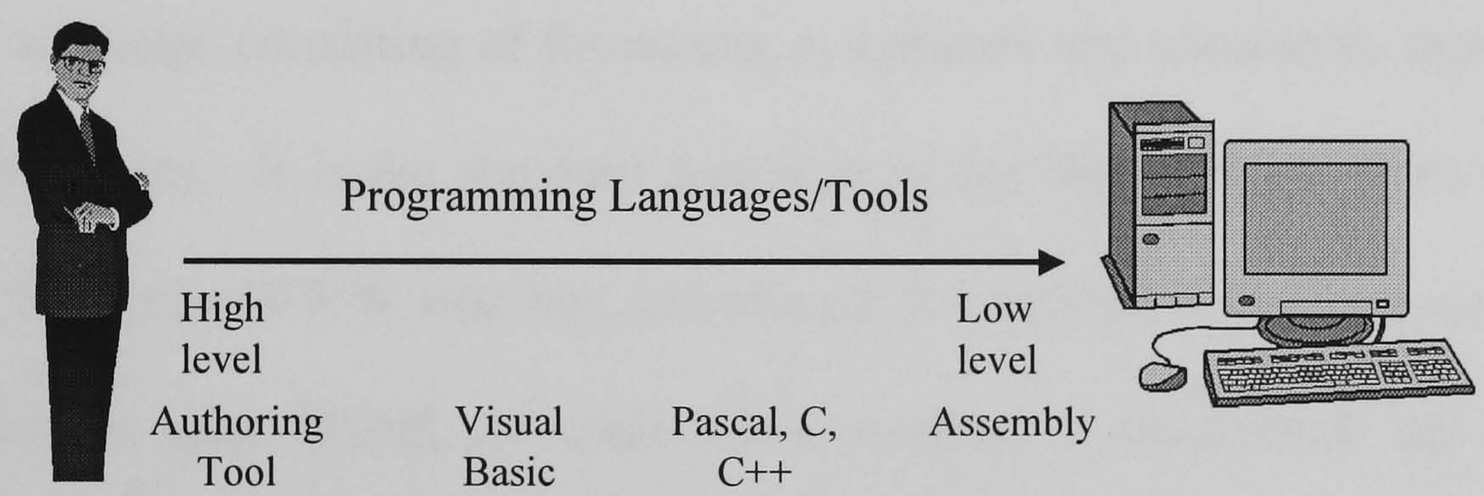


Figure 5: Level of abstraction from computer

It is extremely important, when considering authoring tools, to know the difference between an authoring tool and a presentational tool. A presentational tool such as Microsoft PowerPoint, or Harvard Graphics is very different. Although these presentation tools are similar in that they allow media to be integrated into an end product, they do not offer branching possibilities, support device control and allow the use of variables and functions like an authoring tool does. An authoring tool could therefore be considered a hybrid of a presentational tool and a programming language.

In order to help developers to conceptualise the authoring process, authoring tools all follow particular metaphors [28 chapter 4.2.5]. The metaphor conveys how the program organises elements, sequences, events, and delivers the multimedia application. There are effectively four main types of authoring systems: hypertext or hypermedia based, page and object based, flow line and icon based, and time and script based. All these different authoring system types provide different analogies and offer different programming processes that must be followed in order to build a multimedia or CBT production.

Hypertext or hypermedia based

The most prolific example of this is Hyper Text Mark-up Language (HTML). HTML is a presentation language consisting of formatting commands and statements that can be used to create Web pages. It is the standard language of the World Wide Web (WWW) [29 page 146]. When the WWW was first introduced, it was built entirely around the use of HTML, however, now HTML is most often used as a framework on which other components can be hung [33, page 1037].

These components are 'active' or 'interactive' beyond what HTML alone can do and are typically: ActiveX controls, Java applets, Java script programs and others. Until quite recently these technologies were difficult to employ, especially by a non-programmer (author), and hypertext or hypermedia based authoring would be a poor solution for CBT development. However, there are now hypertext or hypermedia based authoring systems, such as Macromedia's DreamWeaver, which allow people to quickly design web pages, even with some active content.

These tools provide a good level of abstraction from the actual HTML code and much of the underlying technologies of the active content. However, these tools are still quite low in programmable functionality by comparison with the other, more conventional – not particularly web based, authoring tools currently available. As a consequence, hypertext or hypermedia authoring systems are not used by the majority of CBT authors, and are not likely to be used until functionality is increased and authoring complexity is reduced even further.

Page and Object Based

These types of authoring systems have a very object orientated approach and are based on the author editing properties and behaviours of on screen components. The components are items such as graphics, buttons, animations and just about any object the application user is likely to come across. Control is achieved by objects passing messages to each other. Typically a user would cause an *event* by taking some action in the window, such as pressing a button or clicking the mouse, and a message is generated.

Objects normally have built in responses to these messages which the author can switch on/off and configure. Many authoring tools that follow this metaphor also allow custom (user-defined) responses to be programmed into objects through the use of some sort of scripting language.

A good example of a popular page and object based authoring system is ToollBook, designed by Asymetrix. ToolBook provides an intuitive GUI that allows an author to drag multimedia content from a catalogue, similar to a library, onto pages to be viewed by the

user. All objects can contain properties and can be set to respond to events in a standard way or in a user-defined way.

Flow line and icon based

The best example of a flow line and icon based authoring tool is the extremely popular authoring tool , Authorware, developed by Macromedia. Two windows are used during development, the design window and the presentation window. The design window is where the author arranges icons and effectively builds a flow-chart representing the logic of the piece. The presentation window shows how the piece will look when it is run. The author uses the presentation window, when building the piece, to lay out text and graphics, buttons, and all the other visual elements that make up the piece. Creating an application is quite a simple matter with icons being dragged from a palette onto the 'flowline' in the design window.

This flowchart metaphor provides a clear and easily understood overview of the application's structure that is very applicable to the majority of CBT applications, which tend to be based on a flow of control model.

Time and script based

This approach is often linked with a movie metaphor. A good example of a time and script based authoring system is Director, also designed by Macromedia. In Director, applications that are created are known as movies. The author effectively uses three working windows: the cast, the stage and the score. The cast window contains thumbnails of all the media elements, or cast members, that can be incorporated into the piece. Cast

members can be dragged and positioned in the stage window and have their animation controlled by the score. The score is a series of channels divided into frames. When applications are previewed a 'head', much like a video player head, moves over the score activating events specified to trigger at individual frames. These events control the behaviour of the cast members on the stage. Cast members can have standard responses to events or user-defined responses specified by Director's powerful scripting language: lingo.

There is no doubt that a key factor in spurring CBT and multimedia market growth is the availability of new software tools [30 chapter 18]. These object orientated, solution based, authoring tools that we have discussed have made the creation of multimedia products, such as CBT, accessible to the non-programmer.

2.2.1 Delivery of Authored Pieces

The program files that are created through the use of an authoring tool are usually quite large by comparison to files created using an alternative type of software development. There are two reasons for this: firstly, the content almost certainly has media such as images, sounds and animations that require greater storage capacity, and secondly, program files produced using authoring tools are generally 'interpreted' and require a run-time engine component to be included in the delivery if it is to be run on a operating system (Figure 6). Traditionally, CD-ROMs have been the chosen way to deliver CBT applications due to their large storage capacity. However, more recently, increased

connectivity and faster network speeds has made delivery of multimedia CBT applications over computer networks a real possibility. media development creates.

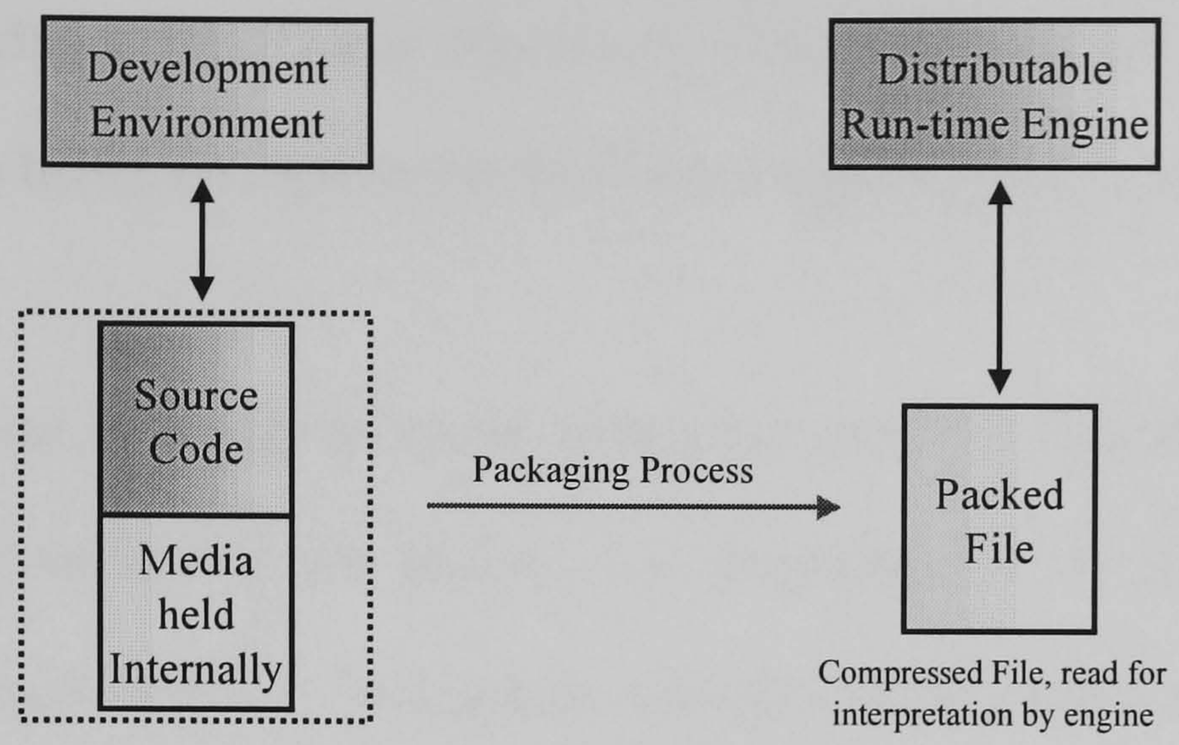


Figure 6: Conventional Authoring System Delivery

The benefits of having a CBT application delivered over a network is very easy to realise as it presents an immediate method of distribution to learners, and provides potential for feedback of CBT usage and student performance tracking. Delivery of networked CBT can be achieved over an ordinary Local Area Network (LAN) or, more recently with the development of browser plug-in technology, the World Wide Web (WWW). When one compares a CD-ROM distribution method to a computer networked distribution method, there is no comparison to the efficiency and flexibility afforded by networked distribution. In fact, the author's opinion, CD-ROM's look prohibitively expensive by comparison.

If this is the case, then why is most CBT still delivered by CD-ROM? The key reason for this is the large amount of bandwidth CBT and Multimedia applications require and the fact that many networks, although the situation is improving, are simply not fast enough to deliver the type of multimedia that customers desire to include in their training. This

factor especially applies to training over the Internet via the WWW. Many Internet users still connect to the network using a dial-up modem that has a maximum connection speed of 56Kbps. This is simply not fast enough for the deployment of professional, interactive, and attractive looking CBT. This is why much of the CBT available over the Internet is either perceived as being too slow or too much like a page turning exercise.

Today, a compromise is struck with most institutions accepting that although professional CBT cannot be delivered over the Internet, it is practicable over a fast company LAN or Intranet (mini internal WWW). In addition, authoring tool vendors are developing more and more effective ways to seamlessly incorporate pieces into the WWW, and are using streaming technology to minimise the data that is downloaded over the network (described in Figure 7). This streaming technology allows CBT applications to sit and operate within a web browser such as Netscape Navigator or Internet Explorer. Large savings on bandwidth can be made, as only the parts the user views of the self navigable training are actually downloaded. Previously, the whole authored piece would have to be downloaded prior to running.

Further advancements are being made with tool vendors developing intelligent look-ahead cache systems, that predict the users next move in order to download the relevant data. Using these technologies CBT is being deployed increasingly in distance learning environments, over computer networks, making monitoring of activities a pressing need.

1. Use special application to split file up into small pieces + index
2. Refer to index in HTML file
3. View application in over WWW in a browser with correct plug-in installed

Figure 7: Deployment of a Authored piece over the WWW

2.3 Developments in Network Technologies

Communication networks are an essential feature of our everyday lives. In our modern society we are exposed to networks continuously. In the home we use the telephone, listen to the radio, browse the World-Wide-Web (WWW), and get distracted by our pagers. At work we send/receive faxes, listen to our voice-mail, read our e-mail, and copy files from one computer to another.

Historically networks were developed to address the need to communicate over long distances. During the 1950's two developments took place that would radically change the way the world communicated. The first was the introduction of commercially available computers (the first mainframes) and the second was the launch of the Soviet satellite Sputnik in 1957. The launch of Sputnik forced the competing Americans to launch the first communications satellite in 1960, Echo I, and in 1964 this was joined by Echo II. Echo I and Echo II were used to reflect radio signals across the world, solving the problems encountered with atmosphere reflection techniques. In parallel with the advent of satellite communications, by the end of the 1960's the computer was firmly established

as an invaluable data processing tool. This led to the development of Wide Area Networks (WAN's) to facilitate the sharing of common data between geographically distributed computers.

Advances in transistor and Large Scale Integration (LSI) technologies led to reduced manufacturing costs, increased reliability, the introduction of minicomputers, and then finally microcomputers. As costs spiralled downwards computers became more accessible and their use began to proliferate. People began to use individual workstations instead of sharing a large central processor, and consequently, a solution was required to address the problem of sharing common peripherals and to fulfil the requirement to exchange data between computers locally. This led to the development of Local Area Networks (LAN's). Table 4 [5, chapter 1.3] specifies the general network definitions and the differences between them.

Network Type	Definition
Wide area networks (WANs)	Connect networks over a large physical area, such as between different buildings, towns or even countries.
Local area networks (LANs)	Connect computers within a single office or building. They typically connect to a common electronic connection – commonly know as a network backbone. LANs can connect to other networks either directly or through a WAN.
Metropolitan area networks (MANs)	Connect networks over a large physical area larger than a LAN, such as around a town or city. Can be considered as a subdivision of WANs.

Table 4: Network definitions

For quite some time now, there have been great advantages in companies establishing a computer network. These range from simply gaining the ability to share resources to the complete enhancement of corporate structure (Table 5 [38, chapter 1.1]) . A computer

network consists of both hardware and software. The hardware defines the network topology used (the shape / physical structure) and includes the network interface cards and the cable that ties them together.

Three network topologies – bus, token ring, and star – account for most networks in use today [16, page 223]. In 'bus topology' all devices are connected to a central cable, called the bus or backbone. Bus networks are relatively inexpensive and easy to install for small networks. Ethernet systems use a bus topology. In 'ring topology' all devices are connected to one another in the shape of a closed loop, so that each device is connected directly to two other devices, one on either side of it. Ring topologies are relatively expensive and difficult to install, but they offer high bandwidth and can span large distances. In 'star topology' all devices are connected to a central hub. Star networks are relatively easy to install and manage, but bottlenecks can occur because all data must pass through the hub [14].

The software components include server operating systems, communication protocols, and network interface card drivers. To enable computers to talk over the networks (and networks to be joined together) there needs to be strict adherence to network protocols. These protocols control how data is routed to and from devices on the network. Any device that can 'speak' in a protocol that the network supports can send and receive information to any other device that can also 'speak' that protocol. One of the most important attributes of a computer network that must be considered, is bandwidth. Bandwidth is the amount of data that can be transmitted in a fixed amount of time. For digital data transmission over computer networks, the bandwidth is usually expressed in

bits per second (bps) or bytes per second. Kbps is 1,000 bits per second (not 1024 per second).

The bandwidth available to a device connected to a network is extremely important in determining its application. If the device is only using the network to transmit/receive text messages on the network, then a low bandwidth connection would suffice. However, if the device had a requirement to transmit 'bandwidth intensive' material such as Video and CD quality Audio, then a high bandwidth connection would be mandatory. As current technologies continue to improve, software sophistication increases, and hardware gets cheaper to build, the potential for multimedia machines to become as common as your VCR is greater than ever [15, page 3]. As a consequence it is becoming increasingly necessary for networks to transfer the huge amounts of data involved in graphics, audio, and videos.

Reasons for establishing a computer network
• Program and file sharing • Network resource sharing • Database sharing • Workgroups • Electronic mail • GroupWare and workflow software • Centralised management • Enhancement of the corporate structure

Table 5: Reasons for establishing a computer network

Today, most established networks have been updated, or are being updated, to faster standards and will be able to send and receive large amounts of data extremely quickly. Also, telecommunication companies have much improved their infrastructure and are now

able to provide high bandwidth connections. As a consequence, it is becoming increasingly common for the bandwidth stumbling block to occur in the off-ramp from the network to the device. For example, in the case of the home user, the standard copper wiring that connects to most phones has a very low data bandwidth. Figure 8 shows the different types of adapters that networked devices use to connect to a network. A different type of network adapter is used for the different types of off-ramps.

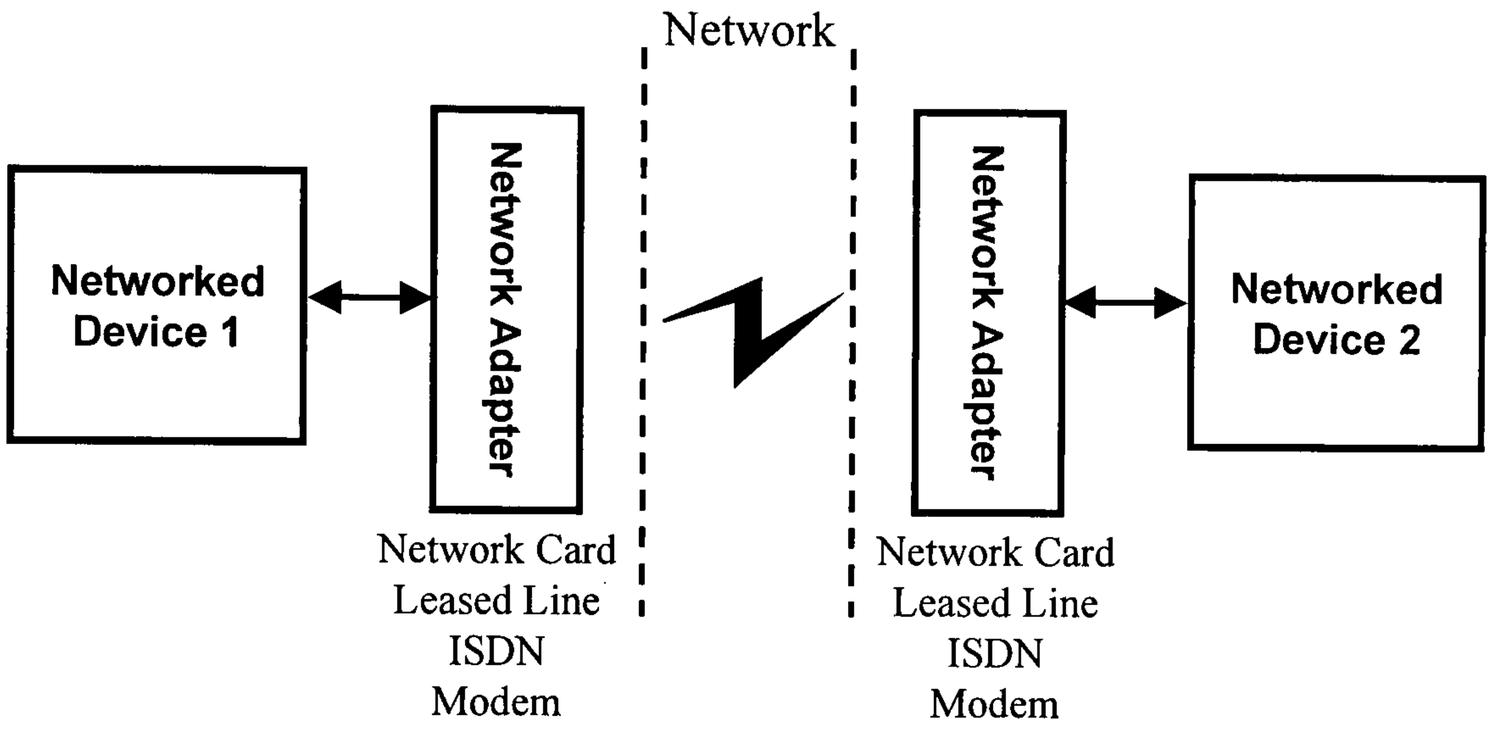


Figure 8: Types of off-ramps to networked devices

The type of connection the device has to the network has a significant effect on bandwidth, and therefore limiting the applications that the device can be put to (Figure 9). There are four main methods of connecting a network (or an independently connected device/computer) to another network [5, chapter 1.3]. The different methods are displayed in Table 6.

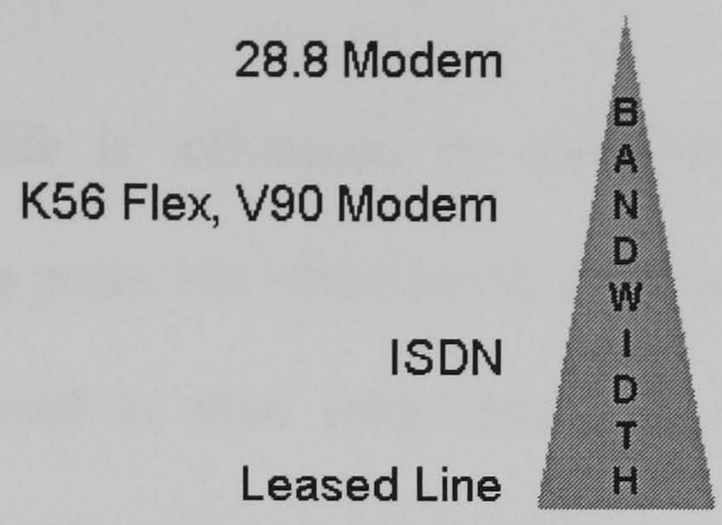


Figure 9: Network Connection Vs Application

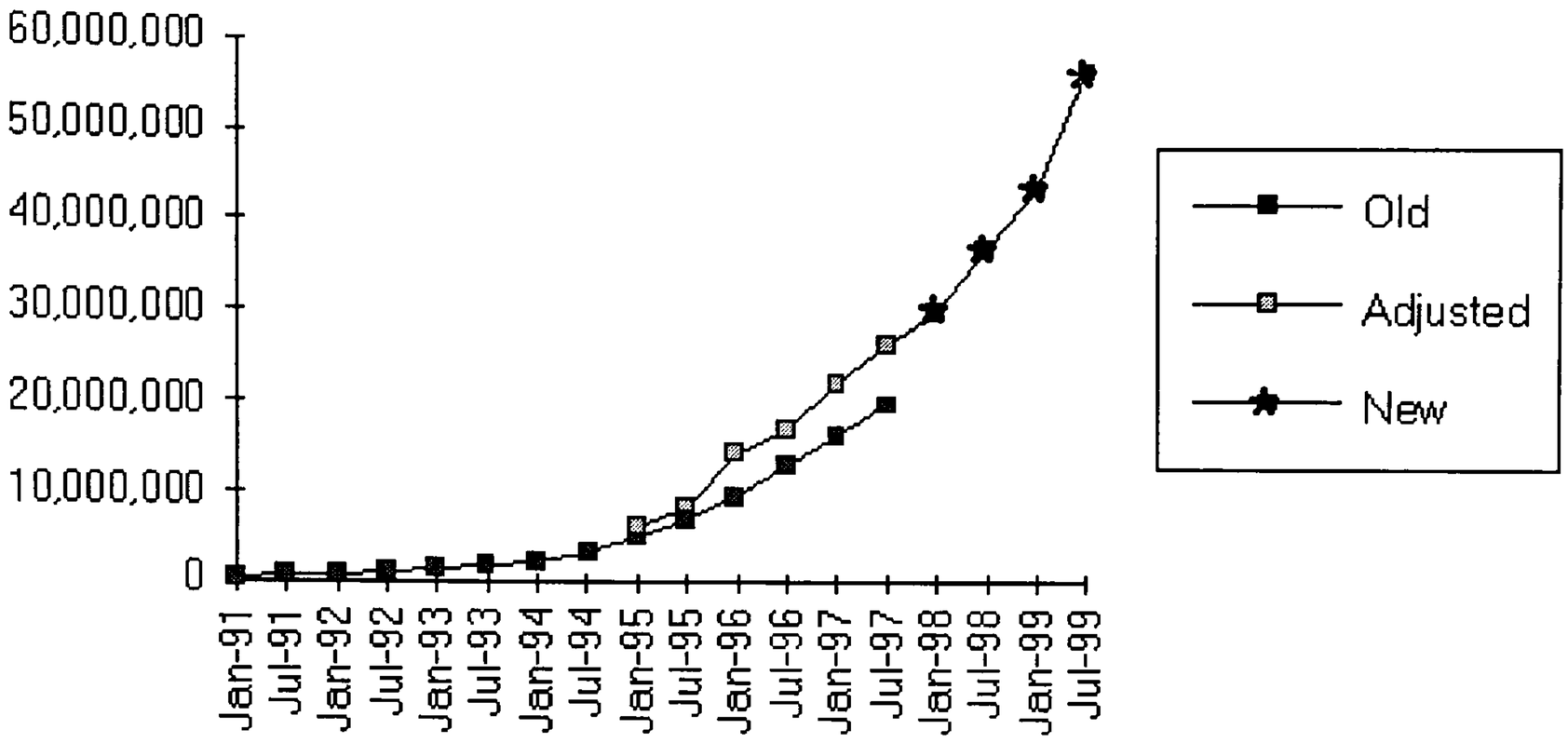
Connecting Method	Description
Network Card	A network card is a computer circuit board or card that is installed in a computer so that it can be connected to a network. PCs connected to LANs typically contain a network card specifically designed for the LAN transmission technology, such as Ethernet (Bus topology) or Token Ring (Ring topology). Ethernet can transfer data at a rate of 10000 Kbps
Leased Line	A leased line is a permanent telephone connection between the computer and the network set up by a telecommunications carrier. Typically, leased lines are used by businesses to connect geographically distant offices. Unlike normal dial-up connections, a leased line is always active. The fee for the connection is a fixed monthly rate. Leased lines are categorised as a T1, T2…etc.. depending on their Bandwidth capability. A T3 leased line has a bandwidth of 44.736 Mbps
ISDN	Integrated Services Digital Network is an international standard for sending voice, video, and data over digital dial-up telephone lines. ISDN requires special metal wires and supports data transfer rates of 64Kbps. Most ISDN lines offered by the telephone companies give at least two lines which can be combined to give data rates of 128Kbps (three times faster than the fastest modem)
Modem	Acronym for modulator-demodulator. A modem is a device that enables a computer to transmit data over ordinary copper telephone lines. The modem converts digital information into an analogue signal so that it can be transmitted down the telephone line in exactly the same manner as a voice. The fastest theoretical data transfer speed that a modem can achieve is 57,600bps

Table 6: Network connection methods

2.3.1 The Internet

Today everyone's life is influenced by computers. Computers have replaced filing cabinets, the printing press, the white board, and may perhaps eventually even replace the teacher. They are used to store information, design products, transfer money, control manufacturing processes and manage warehouses. Most importantly of all, with the introduction of network technology, computers are now used to *communicate*.

The largest communication tool, in the world today is the Internet computer network. The Internet, or INTERnational NETwork, is a global interconnection of networks and independently connected computers that form the largest computer network in the world. At the last count the Internet was found to link some 30 million computers in more than 100 countries world-wide [6]. The number of computers connecting to the Internet, and the uses the Internet is put to, is growing day by day. This is illustrated in Figure 10, where the numbers of Internet hosts can be seen to be increasing exponentially. The two graphs (old and new) demonstrate a switch in the method used to obtain the figures caused by increasing security adopted by the organisations using the Internet. The new method is the most accurate.



Source: Internet Software Consortium (http://www.isc.org/)

Figure 10: Internet Domain Survey Host Count[6]

Most of the networks and computers connected on the Internet are owned by different companies, universities or government departments. As a result, the Internet is not owned by any one organisation or country. Nobody is in charge of the Internet, however, there are some organisations that by common consent rule over things such as the domain names that appear in addresses such as www.zdnet.com, www.whitehouse.gov, www.ucla.edu [16, page 231]. However their rule is entirely at the whim of the governed. The numbers of uses the Internet is being put to is growing fast, but the most common uses of the Internet today are listed in Table 7. This list is by no means comprehensive, in fact the Internet is so new, and its growth in users so prolific, that there is no limit to the way in which it can be used. The only limit is a person's imagination. The key attribute that makes the Internet so vital and important is its universal receptiveness. The Internet is open to anyone with access to a computer and a phone line.

Facility	Description
World-Wide-Web (WWW)	A hypermedia system that allows browsing through lots of information (normally using a web browser). Now the preferred method of presenting and accessing information on the Internet.
Electronic mail (E-Mail)	Text messages sent from one individual connected to the Internet to another. Other standards such as Multipurpose Internet Mail Extensions (MIME) allow graphics, video, sound, or binary files to be sent as email attachments. Email is the most commonly used Internet facility.
File transfer (FTP)	A method of transferring files from one computer to another over the Internet that allows both machines to confirm successful receipt.
Remote control of remote equipment (Telnet)	A program which allows a user to remotely log onto another computer across the Internet and execute programs on it.
Video conferencing	Video conferencing system works much like a video telephone. Each participant has a video camera, microphone, and speakers mounted on his or her computer.
Newsgroups	A distributed bulletin board system about a particular topic. Systems such as Usenet news distribute thousands of newsgroups to all parts of the Internet.
Internet Relay Chat (IRC)	Chat system developed by Jarkko Oikarinen in Finland in the late 1980s. IRC enables people connected anywhere on the Internet to join in live discussions. Unlike older chat systems, IRC is not limited to just two participants.
Multimedia Streaming	Streaming is a technique for transferring data such that it can be used before the entire file has been transmitted. Streaming technologies are becoming increasingly important as most users do not have fast enough Internet access to download large multimedia files quickly.

Table 7: Uses of the Internet

Facility	Description
Common Gateway Interface (CGI)	A specification for transferring information between a World Wide Web server and a CGI program. CGI programs are the most common way for Web servers to interact dynamically with users. Most search engines employ CGI
Java	A high-level programming language developed by Sun Microsystems that includes a number of features that make the language well suited for use on the World Wide Web. Small Java applications are called Java applets and can be downloaded from a Web server and run on a client computer.
Video on Demand (VoD)	A Video-on-demand system delivers video directly to the computer desktop and provides the ultimate flexibility of video services by allowing users to select the video programs, the times they want to watch them, and to perform any VCR-like functions. This service requires an extremely high bandwidth.

Table 7: Uses of the Internet (continued)

The Internet came into being to address the strategic problem of continuing to communicate after the event of nuclear war. It was very obvious to the Cold War think-tank that any network, irrespective of its armour or protection, would always be vulnerable to the impact of atomic bombs. Another problem the scientists had to deal with was that any central network control centre would be an obvious and immediate target for an enemy missile. The solution was extremely simple, the network itself would always be assumed to be unreliable, but designed to overcome its unreliability, and all nodes in the network would have equal status to all other nodes with their own authority to originate, pass on, and receive messages.

The messages sent over the network would be divided into separately addressed packets of information that would be directed through the network on an individual basis. The particular route a packet took would be unimportant, and it would be tossed from node to node like a hot potato, more or less in the direction of its destination, until it ended up in the proper place. This philosophy may initially appear inefficient, but it was incredibly rugged. If large portions of the network were blown away, undelivered packets would travel an alternative route to their destination, and any lost packets would be re-sent.

The Internet was born in 1969 when 4 nodes on the infant network were connected together and named ARPANET. The decentralised structure of ARPANET made expansion extremely easy. Unlike standard computer networks, any machine that spoke the packet switching language 'Transmission Control Protocol / Internet Protocol' TCP/IP could communicate over the network. The brand name or type of computer was irrelevant. Figure 11 shows the general structure of the Internet; a vast interconnection of networks and computers all talking to each other in the same language (TCP/IP).

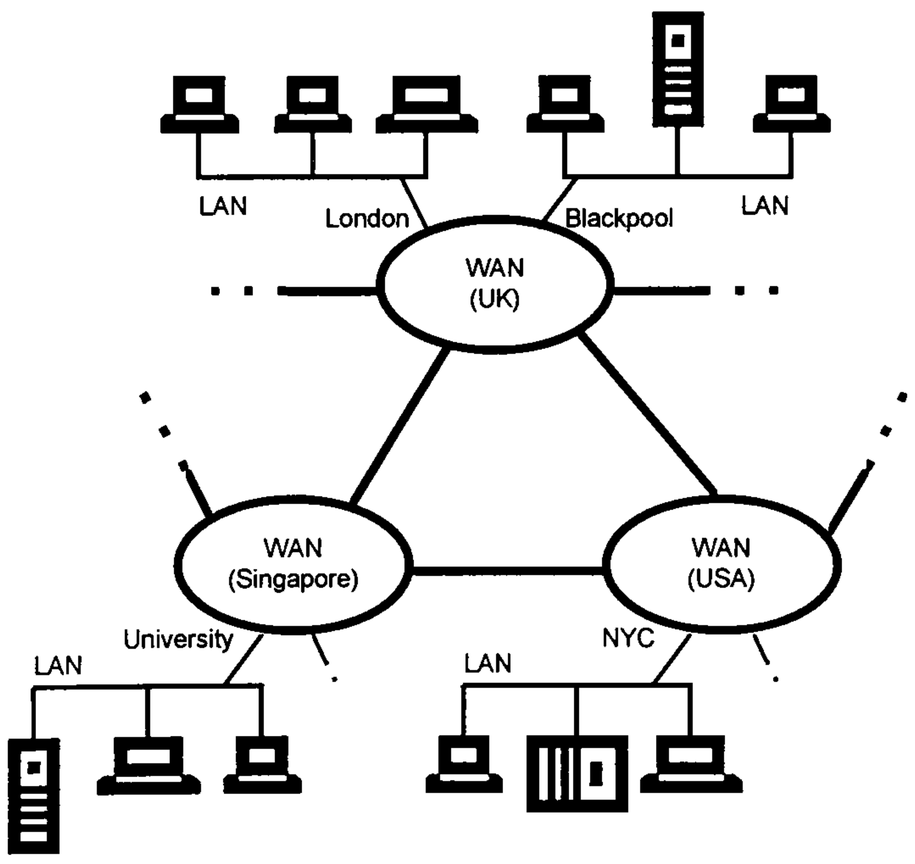


Figure 11: The Internet: a connection of networks

The Internet was first used by scientists who, having a real need to communicate with each other, used the network to speed up the pace of scientific research enormously. In the 1990's the Internet's growth has been ferocious. The introduction of hypermedia systems such as WWW has made it easy for people to find their way around the Internet. The WWW acts as a friendly face to the Internet and provides links that act as signposts, pointing to any other Internet computer, anywhere in the world. The WWW is now the preferred method of presenting and accessing information on the Internet. The Internet is no longer used by a few dedicated researchers but is now a commercial success with billions of pounds of annual investment. As a consequence new applications have been developed to use the Internet - Internet telephone, chat, newsgroups, video conferencing and, still in its infancy, the transmission of independent television channels. The Internet has become the biggest communication tool and the most exciting piece of information technology in the world.

2.3.2 Intranets and Extranets

The Internet is built around standards, rather than a product from a particular vendor. This has led to its universal acceptance and the development of software or further standards that makes it extremely easy to send and receive information without understanding the underlying technology. These applications include WWW, Mail, Newsgroups, etc (listed in Table 7). This universal acceptance has led to prolific growth in the numbers of users of the Internet. As a consequence, the amount of information available on the Internet, and the connectivity the global network provides, is growing at a fantastic rate. In addition, the applications to which the Internet can be put to is growing at an equally fast rate.

These factors have encouraged many companies or establishments to incorporate the facilities designed originally for the Internet to be used on their own local internal networks. These networks are called *intranets*. An intranet is a network based on TCP/IP protocols (the universally accepted Internet protocols) belonging to an organisation, usually a company, accessible only by the organisation's members, employees, or others with authorisation. An intranet's Web sites look and act just like any other Web sites, but security mechanisms are in place to prevent unauthorised access. Like the Internet itself, intranets are used to share information. However, the information is likely to be more relevant to the specific establishment, such as phone lists, health and safety information, and online training. Intranets are growing in popularity, because they are much less expensive to build and manage than private networks based on proprietary protocols. This is especially so for establishments with a wide range of computer types, and a variety of network types. A further advantage of employing an intranet is that the software tools that are used on the network, such as web browsers and email, are identical to those used on the Internet. People who are already using the Internet, will know how to use the intranet, and vice-versa.

An *extranet* is an intranet that an organisation has opened up and shared with selected authorised users outside the organisation, providing various levels of accessibility depending on username and password authentication. It is different from an intranet in that an intranet is only accessible by people who are members of the same company or establishment. An extranet could be used to allow a company to access another company's intranet to place an order.

3 Overview of the Design

This chapter examines the main components of the generic monitoring Management Information System (MIS) and then proceeds to investigate the underlying technologies that were considered or used in its construction. At the very highest level the system can be split into two parts: the CBT application itself and the MIS that communicates with it and generates the reports. This is illustrated below in Figure 12.

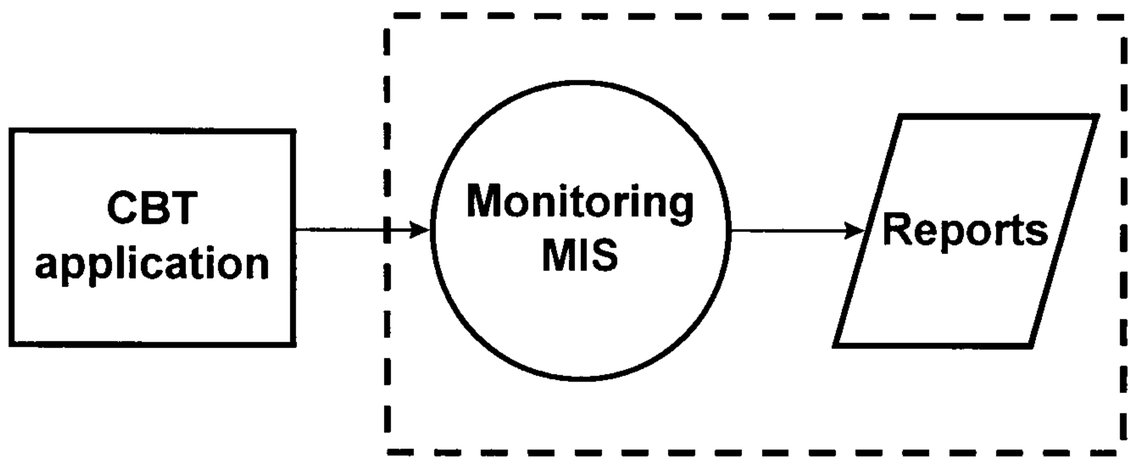


Figure 12: Highest level monitoring system overview

If issues such as distance learning are considered, wherever the CBT application and the monitoring MIS are separated by a network, it is quite clear that some form of delivery mechanism needs to be added so that information can be delivered from the CBT application to the Monitoring MIS. The situation is further complicated when several CBT applications are using the system at the same time. There will be numerous records of information produced by each of the individual CBT applications. All of these individual records of information have to be merged together into a single entity, or a master database. This would allow a complete picture of CBT usage to be painted. Finally, some means of interrogating this database would be required to yield the end product - a report

acceptable for the eyes of a training manager. Bearing these issues in mind the system represented by Figure 12 can be expanded to that of Figure 13.

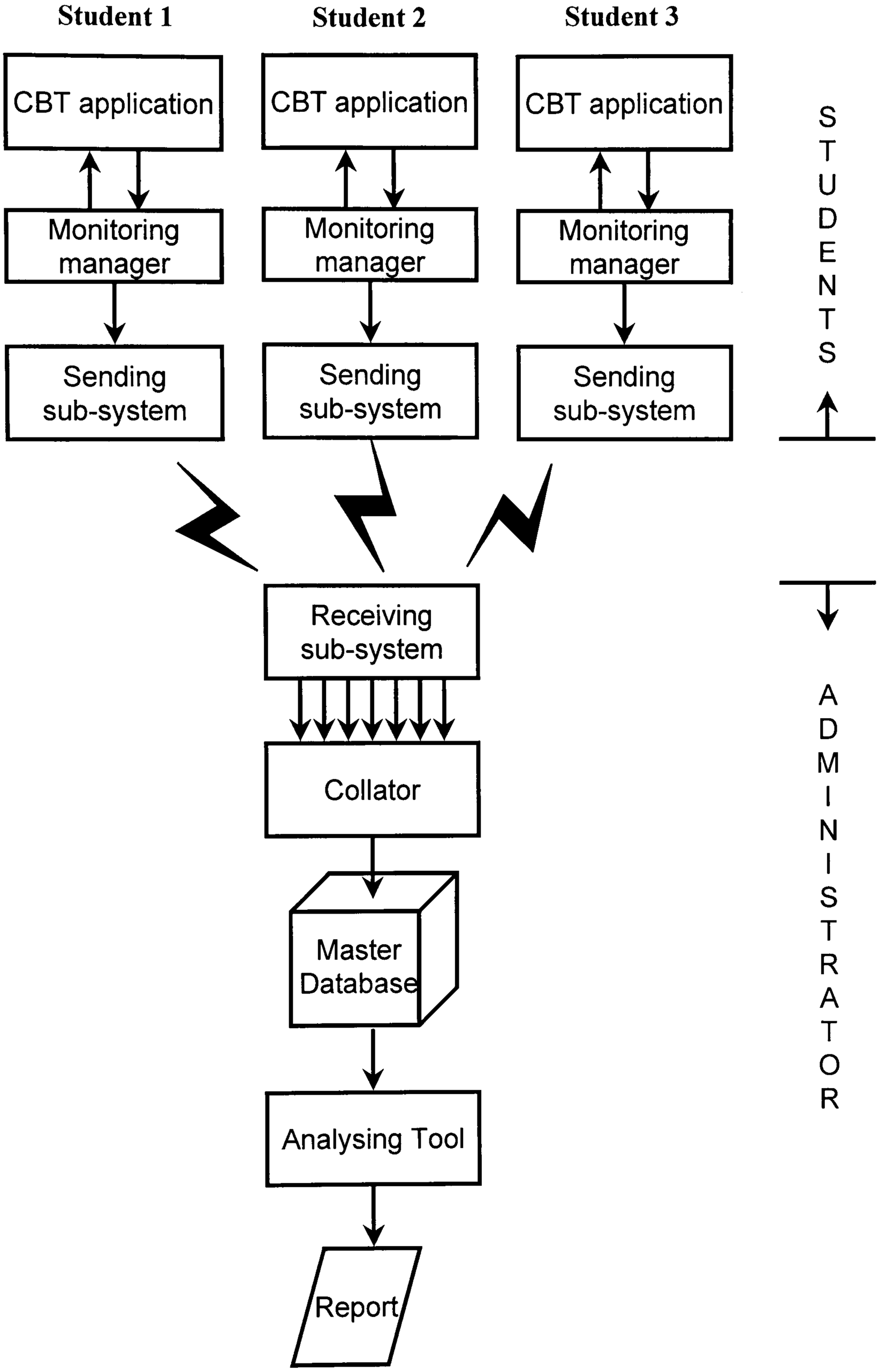


Figure 13: Monitoring system overview

The system shown in Figure 13 is split into two halves; the student (client) side and the administrator (server) side. The two sides may be geographically separated but are able to communicate with each other over some kind of network, such as the Internet. In any one implementation of the system there are likely to be many students being trained but only the one administrator. Also, it may be anticipated that any of the students may be geographically separated but have a connection to the same network.

The role of a student is to progress through the CBT material aware that their movements and actions are recorded, but blissfully unaware of how it is achieved or the underlying technology used. The role of the administrator is to use the analysing tool to extract meaningful information from the master database. On the student side of the network the CBT communicates with a 'monitoring manager' to report information externally. This monitoring manager is the sub-system which co-ordinates how, and where the reported information is stored. However, the author dictates the majority of *which* information is stored. This is because the monitoring requirements are specific to individual CBT packages. The author of the CBT application must build the capability into his/her product to enable communication with the monitoring manager.

In general, the monitoring sub-system has to execute reporting tasks passed to it by the CBT. The final component on the student side is the 'sending sub-system'. This system delivers the reported information to the administrator side at the CBT application's request. In a busy training scenario there will be numerous students training. As a consequence several deliveries would occur and some form of a receiving sub-system would be required to confirm successful receipt. The components of Figure 13 raise several technological

issues which are listed below and considered in the remainder of this chapter. These are summarised as:

- how does the CBT talk to the 'monitoring manager'?
- what file formats should be used to store reported information?
- how is the reported data transported over computer networks?
- how are the individual records of information collated into one?
- what are the resource implications of the above.

3.1 CBT to Monitoring Manager interface

For a CBT application to report monitoring/tracking and performance information externally it must have this capability built into it during the authoring process. For example (referring to Figure 13) the CBT must be aware of the existence of the 'monitoring manager' component, and also be able to communicate with it. If this is not achieved reporting will take place. The monitoring manager interface must, therefore, be accessible from within all the authoring tools that are used to write multimedia and training applications.

Authoring tools are programs that enable an author to create a final application merely by linking together objects, such as a paragraph of text, an image or a piece of video. The author (who uses authoring tools) defines the relationships between the objects and sequences them in an appropriate order. This system allows people with little programming knowledge to produce attractive and useful graphical applications. Many authoring tools quite often also support some form of scripting language for more sophisticated operations. Authoring tools, often considered as 4th generation programming languages, are distinguishable from ordinary programming languages because they require

less programming knowledge to master, and also because they are used exclusively for applications that present multimedia.

There are currently a wide variety of different authoring tools used to develop CBT. If the monitoring system is to be widely accepted, a generic communication method must be selected that can be used by the majority of the available tools. If this is accomplished the monitoring interface will be able to be incorporated into any CBT application designed with just about any development tool. Of the wide variety of authoring tools available the 1997 CBT Report [17] selected six products that currently dominate a field of more than 34 choices. These are listed in Table 8. These results were obtained by administering a questionnaire asking the respondents (CBT developers) to indicate which authoring tools they were using. Each respondent was allowed to tick a maximum of three choices and, in all, there were 1931 boxes ticked. Table 9 shows the market share achieved by each of the major authoring toll vendors [9].

Authoring Tool	No. in use	Product Manufacturer
Authorware	424	Macromedia
Toolbook	266	Asymetrix Corporation
Director	192	Macromedia
Quest	183	Allen Communications
Instructor	179	Asymetrix Corporation
Designer's Edge	136	Ask International
Other	116	-

Table 8 : Six most-used CBT authoring systems

Tool developer	Development tool	Market share
Macromedia	Authorware, Director	45%
Asymetrix	Toolbook, Instructor	47%
Others	-	8%

Table 9: Authoring tool developer market share [9]

From Table 8 and Table 9 it can be determined that the market is dominated by two companies, Macromedia and Asymetrix. In addition 92% of all CBT applications are developed using the four most popular Authoring tools listed. Of these tools Authorware (by Macromedia) currently has, and has always had, the largest installed base of users [10]. The significance of this is that the monitoring manager, in order to be considered generic, must be capable of successfully interfacing into Authorware, Director, Toolbook and Instructor. In addition, further flexibility would provided if the monitoring manager could also integrate applications developed in C, C++ and Visual Basic. The issues in designing the CBT to monitoring manager interface are summarised below:

- the interface should not dictate which authoring tool is to be used
- the interface should be made as easy as possible to use (authors are normally subject matter experts with limited programming expertise)
- the interface should be easy to update for an installed CBT application

3.1.1 Monitoring Manager Design Strategy

The objective of the monitoring manager interface is to add monitoring functionality to the CBT application. The monitoring manager itself must remain external to the CBT application so that it can be re-used as a generic tool, updated if need be, and incorporated into any further CBT applications produced. As a consequence the monitoring manager has to be a separate function or process that communicates using a predetermined interface with the CBT application. This is illustrated in Figure 14.

The vast majority of CBT applications are designed to be delivered on Microsoft's Windows 3.x, Windows 9X, or Windows NT operating systems. As a consequence communication methodologies and functional re-usability between windows applications needs to be examined. Microsoft's capacity to develop Windows into the operating system it is today arose from Intel's introduction of protected mode operation in the 8086's successors [11, page 10]. This led to the definition of way for applications to share information through a common piece of memory. This interface was called DPMI (Dos Protected Mode Interface). This is a very low level interface, so DPMI was encapsulated by a higher level interface DDE (Dynamic Data Exchange) for use with windows. DDE was the first technology to allow Windows applications to communicate with each other and formed the essential building blocks for Microsoft's COM (Component Object Model), first as OLE (Object Linking and Embedding), and eventually as ActiveX.

The challenge presented using DDE, OLE or ActiveX to facilitate communication between the CBT and the monitoring manager is determining the details of the CBT applications interface. DDE is limited because an application has to know the details of an object's DDE interface in advance in order to call on the facilities it provides. This means that the monitoring manager must be aware of all possible CBT applications (or rather the tools used to create them) and their individual interfaces. OLE and ActiveX solves this problem by forcing applications to publish their interface, so the calling program can learn how it is used. However, an extra level of complexity would be required in the monitoring manager to accomplish this and even more to make the monitoring manager's facilities fully integrated into the authoring tool.

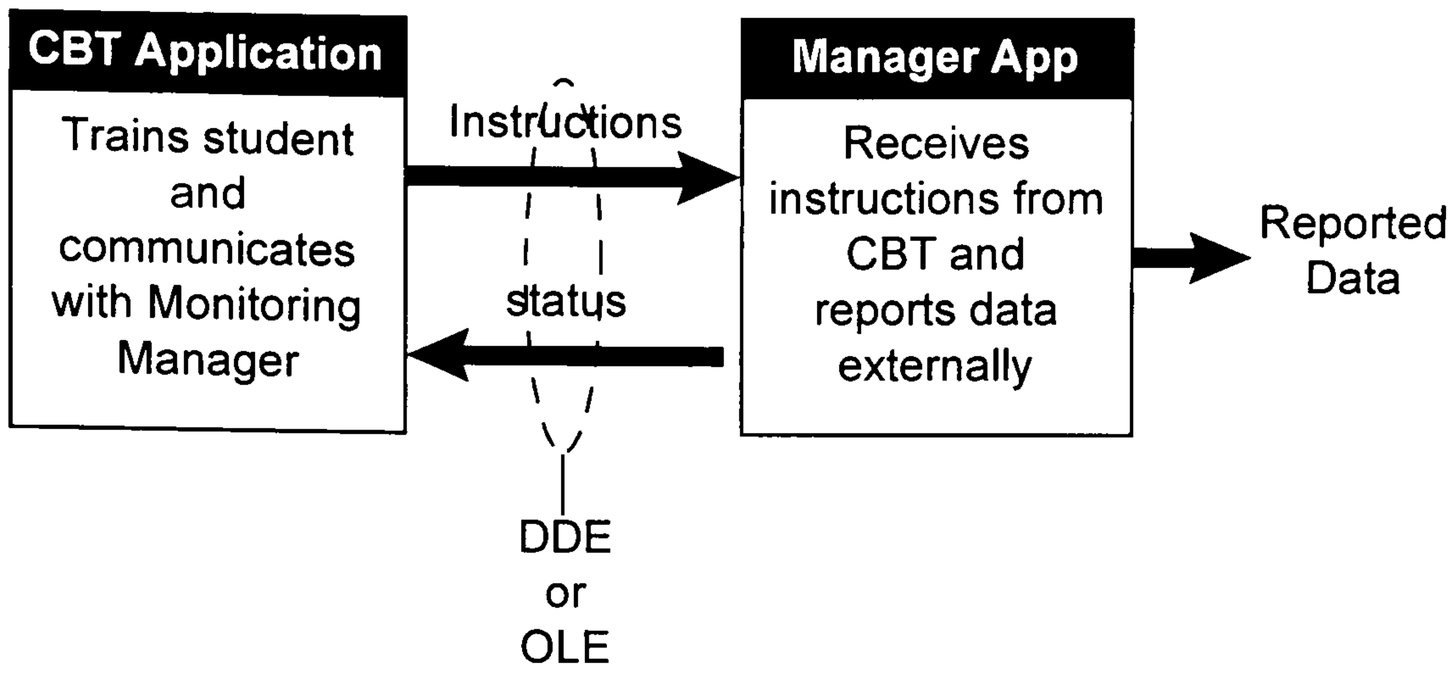


Figure 14: Application implementation of Monitoring Manager

An alternative to designing the monitoring manager as a separate windows application (that communicates with the CBT using the methodologies discussed above) is to develop it as a DLL (Dynamic Link Library). DLLs are one of the most important structural elements of Windows [12, chapter 19]. Most of the files associated with Windows are either program modules or dynamic link library modules. DLLs are generally not directly executable, they are separate files containing functions that can be called by programs and other DLLs to perform certain tasks. A dynamic link library is brought into action only when another module calls one of the functions in the library.

One purpose of dynamic link libraries is to provide standard functions and resources that can be used by many different programs thereby saving memory and disk space. This is demonstrated in Figure 14, where several applications can be seen using functions and resources in a single DLL [14 chapter 15.2.2].

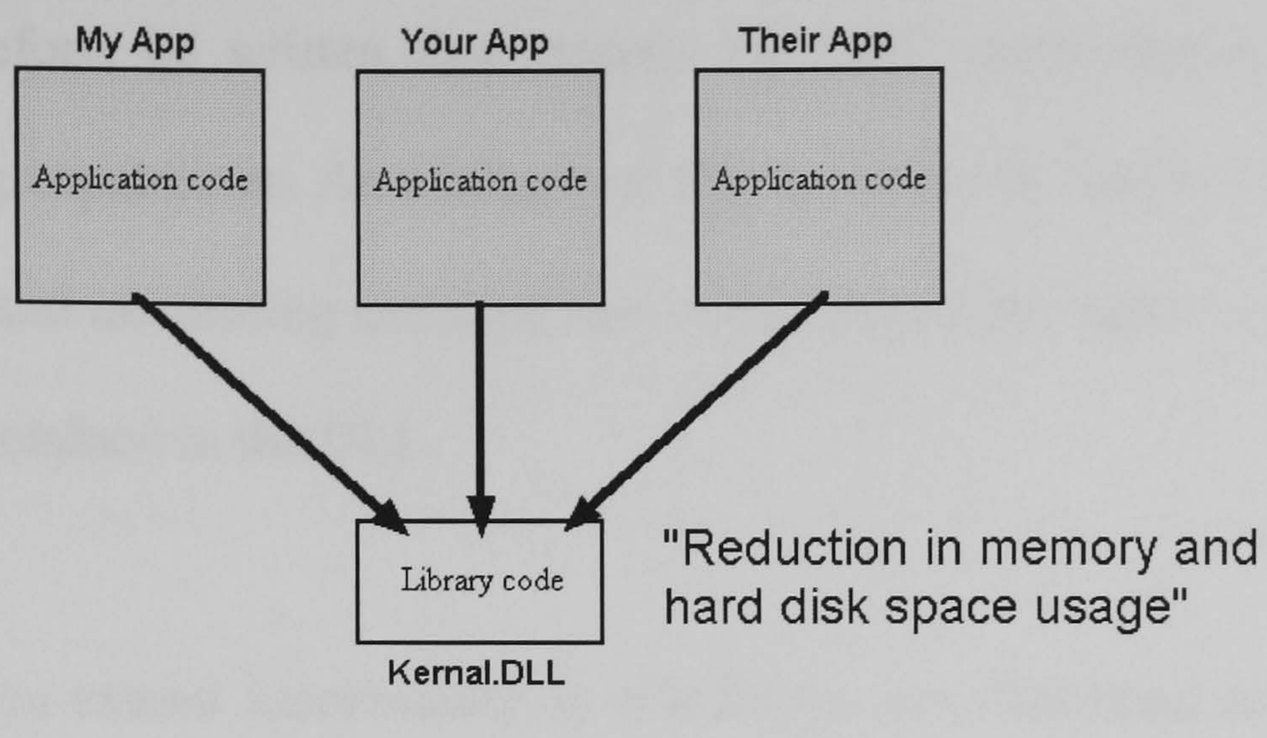


Figure 15: DLLs: Used as shared componenets

In addition, DLLs (including those that make up Windows) can be thought of as extensions to a program. Programs can use DLLs to load extra resources or functionality at any time, based upon user preference. For example, a program may access text strings within a different DLL, depending on which language the user chose upon entering the application (Figure 16). This also allows a programmer to perform incremental upgrades, or perhaps add extra functionality, which had not originally been included, or not thought of as being needed, when the main calling application was compiled.

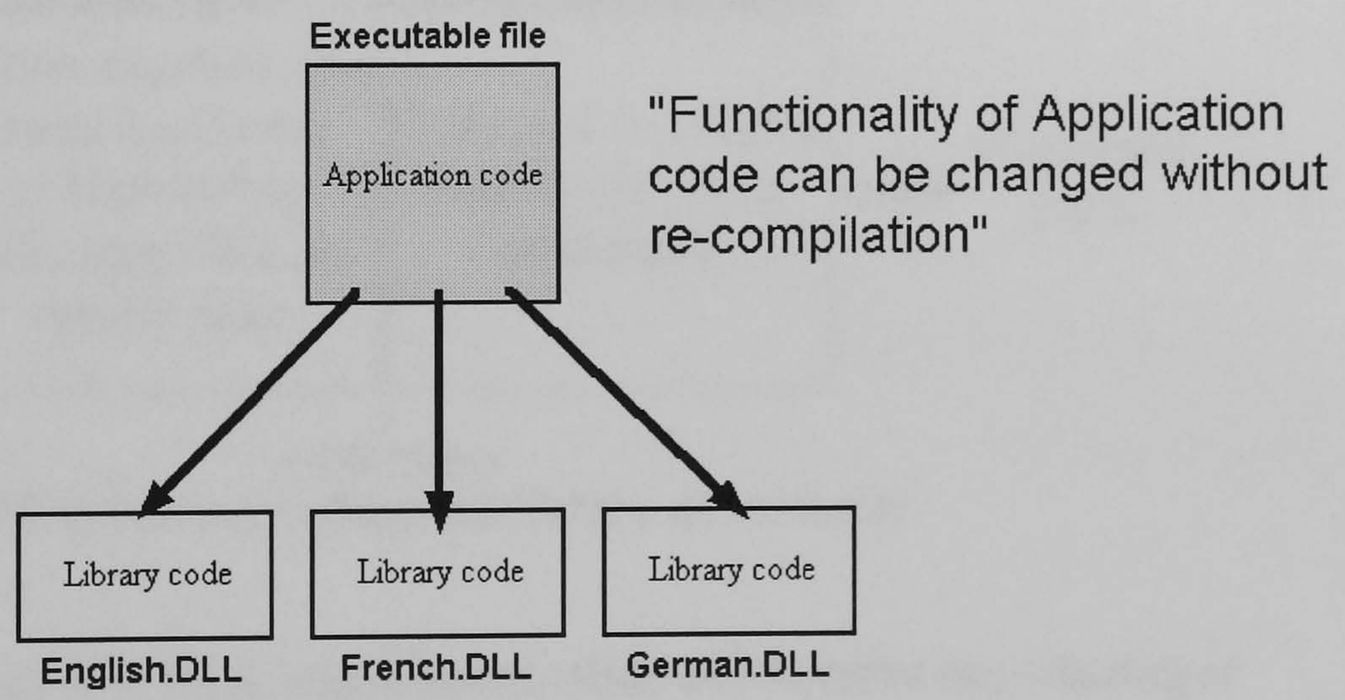


Figure 16: DLLs: Used to change functionality without recompilation

A DLL can, therefore, be written that extends the CBT application's functionality to include monitoring capabilities. An example of this is shown in Figure 17. The interface between the CBT and monitoring manager then comprises of the input / output parameters of the functions contained in the DLL.

The use of DLLs to extend functionality is effectively an established procedure, and an open standard. As a consequence, all the authoring tools listed in Table 9 provide facilities to extend functionality via DLLs. This means that it is entirely possible to write a DLL that can be used in just about any popular authoring environment in use today. However, care must be taken to provide a level of abstraction from the low level functions in the DLL, as it is very different from the GUI interface the authors are familiar with. Although DLLs are not software applications, they are still viable products in their own right and can be licensed for use by third party software developers.

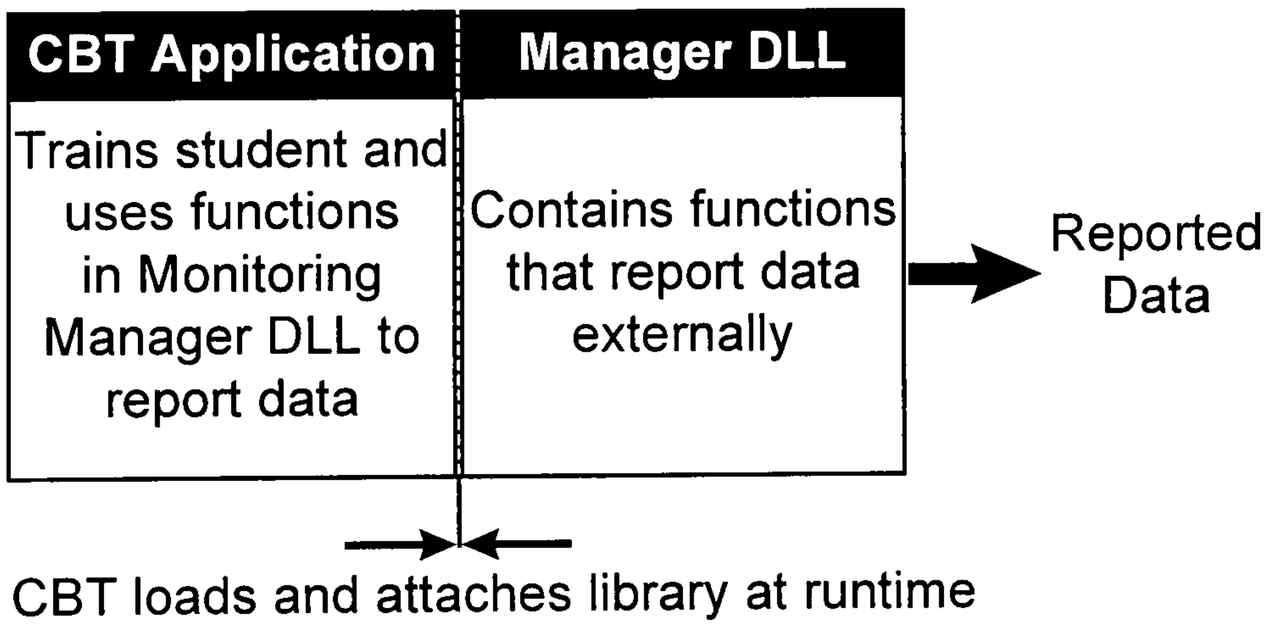


Figure 17: DLL implementation of Monitoring Manager

3.2 *Identifying Student*

A key requirement of a monitoring system is to establish the user's identity. Without the user's identity being known, any information reported cannot be traced back to them. If this was the case, many of the benefits of employing a monitoring system would be disabled. With the users identity unknown, specific user performance could not be retrieved, corrective action to assist struggling users could not be deployed and users would not feel accountable for their actions. An example of the sort of information that would be useful to obtain about a users identity is listed in Table 10.

User information
Unique user identification
First name
Surname
Department
Training location
Supervisor name

Table 10: Useful Login Information

Every student or employee will require, and will probably already have, some form of unique identification. This could be a staff number, student number or a network login username. This requirement is mandatory. Other information that would be useful to obtain about the user, but not mandatory, is their full name, which department/group they are in, where the training took place and the name of their supervisor. The location of the training would be useful data item to obtain as it could be used to determine how the CBT is used in a distributed environment, or perhaps ascertain seating positions in an assessed exam. The name of the user's supervisor could be used to automatically submit reports.

perhaps via email, directly to their line manager. This sort of 'user data' can be retrieved by either the CBT application itself (through the use of a login script), the monitoring system (automatically) or perhaps even both.

3.2.1 Login Script

The most common way used to identify a user is to deploy a login script. A login script is a form presented at the beginning of the session, which the user must fill out before being allowed to continue. A CBT application can provide such a script for students to fill out prior to their training commencing. The user details obtained from the login script can then be passed by the CBT to the Monitoring system. Figure 18 shows an example of such an login script. An advantage of using a login script is that each CBT application can tailor the information gathered specifically. For example, some applications may require a date of birth, whilst others would not. However, a disadvantage of using a login script is that it is labour intensive for the user to fill in their details every time a new module of CBT is started.

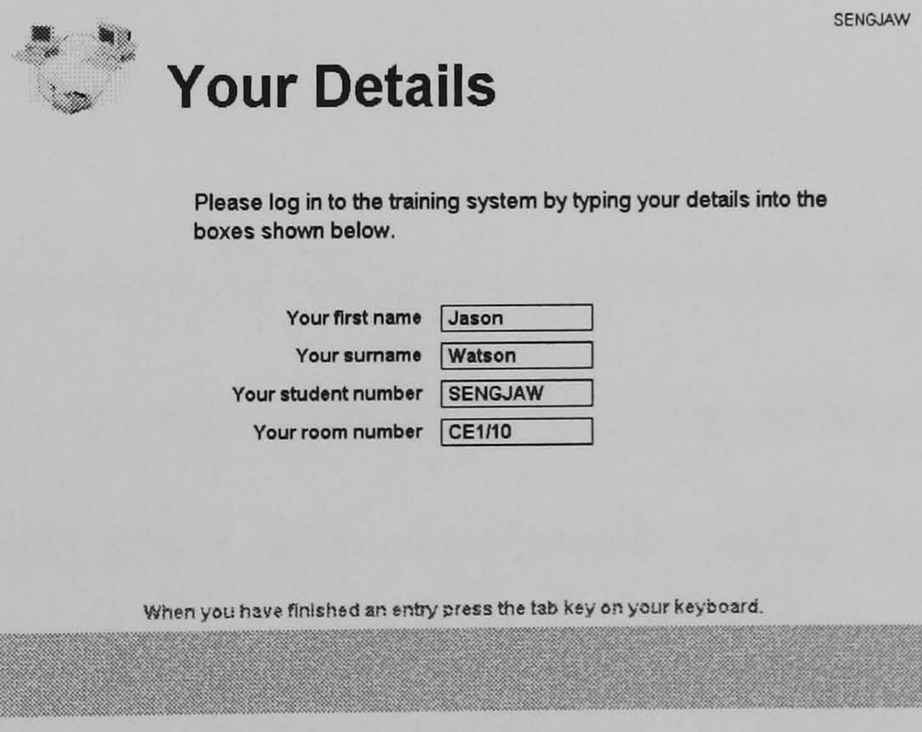


Figure 18: Example of Login Script

Login scripts can be considered quite a redundant process if users have previously identified themselves, such as to a network or computer. It would be more efficient to retrieve any details obtained previous, directly from the network or computer system, so that duplicate questions do not have to be asked. This would have the added advantage of authenticating the users identity, as a network or computer login would, almost certainly, have required a password.

3.2.2 Automatic Identification from previous login

Login scripts are useful method to obtain specific information about a user. However, general information, such as unique usernames, can normally be obtained directly from the computer system. Most computers connected to a network authenticate a new user by asking them to input a unique username and a password at the start of their session. It is, therefore, possible for the monitoring system to communicate with the network in order to acquire this information. The information can then be stored directly in a database, or passed back to the CBT application where it could act as pre-filled parts of a login script, or confirmation dialog.

The monitoring system must have knowledge of the networks Application Programmable Interface (API), in order to communicate with, or obtain information from, the network. Unfortunately network APIs differ from one network vendor to the next. IDC Research [18] has performed an analysis of Network Operating Systems (NOS) current use. The

results of this research are shown in Figure 19 where it can be seen that 81% of all NOS are provided by Novell or Microsoft.

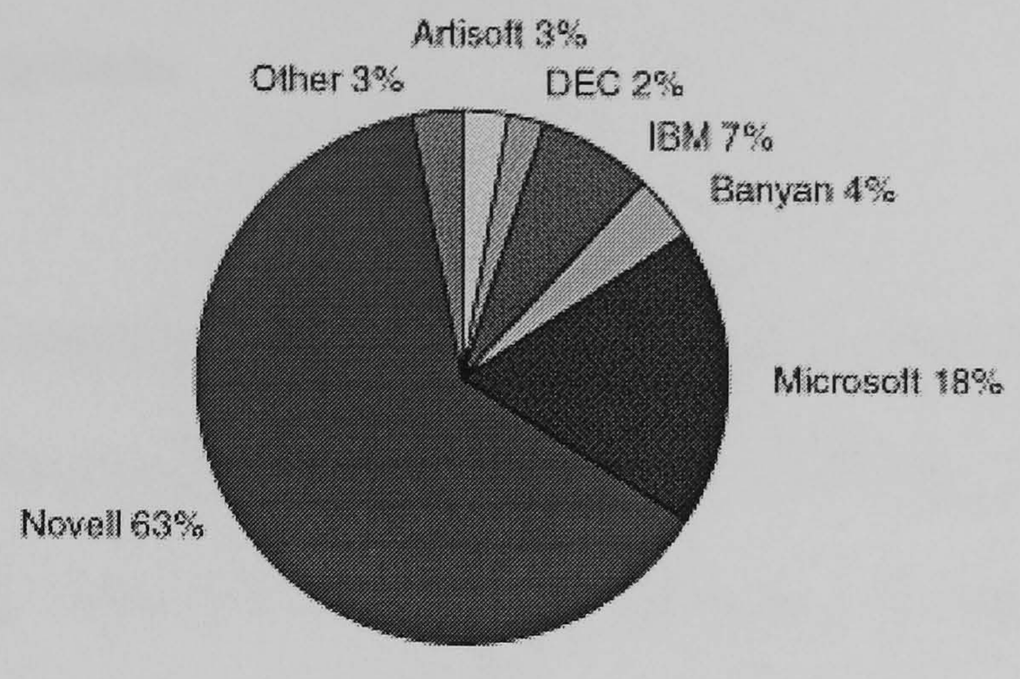


Figure 19: Network Operating System market share 1997, IDC Research [18]

If a monitoring system can obtain user information from either Novell or Microsoft networks, then the vast majority of users can be automatically authenticated. It has already been established that the location at which the training took place is also a desirable data item to obtain. The location at which the user undertook the training is actually the identity of the computer which the user used to access the training. This information can be obtained automatically from the computer network. Most networks today support the use of the TCP/IP protocol, the protocol that is used on intranets. This means that most computers will have been allocated an IP address, an identifier for a computer or device on a TCP/IP network.

The format of an IP address is a 32-bit numeric address written as four numbers separated by dots. Each number can be zero to 255. For example, 161.112.116.43 could be an IP address. An IP address can be used as a unique identifier for the location the user trained at. The IP address of a computer can be obtained by writing software to access an external

software component called 'Winsock'. All networks supporting TCP/IP will already have the Winsock software component installed on their clients.

3.3 Delivery sub-system

In a LAN training environment, reported data (such as that produced in Figure 17) can be written in a common place prior to analysis. However, if a training scenario consisting of several geographically separated students is considered a fundamental problem is encountered. None of the separated components (the students, or the training manager) share resources. An alternative method is required to transport the reported information to a centralised point. This could be done in a variety of different ways; for example reporting the information directly onto a floppy disk and posting it to a manager. The simplest and easiest way (from the perspective of the student and training manager) to address this problem is to use the network to transport the reported information seamlessly and effortlessly.

This concept is illustrated in Figure 20, where students training throughout the world have their performance and monitoring information transmitted over the Internet to a centralised training server. This information can then be queried and analysed by a training manager at his/her leisure. In this system students are able to train and have their actions monitored at any place (assuming an Internet connection exists) and at any time (assuming the training server is online).

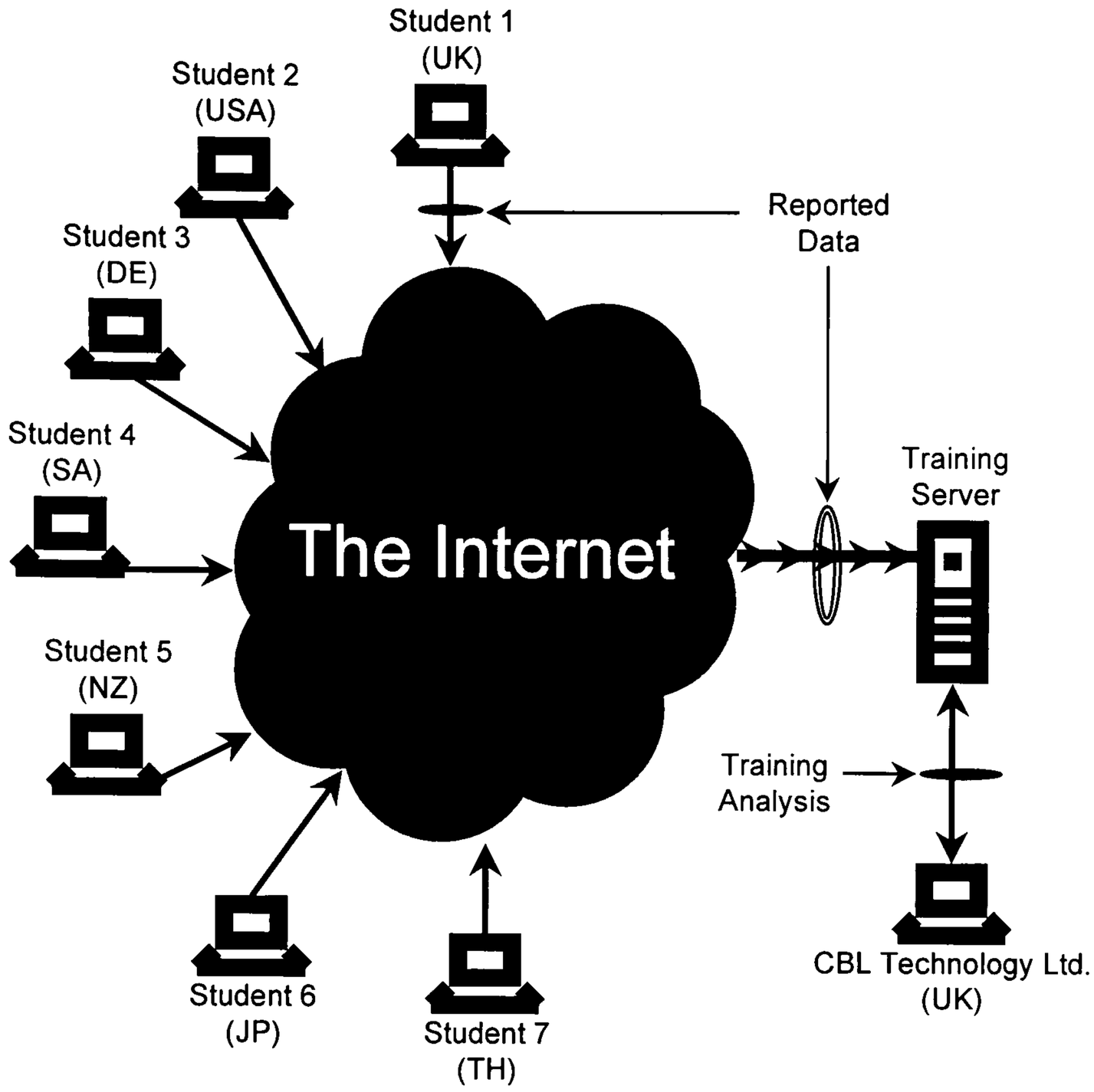


Figure 20: Distributed training scenario

Some of the standard mechanisms that can be used to transfer data from the client to the server, over the Internet, have already been described in Table 7. These include Email and FTP. There are a variety of other ways of transporting information over the Internet, however, they comprise of less popular or proprietary systems that are quite often prohibited for security reasons. Many students will not have a direct connection to the Internet and will instead connect via another network (such as a LAN) and be limited by that network's security restrictions. In the computer industry a mechanism known as a *firewall* is used to separate internal networks from the insecure public Internet. A firewall is a combination of hardware and software that acts as an electronic gatekeeper, protecting a network from outside intruders.

A firewall basically stops each data packet, vets it, and allows it to pass only if it meets specified criteria. That criteria often restricts Internet traffic to the more standard data transfer mechanisms, such as HTTP (WWW), SMTP (Simple Mail Transfer Protocol) and FTP (File Transfer Protocol). As a consequence, any delivery system employed by the monitoring system needs to meet the same criteria by avoiding the use of proprietary systems which could cause firewall tunnelling problems. The CBT monitoring system must comply with existing network security settings rather than dictate what they should be.

3.3.1 Transferring reported data using Email

Email could be used to transport the reported data from the student terminal to the central administrators site. A major feature of Email is that transportation of data via this method is asynchronous. This means that information can be sent, even if the server is not ready to receive it. Other methods would require server software to be running continuously at the administrators end, waiting for the data.

The use of Email to transport the data records may be seen as a very flexible way but does have several drawbacks. There are many different types of mail systems in use today (Microsoft Exchange, Lotus Notes, etc…) all of which use their own proprietary formats. As a consequence, the design of the interface between the CBT and the mail system could get extremely complex. This is because all the popular mail systems would have to be catered for to make the system widely acceptable. In addition, many organisations do not provide all of their training workstations with Email capabilities. This means that the

delivery mechanism cannot depend on all systems having access to e-mail. The extra installation of any client software must be avoided at all costs, due to the considerable expense (there could be thousands of clients). Further complications arise with the use of Email that depend on the type of information that is going to be transmitted. Most Email systems are limited to the transport of text only - i.e. 7 bit data. Therefore binary data (such as databases, executables, etc…) must first be encoded into a 7 bit format before they are transmitted.

3.3.2 Transferring reported data using FTP

File Transfer Protocol (FTP) could also be used to transport the reported data from the student terminal to the central administrators site. FTP is a standard method of transferring files (of any format) across the Internet in a reliable and secure fashion. If FTP was used it would require the delivery mechanism to be able to simulate an FTP client, and the receiving administrators site to be continuously running the FTP server software. FTP is such a widely used protocol that there are many public domain code libraries that facilitate the client functionality. An issue could arise, with the use of FTP, if the FTP server was not operational at a time when data must be transmitted. This could be overcome by allowing data records to be stored locally and transmitted at a later date, should FTP be temporarily offline.

3.3.3 Transferring reported data using a proprietary system

It is entirely possible for a proprietary type of delivery protocol to be implemented to transport the reported data over the Internet. Such a system would most likely use Windows-Sockets. Windows-sockets facilitate the transmission and the receipt of data over TCP/IP networks (the type of networks used by the Internet). The development of a proprietary system could improve overall efficiency, and perhaps remove the need for the data to be stored locally. However, this would require the complete design of both the client and server software which is outside the remit of this project. In addition, network managers are reluctant to have proprietary systems installed on their existing systems, and such systems could be prone to installation difficulties due to firewall tunnelling issues.

3.4 Record collation

The ultimate objective of the whole system is to facilitate the generation of reports for the CBT that has been monitored. To allow this a single database for the CBT application being monitored must be produced. This database can then be queried and manipulated to generate useful training information in the form of reports. However, due to the distributed training environment the reported information is not initially contained in a single database but fragmented into small pieces. This fragmented data must first be merged together into a single entity by a 'collator application' to facilitate analysis.

The 'collator application' is demonstrated in Figure 21 where several data fragments (mini databases) are received from over the Internet and directed to an application that merges them into a single master database. It can be seen that it would be very advantageous for the 'collator application' to have the ability to read and write databases of a popular format. The input data records (received from over the Internet) do not necessarily have to be in a known database format, However, if the master database was in a recognised format (such as Microsoft Access) no additional software would have to be developed to produce the end reports. The standard facilities provided by the commercial database application for producing queries, calculations and reports would suffice.

Having established that the collator application must have the ability to read and write in a popular database format we shall now investigate ways in which this functionality can be achieved. The ability for the collator application to read/write a variety of different database formats would be an extremely desirable feature. The impetuous behind the design of the Generic Monitoring System is *flexibility*. If the customer, or training manager is used to using a specific database format, then the system should accommodate that format. The obvious dilemma with this situation is that many companies use different types of databases. Each of these different databases use a proprietary interface specific to that database.

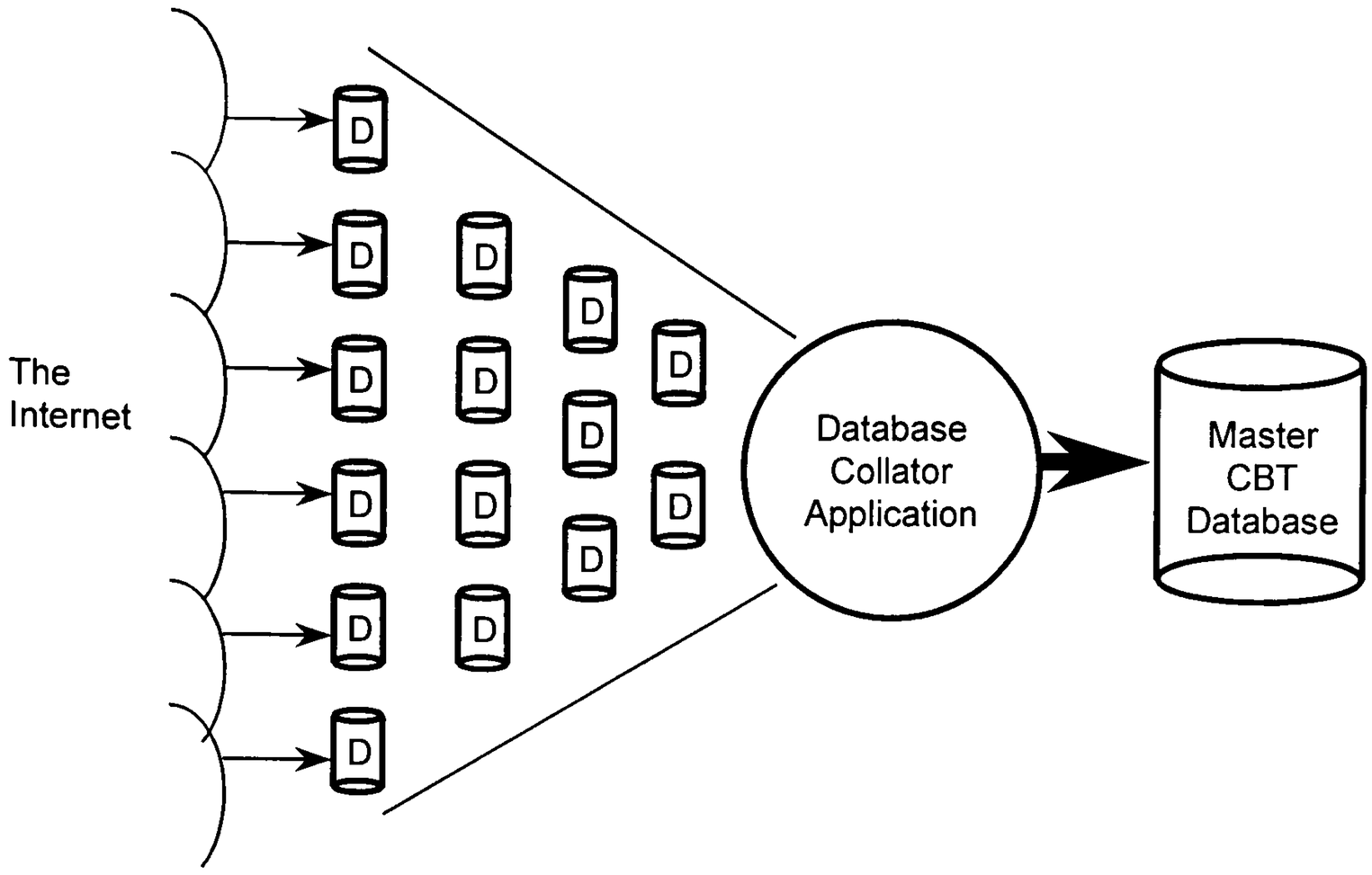


Figure 21: Database Collation Procedure

If the collator application is to provide heterogeneous data access (access to data in more than one data source), it will require different code to be developed to interface to each data source. Applications written in this manner are difficult to code, difficult to maintain, and difficult to extend. Fortunately, Microsoft have developed Open Database Connectivity to overcome these difficulties.

3.4.1 Open Database Connectivity

Microsoft developed the open database connectivity (ODBC) interface as a means of providing applications with a single application programming interface (API) through which to access data stored in a wide variety of database management systems (DBMSs). Prior to ODBC, applications written to access data stored in a DBMS had to use the proprietary interface specific to that database.

The ODBC architecture has four components [13, ODBC SDK 2.1]:

- **Application**. Calls ODBC functions to submit SQL statements and retrieve results.
- **Driver Manager**. Loads drivers on behalf of an application.
- **Driver**. Processes ODBC function calls, submits SQL requests to a specific data source, and returns results to the application.
- **Data Source**. Consists of the data the user wants to access and its associated operating system, DBMS, and network platform (if any) used to access the DBMS.

This architecture is designed to permit maximum interoperability: It allows application developers to create an application without targeting a specific DBMS. End users can then use the application with the DBMS that contains their data by adding modules called database drivers, which are dynamic-link libraries (DLLs) available from the database vendor or third-party driver developers. The ODBC interface allows applications to access data in DBMS using Structured Query Language (SQL) as a standard for accessing data.

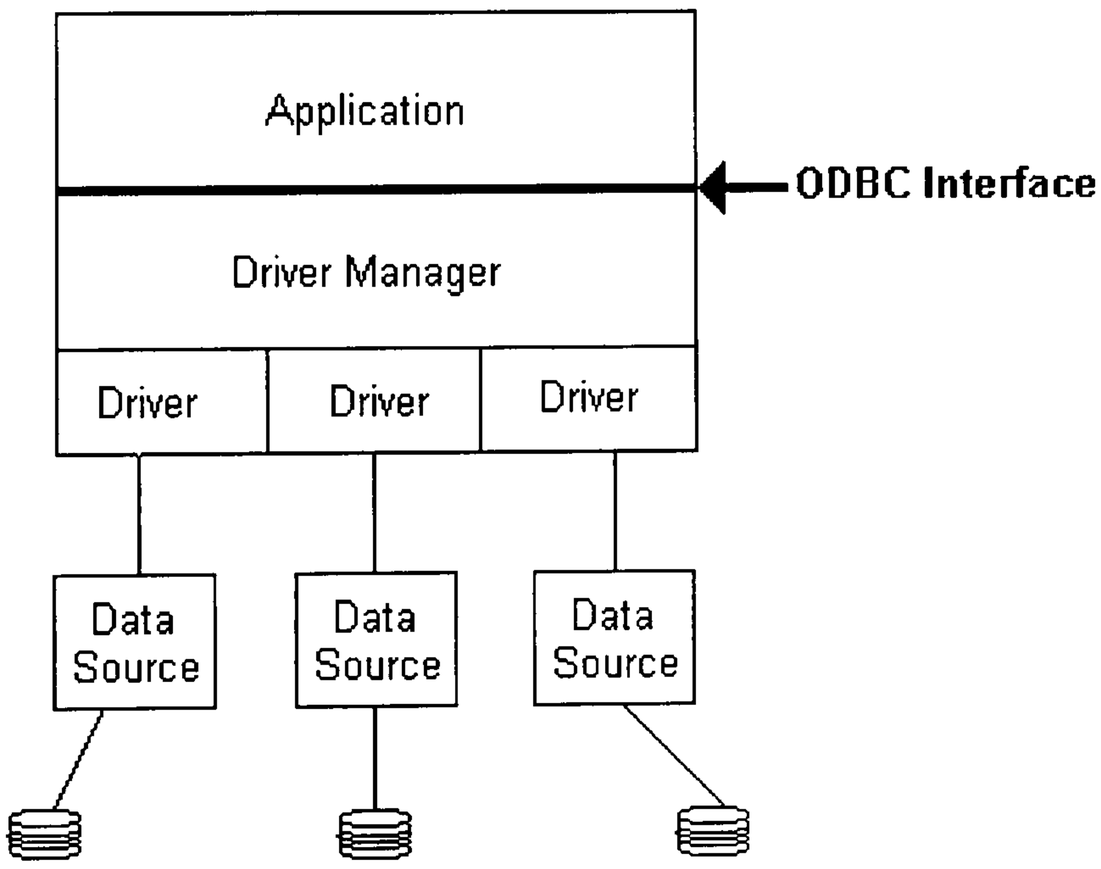


Figure 22: ODBC Components

The Driver Manager and driver appear to an application as one unit that processes ODBC function calls. Figure 22 shows the relationship between the four components. Through the use of ODBC the collator application would be able to create and manipulate any of the popular databases available today. The application could be written to communicate with the ODBC driver manager, using standard ODBC function calls and SQL, and be detached from the actual format of the data source itself. Microsoft Windows 95 comes with ODBC already installed and includes standard drivers such as; Microsoft Access, Dbase, FoxPro, Excel… and more.

3.5 Production of reports

If the output of the collator is in a standard database format, then the specific tools that this database provides should be sufficient to facilitate the end reports. Applications such as Microsoft Access come with built-in wizards that aid the creation of its reports. The reports themselves are to provide feedback to the training manager about the performance of individuals or the performance and effectiveness of the course itself. The reports would include information such as which pages have been visited, when they last viewed the course, the results of any scores or interactions and perhaps any comments from the student.

If the system is used to train a very large amount of students, the type of reports that database management systems produce may not be the most suitable, and alternative methodologies may have to be investigated. However, holding the reported data in a

popular format should ease access and integration should it be deemed that a purpose-built visualisation tool has to be developed.

3.6 Conclusion and Recommendation

From this analysis, it was decided that the most interoperable way to report data from a CBT application was to develop the reporting functionality in the form of a DLL that employed ODBC to write data in any popular database format. A DLL solution will ensure compatibility with all current and future authoring development systems, and the use of ODBC will allow the customer to use any Database Management System (DMS) to Query data and create reports.

The optimum delivery mechanism for this data is FTP, as it is the most secure and widely accepted way of transferring files over computer networks. In addition an FTP server will be able to automatically handle operations from concurrent users of the monitoring system. The use of FTP is a straightforward, robust, procedure, and will not cause any concern to network managers like a proprietary system may. In addition, as well as FTP being compatible with network security devices, it uses TCP/IP protocols and is fully capable of being implemented wherever these protocols are installed: whether on a LAN or the WWW. One aspect of this delivery mechanism is that the FTP server must be fully up and operating in order for the data to be received from the reporting system. This creates a weak link in the chain, where data could be lost if the server is not running or if too many concurrent users are logged in. The reporting system must be designed to recover from

this situation, in order to minimise data loss, by being able to operate as normal in the absence of the FTP server.

It has also been identified that a separate collator application will be required in order to merge results obtained from individual sessions into a master database. The collator will also need to use ODBC in order to read the transported files, and to ensure interoperability with all popular database formats. This will mean standard DMS wizards and functions can be employed to query and analyse the data stored in the master database.

4 Implementation of the Reporting system

This chapter describes the steps that were taken in the design of the reporting part of the generic monitoring system. It first analyses the reporting system from a user point of view and then proceeds to describe each of the components, and how they operate. Finally, different examples of how the system was tested and employed are discussed.

4.1 The role of the Author

Authors are people who develop or "author" multimedia or training applications. This type of authoring can be thought of as "programming by non-programmers", as authors are usually subject matter experts with little programming expertise. Authors use high-level, drag and drop, authoring tools to produce training applications. Authoring tools, unlike programming languages, require less technical knowledge to master and are used exclusively for the production of multimedia and training applications.

In CBT the author of the training application is in fact the teacher, and it is these people who must utilise the reporting system so that the monitoring objectives of the CBT are met. This is achieved by the author integrating monitoring functionality into the CBT being authored, using the facilities provided by the monitoring interface. This is demonstrated in Figure 23. The kind of facilities that the author would expect from the reporting system are:

- Automatic recording of standard details such as; session start and finish times, identity of the host computer/student, identity of CBT application.
- The ability to report any variable. This facility would allow the authors flexibility to structure what was monitored and when.
- The ability to group and organise all reported data in a logical format.
- For all data to be automatically time and date stamped.

It is also important that the interface used by the author is not over-complex, and can be used by people with limited programming knowledge.

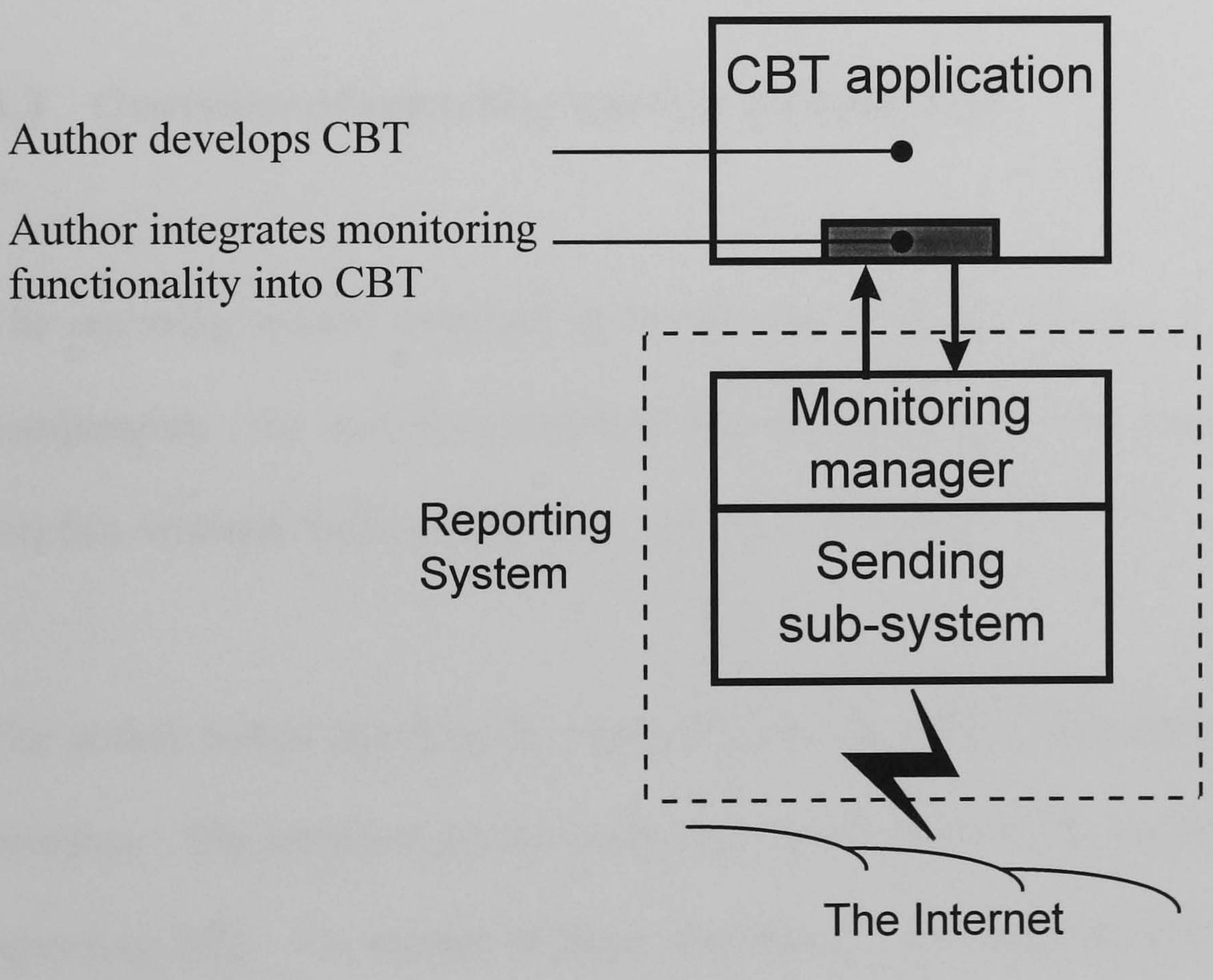


Figure 23: The Author Role

4.2 The role of the Student

The role of the student is to use the CBT application as intended, in order to receive training/tuition. The monitoring system should remain as seamless and transparent as possible to the student. The student will only know what was being monitored and recorded at the discretion of the author. The only additional activity that the student may experience is the log-in script. The monitoring system must use some method to obtain the identity of the student undertaking the training. Ideally this would remain hidden from the student if the system is able to obtain their identity from a previous network log-in. If this isn't possible, the CBT application must ask the student to identify themselves.

4.3 Overview of reporting system components

The reporting system overview is represented in Figure 24, and it contains the following components: the reporting interface, the reporting DLL, the reporting configuration (or ini) file, winsock DLL, ODBC DLL, and the FTP DLL.

The author builds reporting functionality into the CBT application through the use of the *interface.* The interface passes commands (determined by the author) from the CBT to the *reporting DLL.* On receipt of these commands, the reporting DLL then uses the ODBC DLL to create and manipulate a database. The structure (tables, fields, etc.…) of this local database is, in the main, dictated by these commands. The commands that the reporting

DLL receive from the CBT (via the interface) specify which variables the CBT reports and stores.

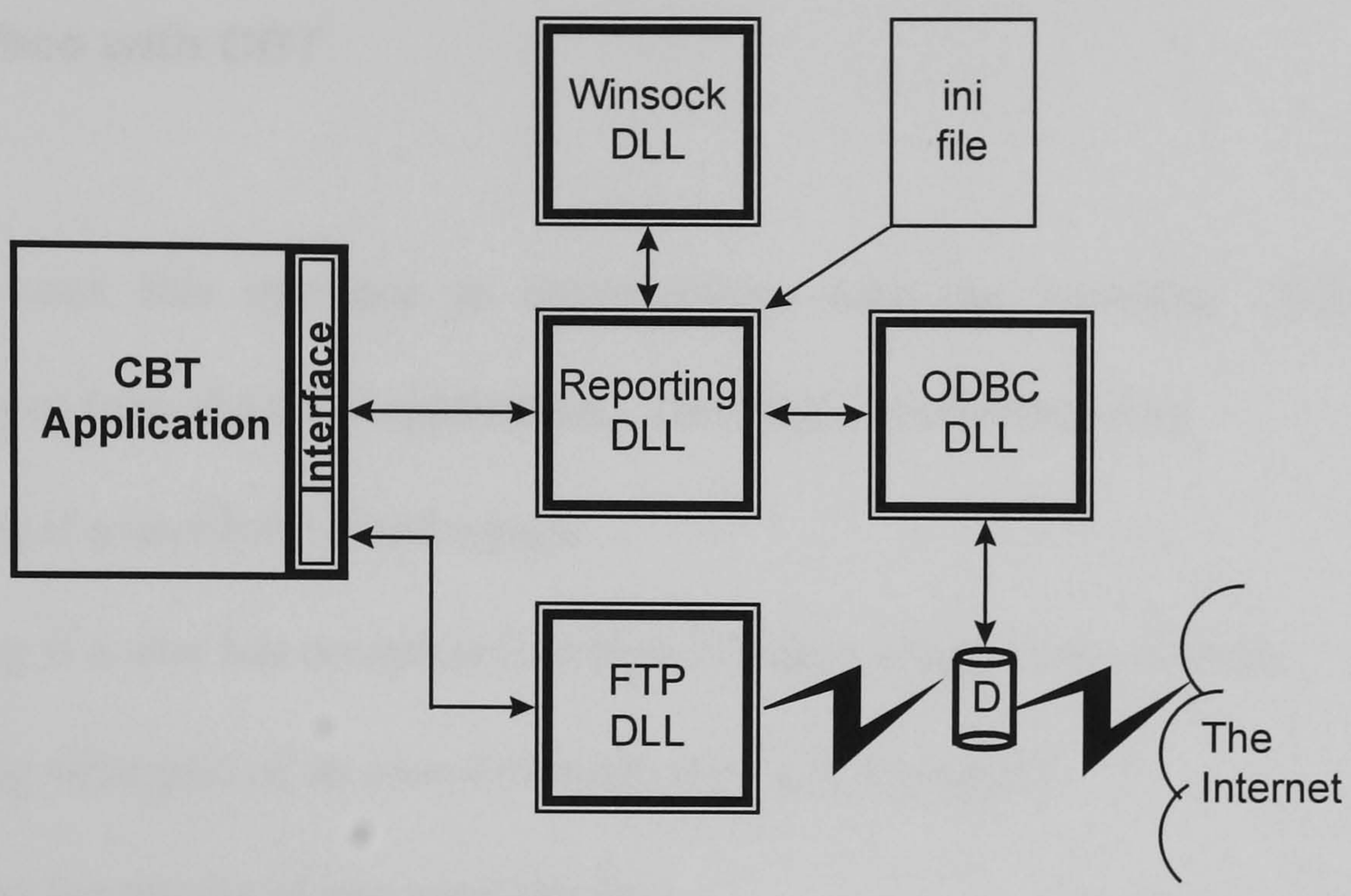


Figure 24: Reporting system overview

The author builds reporting functionality into the CBT application through the use of the *interface*. The interface passes commands (determined by the author) from the CBT to the *reporting DLL*. On receipt of these commands, the reporting DLL then uses the ODBC DLL to create and manipulate a database. The structure (tables, fields, etc.…) of this local database is, in the main, dictated by these commands. The commands that the reporting DLL receive from the CBT (via the interface) specify which variables the CBT reports and stores.

The database is split into two parts; a part which contains information that is automatically reported at the beginning and end of a session (with its structure determined by the *ini file*), and the user-defined part (with its structure determined by the reporting requirements of

the author). At the end of the session the database is transmitted over the Internet, or LAN, using Winsock implemented through the *FTP DLL*.

4.4 Interface with CBT

The author uses this interface to communicate with the reporting DLL and report information out from the CBT application. The kind of information is:

- Reporting if a user has visited a page
- Reporting if a user has completed a range of pages, chapters or sections
- Reporting what part of an assessment the user has attempted
- Reporting the results of any assessments

This is the sort of information which would normally be stored as variables in the authoring language. Therefore, the reporting interface is effectively a way of sending a variable description and value to the reporting DLL so that it can be stored in the database. From the kind of reporting requirements listed above, it can be seen that ideally the reporting system should allow the author the choice of grouping together related data or variables that are to be reported from the CBT. For example, if the author wishes to report the results of an assessment that occurs on the last page of each section it would be useful if there was an in-built method in the reporting system that allowed the Author to tag all the results from each section together.

In addition to allowing the reporting of variables, the interface must be capable of informing the reporting DLL when a session begins and ends. A session is defined as the

period of time in which the user invokes the CBT package, interacts with it, and closes it down again. Immediately upon a user entering the CBT there are details which can be stored about the session that will not change, such as:

- Company Name
- CBT Name
- CBT Version
- User Name
- User Number or ID
- Information Delivery Details
- Session Start Time / Date

Having established the main requirements of the interface, it is now possible to discuss how this could be implemented. When the student enters the CBT application they are identified and the CBT informs the reporting DLL that a session has started. When the student navigates through and interacts with the CBT there will be data which will be reported, as dictated by the Author, at set points during the session. At the end of a session the CBT informs the reporting DLL that the session is complete. Thus the interface must provide the following commands:

1. Open (a session)
2. Report (a specific variable)
3. Close (a session)

Command 2 would be used several times during a session. As previously stated, it would be highly desirable if the system could allow the grouping together of specific variables that have something in common. The system therefore allows the author to group together related data in the following way:

Page 1

Variable Name	Value
Xcords	2
Ycords	59
Visited	Yes
Completed	No

Question Bank A

Variable Name	Value
Question 1	"A"
Question 1	"D"
Question 1	"C"

At the end of a session more details are stored, such as:

- Session End Time / Date
- User Feedback Message

Therefore, the reporting system was split into four functions:

- Begin Reporting - Initialise Reporting, Store Start-up Details
- Group specify - Specify Group which next reported variable will belong to
- Variable Reporter - Report Variable Name and Value
- End Reporting - Store Finish Details and tidy up

At the start of the session, a database is opened and some initial information is stored in an attendance or 'details' table. During the session, variables are reported into specific 'groups' or tables. At the end of the session, some final information is added to the database before it is closed. The simplest interface to facilitate this functionality is to use commands combined with operands. The 'open' command has an operand which specifies the name of the database in which to store the information. The 'group' command has an operand specifying the table in which information should be reported. The 'report' command has an operand which is the name of the variable to report. The function that the interface uses to communicate with the reporting DLL therefore looks like this:

status = **Monitor** (Command, Operand, Value)

Where 'Monitor' is the name of the function used to communicate with the reporting DLL, 'Command' is the functionality that the reporting DLL is to perform (e.g. open, close, report, group etc.), 'Operand' is the data which is required to perform the command (or name of variable to be reported), and 'Value' is the value of a variable to be reported. The function returns a 'status' indicating whether or not the instruction was successful. Additional functionality was added to the interface to allow the creation of more than one database per session. If the command 'Open' is used with an operand containing a specific identifier for an individual database, then the system is be able to manipulate more than one database.

Every database requires its own 'Open' and 'Close'. However, the author also requires a way of indicating which database is active and the target of any 'Report' commands. This was achieved by the introduction of an additional command 'Point'. This command upon

receipt of a specific database identifier marks that database as active. Another useful facility is that the CBT has also been given the ability to report information directly into the automatically created table ('Details' table). This means that all the standard details about the session, such as 'CBT ID', and any additional information about the student (over and above their name and ID) can be kept together. This was incorporated into the interface by including a 'Detail' command. The 'Detail' command works exactly like the 'Report' command, except the variable name being reported has to match a user-defined field in the standard 'Detail' table. A summary of the interface commands is shown in Table 11.

Command	Operand	Value
Open	Database identifier	Physical filename of DB
Close	Database identifier	- not used -
Switch	Database identifier	- not used -
Group	Table name	- not used -
Report	Variable name	Variable value
Detail	Field / Variable name	Variable value

Table 11: Reporting system interface commands

The next major issue that was of concern was making the interface between the system and the author as simple as possible. The author needs to be able to incorporate these functions into the CBT application as quickly and painlessly as possible. It was important to provide a good level of abstraction from the DLL interface, so that the author can continue on with what they do best – Authoring, without worrying about how the monitoring system worked. Another issue was how to make the interface compatible with the wide variety of

authoring tools that are used to develop CBT. The following section describes how this was achieved in a variety of languages/tools commonly used in industry to develop CBT.

4.4.1 Implementing the interface with Authorware

With Authorware an author constructs a multimedia application by assembling icons on a flowline, very similar to a flow chart. The position of these icons dictate the flow of control (Figure 25).

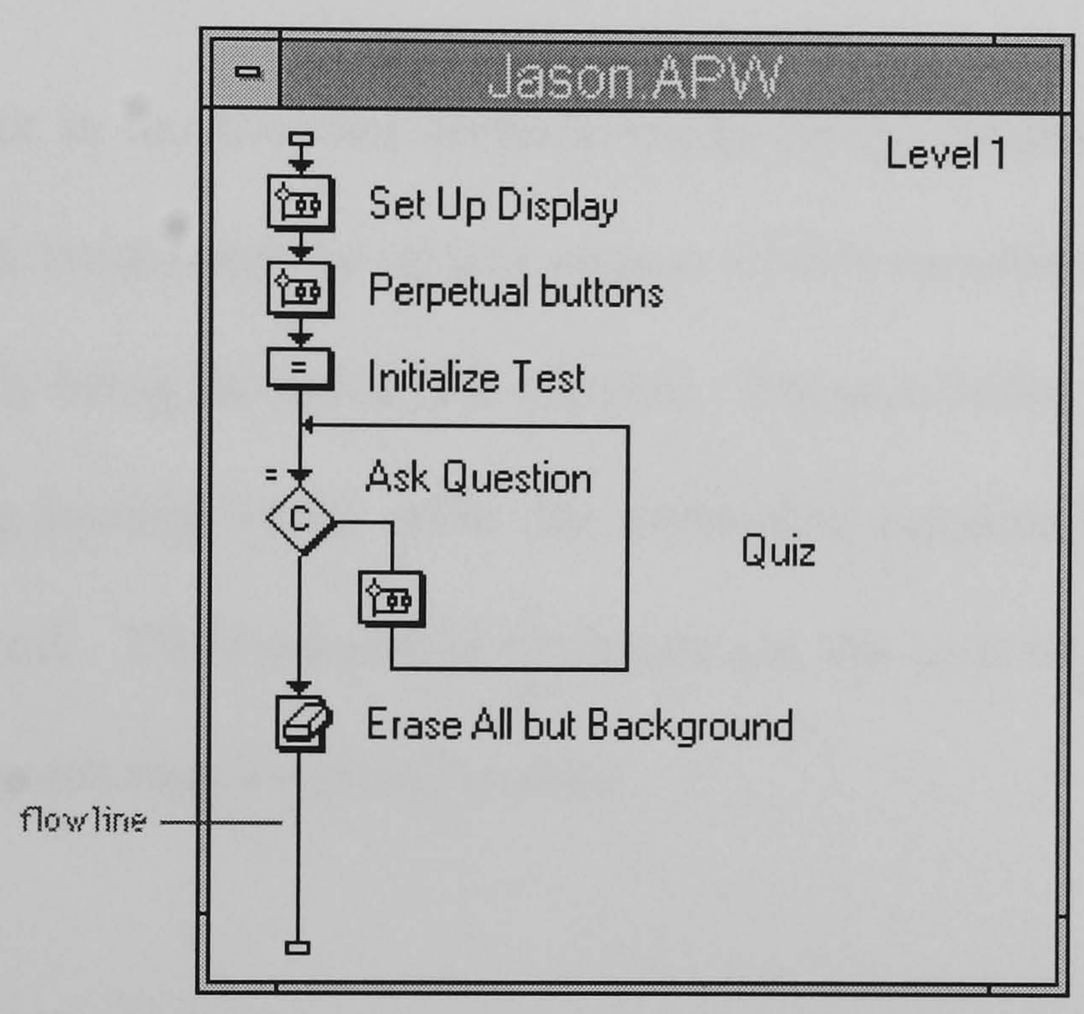


Figure 25: Authorware: The flow-line

Icons contain the objects that the author includes in the application. Different types of icons contain different types of objects, such as graphics, text, sound, digital video movies, or a set of instructions. The visual parts of the application appear in the presentation window. This is where objects can be arranged on the screen and progress viewed as the piece is constructed.

Since Authorware is centred around the use of icons, the ideal solution is to create a new icon, a 'monitoring icon', which can be dropped onto a flow-line as easily as any of the standard icons. As Authorware does not allow the creation of new icons, with user-defined functionality a compromise was made by using a standard Group icon (an icon that normally contains a group of more icons) with a title consisting of the instruction to be sent to the 'Monitor' function contained in the reporting DLL [40]. For example:

The 'Value' parameter in the function Monitor would be the evaluation of the Operand. The 'Value' parameter would only be of importance when a variable is being reported due to the Operand actually being the name of a variable. The monitoring icon itself contains a simple piece of string parsing which splits the instruction contained in the title into the Command and Operand. The Operand is evaluated (in the case of the command being 'Report') before the monitoring function is called.

This method proved to be extremely easy to implement, as authors could access the whole suite of monitoring functionality through the use of a single icon. The actual title of the icon itself dictates its functionality. The authoring process was greatly simplified, as the author did not have to select from a range of different icons. This system also completely detached the author from the inner workings of the icon's function and definitively defined the action of the icon at a glance. There is some redundancy in using this method, due to the repetition of the string parsing routine that is contained in every icon (repeated with every instruction), but compared with the vast size of a multimedia application it was

decided that the wasted space would be negligible. An example of how this interface could be used to report data from Authorware is shown in Figure 26.

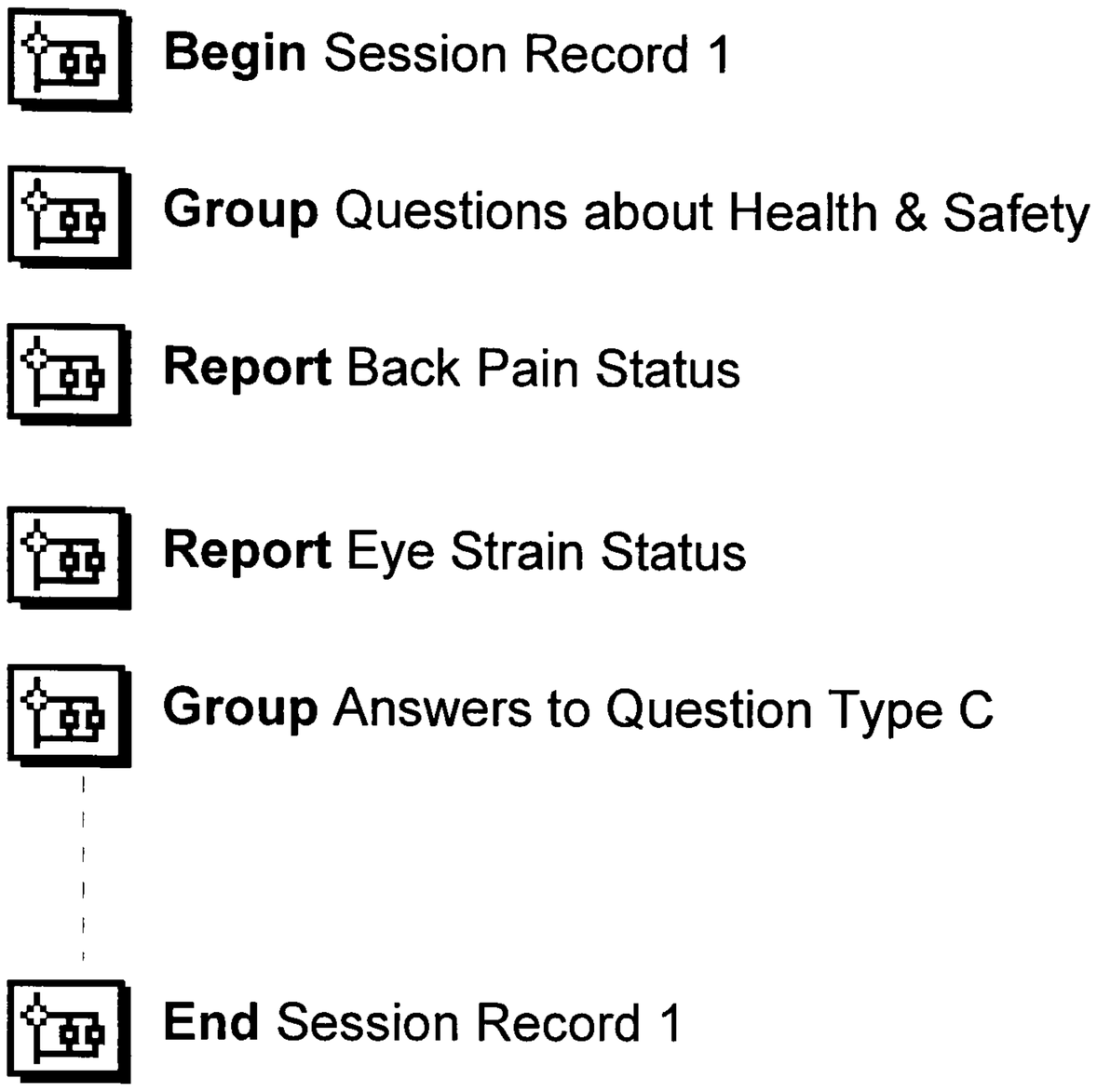


Figure 26: Using the generic monitoring icon

This methodology was found to be most effective, as the authors were already familiar with using high level descriptive titles in order to decorate their group icons. In practise, the generic monitoring icon was given its own specific colour, red, to signify that it was special and that authors should not open or alter the contents therein.

4.4.2 Implementing the interface with Director

Macromedia Director is an authoring language used to create and deliver interactive multimedia for both the Internet/intranet and CD-ROMs. In Director, the author creates an interactive "Movie" using an intuitive visual development metaphor in order to import, animate, and control media. Director provides quite sophisticated interactivity through easy-to-use drag-and-drop behaviours or a powerful object-oriented scripting language. In Director members of the 'cast' (media objects) are dragged onto the "stage" (screen shown to user). The order in which these objects appear is dictated by a score.

The score is effectively a time line representing the frames at which the object is visible to the user. When the "movie" is run, a "play-back head" runs along the score allowing the user to see each of the individual media objects. Cast members are not only media elements but can also be behaviours or scripting language components (called "Lingo") that allow complex functionality to be included.

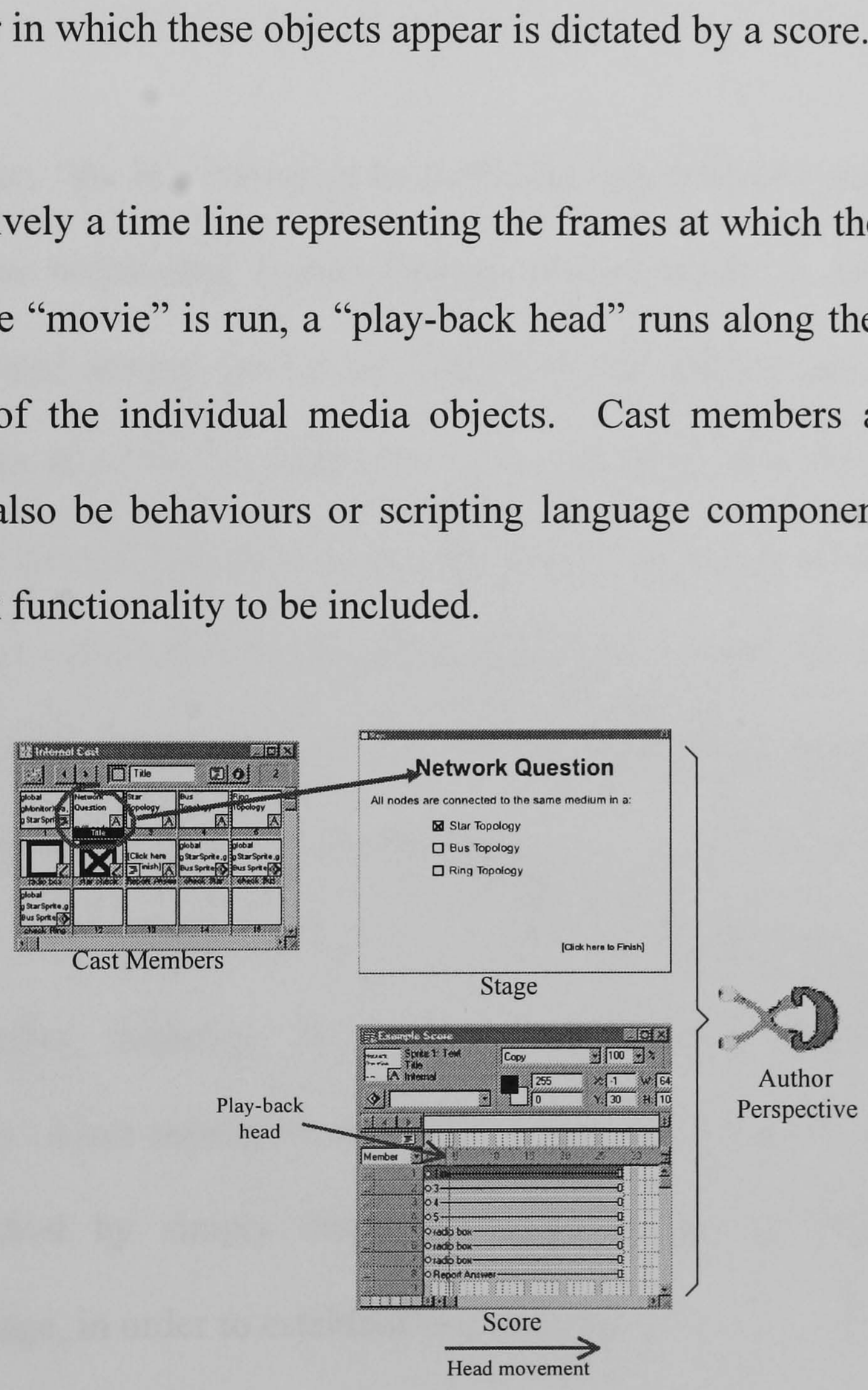


Figure 27: Overview of authoring components in Director

Figure 27 shows the major authoring components used to build applications in Director. Director is more technically difficult to master than Authorware, and as a consequence, it is far more likely that an author using Director would have more programming expertise than an author using Authorware. However, even if this is the case, the designers of Director have provided a way for authors to include interactivity through either a scripting language and/or pre-made drag and drop behaviours. It is therefore important to provide an interface to the monitoring functionality as easy to use as these drag and drop behaviours, as it cannot be assumed that the depth of the authors programming ability goes further than this.

As with Authorware, the ideal solution to providing a monitoring interface is to give the author access to the monitoring system through objects he/she is already familiar with. Authorware is centred around the use of "Icons" so the ideal solution was to provide a generic "Monitoring Icon" for the authors to gain access to the monitoring functionality. Director is centred around the use of "Cast Members" so the ideal solution is to provide authors with a behavioural cast member that provides the monitoring functionality built in. This generic multipurpose cast member will allow the authors to access the whole suite of monitoring and reporting functionality in the easiest possible way. Director allows behaviours of one cast member to be associated with events in a sprite derived from another cast member, therefore, the author can add the reporting cast member's functionality to any object shown on the stage, or even to a specific frame in the movie. This is accomplished by simply dragging the reporting cast member onto objects, contained on the stage, in order to establish associations.

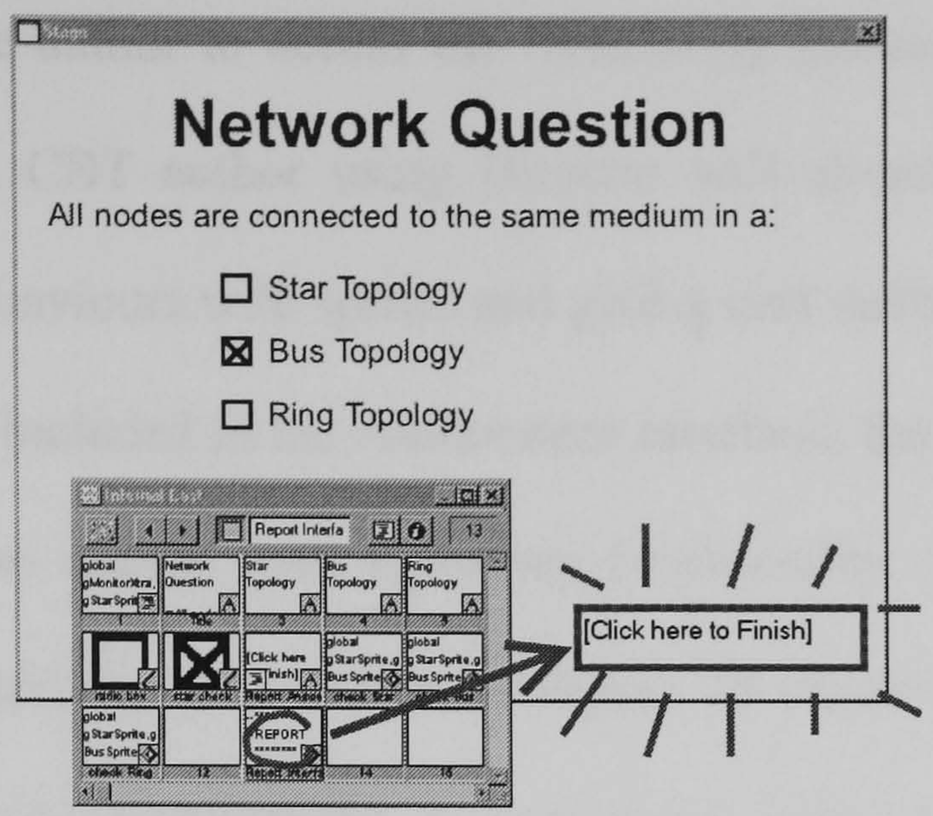


Figure 28: Using the generic reporting cast member

Figure 28 shows an example where reporting functionality is added to a sprite through the use of the generic reporting cast member. The sprite ("Click here to finish") has adopted the behaviour of the reporting cast member and will report information out of Director through the TIS. For the reporting cast member to be generic and fully reusable, specific functionality must be derived from the sprite itself. For example, the instruction passed to the external monitoring system must be variable depending in the properties of each individual sprite. This has been achieved by using the name of the cast member that the sprite is spawned from as the monitoring command line instruction (Figure 29). This enables the generic reporting cast member to be re-used in many different reporting scenarios, and as many times, as the author wishes.

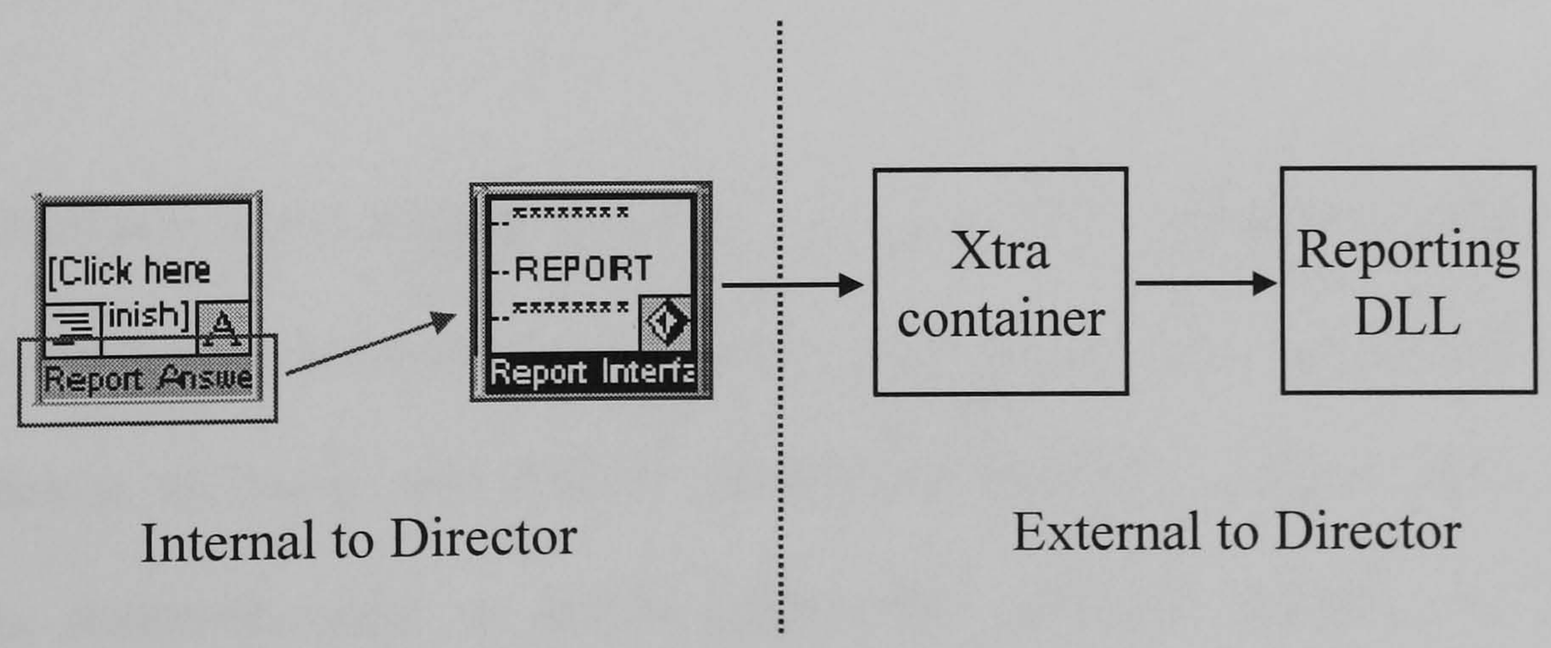


Figure 29: Instruction execution via Reporting cast member

This interface allows the author to access the monitoring functionality in a familiar and uncomplicated way. A CBT author using Director will already be familiar with cast members, associating behaviours with sprites and giving cast members names. As with the generic monitoring icon included in the Authorware interface, the generic cast member for director can provide the author with reporting functionality without them having to understand the inner workings of the system. Figure 29 shows how the title of the cast member that the reporting functionality is associated with, is passed through to the reporting DLL.

The normal way of extending Director's functionality is through the use of Xtras. Xtras are instantiable code libraries that are dynamically linked at runtime – much like a DLL, but in a proprietary format. Director supports the use of DLLs through the use of an "Xtra" wrapper. This Xtra, named DLLGlue, wraps the calls to the reporting DLL in order to make it compatible with Director. This is of little consequence to the author who uses Director to develop CBT because he/she is shielded from the calls by virtue of the generic cast member interface.

4.4.3 Implementing the interface with Asymetrix Toolbook

Toolbook Assistant and Toolbook Instructor are both part of a popular family of learning products developed by Asymetrix. Assistant is an authoring tool that allows authors with little experience to create and deliver educational training software extremely easily. Assistant is extremely easy to get to grips with, and few authors, no matter how

inexperienced they are, would have much difficulty mastering it. Assistant is probably the easiest to use authoring tool available today, but is somewhat limited in its functionality.

Instructor is a similar authoring tool, but is aimed at the professional courseware developer. Instructor combines the functionality of Assistant but with the ability to write programs in OpenScript language. When an author uses Instructor or Assistant to create CBT, he/she uses the built-in-tools for tasks such as drawing objects on pages, creating hyperlinks, associating behaviours. However, Instructor allows the author to take advantage of further capability through the use of OpenScript programming language.

OpenScript allows the author to control the behaviour of the application more directly and precisely by including commands to accomplish a wide variety of tasks. It is through the use of OpenScript that Instructor can link to functions in DLLs such as those provided by the monitoring functionality. Although an author who develops in Assistant is not allowed access to OpenScript programming, he/she can take advantage of pre-built scripts, created previously in Instructor, to accomplish tasks that Assistant cannot normally achieve. This has the fortunate consequence, that an interface developed using OpenScript, in Toolbook, in order to facilitate communication with the monitoring system, can also be used in Assistant.

Since Instructor and its companion product Assistant share a common file format and authoring methodology, and that Instructors advanced features can be used to create objects and solutions that work in the Assistant environment, the remainder of this section shall only deal with Instructor.

An author creates an application in Instructor, in much the same way as Macromedia Director. He/she drags multimedia content, or objects, from a catalogue onto the page viewed by the user. A complete application, or 'book', is made from a collection of pages. Each of these pages contain objects that have both properties and scripts. An object property is a variable associated with a specific object that indicates the objects status, e.g. the object name, the object position etc.. An object script is a segment of OpenScript code that has been associated with that object in order to respond to events that the object receives. An object may also have other multipurpose (shared) scripts associated with it. It is, therefore, possible to employ the monitoring interface as a shared script so that it is available to any object the author wishes to associate it with.

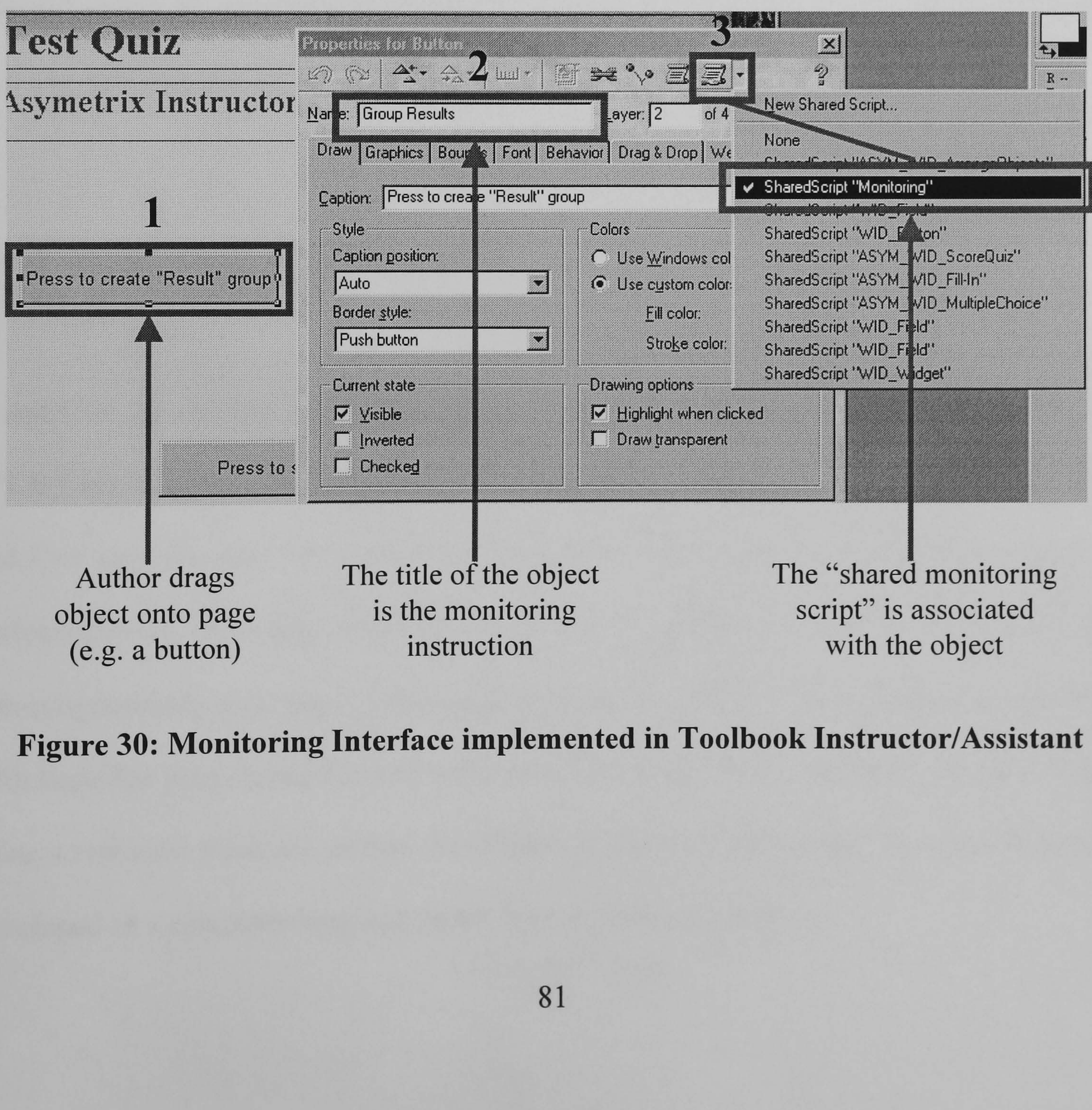


Figure 30: Monitoring Interface implemented in Toolbook Instructor/Assistant

Figure 30 demonstrates how an author can access the monitoring functionality by simply setting the title of the object to the instruction that they wish to send to the monitoring system. For this to work, the object must be able to parse the title, extract the value of the instructions operand and send this information to a function contained in the monitoring DLL.

This functionality has been pre-written as a shared OpenScript component that can be linked to any object. If the object title contains a valid instruction and the shared monitoring script is associated with the object, events such as a user click will cause that instruction to be executed by the monitoring system. This method has the advantage that the inner workings of the monitoring system remain concealed from the author, the author can still access all monitoring functionality and the author need only use the authoring tool in a conventional and easy manner.

4.4.4 Implementing the interface with C/C++

C and C++ are still the most used applications languages of today. Although the majority of CBT applications are developed with authoring tools, programming languages such as C and C++ are still used to develop CBT, especially where speed, efficiency and graphics performance are extremely important. Authoring tools fulfil an important role by allowing non-programmers to produce multimedia applications quickly, but inevitably compromise with large file sizes, limited functionality and slow speeds. For example, training a student using a real-time situation, perhaps in a flight simulator, would require the use of a product developed in a computer language rather than an authoring tool.

It is therefore important to ensure the monitoring interface can be used from the languages C and C++. The reporting system has been developed as a windows Dynamic Link Library (DLL). DLLs are the basic building blocks of the Windows operating system, and as a consequence, all programming languages that can develop applications for this operating system can use DLLs (including C & C++). It is not difficult for a C/C++ application to access the monitoring functionality contained within the reporting DLL. The reporting DLL must be loaded into memory and a pointer created to the 'Monitor' function contained within it. This pointer can then be used to access the monitoring functionality in the following way.

Firstly, define the 'Monitor' function with its parameters early in the program this will enable a pointer of that type to be created later on.

```
typedef GLOBALHANDLE (CALLBACK *MONITOR)
(
  LPCSTR   szCommand,          // Command – open, close, group, report etc…
  LPCSTR   szOperand,          // Operand – database name, variable name etc…
  LPCSTR   szValue             // Value – the reported data
);
```

Then load in the Reporting Dynamic Link Library and create a pointer that points to the 'Monitor' function that contains the monitoring functionality.

```
// load monitoring dll
HINSTANCE hMonitor = LoadLibrary("reporting.dll");


// create pointer to to monitoring functionality
MONITOR pMonitor = (MONITOR)GetProcAddress(hMonitor,"Monitor");
```

The pointer to the function 'Monitor' (pMonitor) contained in the reporting DLL can now be used to access the suite of monitoring functionality in the standard function interface:

Status = **Monitor** (Command, Operand, Value)

The equivalent reporting functionality demonstrated in the Authorware example (Figure 26) can be implemented in C/C++ in the following manner:

```
// Open Database
(*pMonitor) ("Open", "Session Record 1",0);

// Report variables
(*pMonitor) ("Group", "Questions about Health & Safety",0);
(*pMonitor) ("Report", "Back Pain Status", StrBackPainStatus);
(*pMonitor) ("Report", "Eye Strain Status",StrEyeStrainStatus);
(*pMonitor) ("Group", "Answers to Question Type C",0);
|
| etc...
|

// Close Database
(*pMonitor) ("Close", "Session Record 1",0);
```

Finally, the Reporting Dynamic Link Library must be removed from memory when all reporting is complete, and all database connections are closed.

```
// unload DLL
FreeLibrary(hMonitor);
```

This method of accessing the monitoring functionality may seem unwieldy, but it is important to remember that a programmer producing CBT applications using C/C++ will be extremely familiar with the programming language and will have little problem using the monitoring system in the above manner. Some may ask why a C programmer, with all the language's power at their disposal, would wish to use the reporting DLL in the first place. There are several advantages to using the generic monitoring system to report data. Firstly, one of the primary uses of Windows DLLs is to allow proved, established and bug-free pre-compiled code to be re-used by various applications. This means that the C developer will find it faster and more efficient to use the reporting DLL already developed than to proceed through the design and test cycle themselves.

In addition, the use of the reporting DLL will allow their CBT application to report into the TIS without compatibility problems and allow the reported information to be stored and viewed alongside data received from other CBT applications, even if they have been designed in alternative programming languages or authoring tools. The monitoring system provides a single interface to all courseware irrespective of the development method.

4.5 Reporting DLL

In its simplest form the reporting DLL is a library that extends the capabilities of the CBT to include reporting of variables into a structured database. The role of the reporting DLL is to react according to commands received from the CBT, such as 'Open', 'Close', 'Group', 'Report', etc. . These commands are received from an interface embedded within the CBT application (such as the one demonstrated for Authorware in 4.4.1). The

reporting DLL facilitates the receipt of these commands by exporting the function 'Monitor'. This function receives an instruction in the form of Command, Operand, and ValueOfOperand and dynamically generates SQL commands that are sent to the ODBC DLL to manipulate the database.

Figure 31 shows an overview of the system, and demonstrates the format of the database the system creates. All databases the system produces have a table called 'Details'. The 'Details' table contains information that is relevant to the whole session (start time, finish time, username etc.). The remaining tables in the database are produced as a consequence of a 'Group' command accepted from the CBT application. These user-defined tables are where the reported variables are contained.

The Details table also has a user-defined part, which the CBT application can use to store any data that is relevant to the whole session. The reporting DLL was written in the language C. This was selected as the most appropriate language to develop the DLL, due to the external components (ODBC & Winsock) offering a C API, the Microsoft Windows operating system, and the majority of authoring tools, that the system must interface to, being also based upon C.

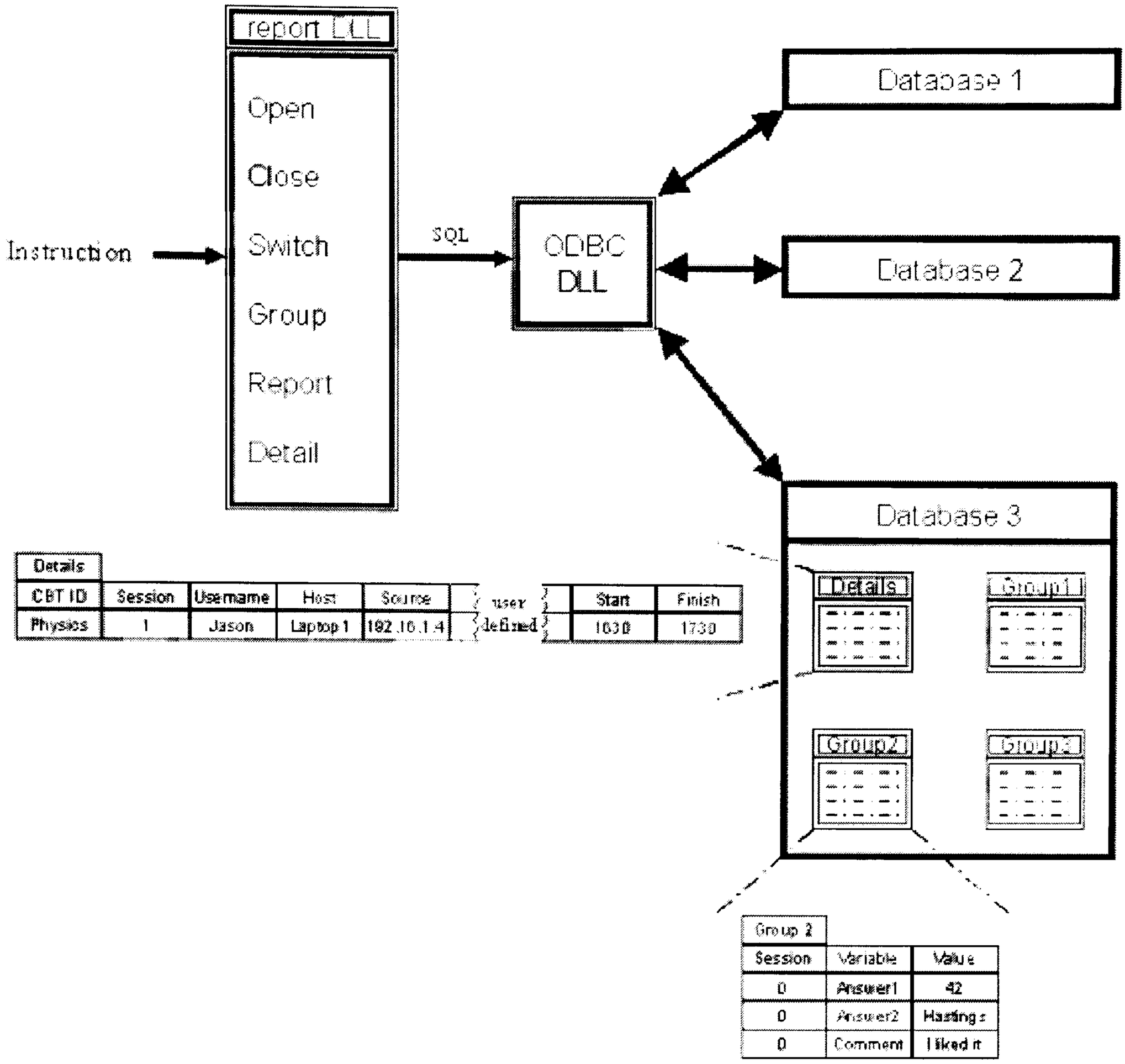


Figure 31: Overview of reporting DLL

4.6 Reporting Configuration File

The reporting DLL reads a standard windows 'ini' file when it is initialised. This enables the DLL to be configured externally each time it is implemented. A detailed explanation of this file is included in Appendix E. The file is used to specify things such as:

- CBT Unique Identifier
- Output Directory for databases
- Which ODBC.DLL to use
- Type of output database (Access, Dbase, MsText, etc.)
- Maximum Size of variable name/values
- Field name and sizes for user-defined part of Details table

4.7 ODBC DLL

Two ODBC DLLs are in fact used to facilitate reporting.

odbccp32.dll

This DLL is used to configure the ODBC environment. It can register connections to physical databases, and also create the actual databases themselves. It is necessary to use this DLL because the reporting system cannot assume that the ODBC environment is already set up. The type of database that the reporting system uses is selectable by changing the ini-file. As a consequence, the system needs to be able to configure its own data sources prior to reporting. This is achieved by using the "SQLConfigDataSource" function that is contained within odbccp32 library.

odbc32.dll

The odbc32 DLL is used to connect to, and manipulate existing data sources. There are many functions within this DLL that have to be used to set up and close down the

connection to the data sources. However, once a connection with a data source is opened, the "SQLExecDirect" function can be used to execute SQL statements upon the data source. SQL statements containing commands such as "SELECT", "CREATE", and "UPDATE" are built dynamically, in the reporting DLL, and submitted to SQLExecDirect for execution on the data source.

4.8 Winsock DLL

There is a requirement for the reporting system to identify where the training has taken place. This is achieved by obtaining the host IP (Internet address) number, and the computer hostname through using functions in the Winsock library. The functions used to get this information are "GetHostName" and "GetHostByName".

4.9 FTP DLL

The local database produced by the reporting DLL (through the use of ODBC DLL's) has to be transmitted over the Internet to the central administrator's site. It was decided that the flexibility offered by File Transfer Protocol (FTP), a protocol that makes it possible for a user to transfer files from one location to another over the Internet, would be the most robust method. The solution was implemented by using Macromedia's FTP user-defined code library. The connection to the administrator's site was opened by using the functions "FtpOpen" and "FTPConnect". The database file was then transmitted using the

function "FtpStore". The specific details of the connection, such as the servers IP address, were obtained from an external configuration file.

4.10 Retrieving Student Identity via Network

The different ways of obtaining the student's identity were discussed in section 3.2.2. where it was concluded that the most convenient, way to retrieve this information was from a previous login. This can be achieved by building in 'sniffing' methodology that allows the reporting DLL to identify the network type, Novell, Microsoft, etc, that the student is already logged into, prior to attempting to communicate with it. To test the validity of this idea a solution was implemented to sniff out, and communicate with, Novell networks.

The reporting DLL was given an additional function:

String **GetUserID** (void)

The GetUserID function instigates a search for information about the current Network Operating System (NOS), and tries to establish if it is Novell. If it finds evidence of a Novell network connection, it communicates with the Novell NOS API via the DLLs clxwin32.dll and calwin32.dll. If this is successful, the function retrieves the username and domain name of the person currently logged into the system as a string. If the function fails, perhaps because the Novell NOS is not present, it returns an empty string. This provides the CBT application with the opportunity to resort to a back-up login script as an alternative way of obtaining the same information. Although this solution was only

implemented and tested with the Novell NOS, there is no reason why it cannot be extended to include other network operating systems, in a similar way.

5 Implementation of collation system

This chapter describes the steps that were taken in the design of the collation part of the generic monitoring system. The chapter describes how the system is used by the training manager to produce reports, and then describes each of the individual components of the Collator application.

5.1 The role of the Training Manager

The role of the training manager is to create, customise and maintain a successful and efficient learning environment. To do this, the training manager must ascertain who has trained in what, how well students are performing and how well courses are being received by the students on the whole. It is therefore the task of the training manager to analyse the results of the training activity through the use of the monitoring system. The training manager is also likely to liaison with the authors of the application and specify what it is important to record about the students activities or test results. The training manager would also use the collator application to merge the individual reported databases (one per session of training) into a single master database and execute standard reports to facilitate analysis.

The collator application is designed to be maintenance free and extremely simple to use. The program should be run at regular time intervals, or most certainly before a report is output from the master database. If the collator is not used regularly, a large amount of

disk space would be used up by the reported data building up in the receiving directory, and any reports output from the master database would be out of date.

5.2 Overview of the Collator components

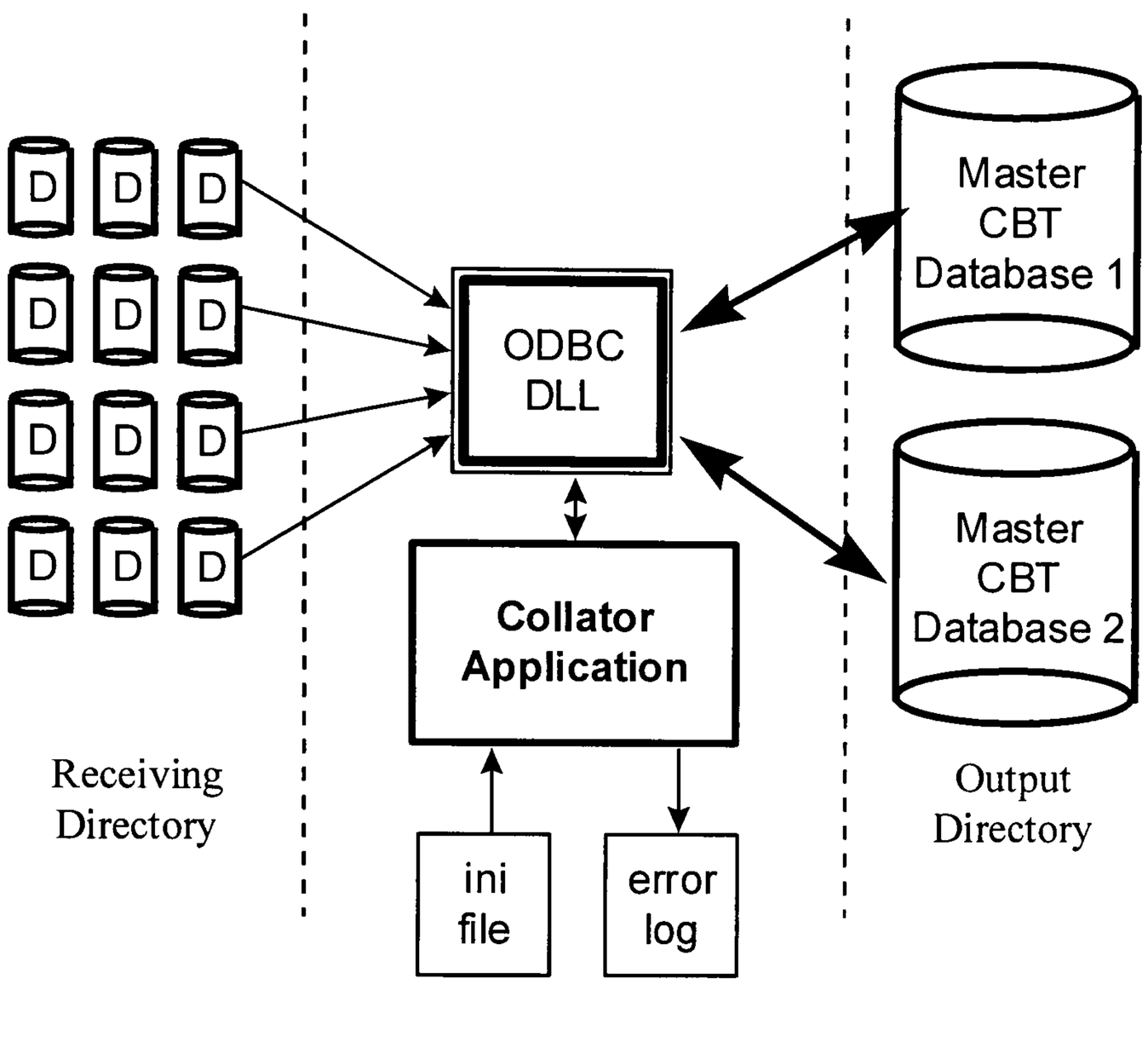


Figure 32: Overview of Collator application components

The collator is a dialog box application developed in C that merges the individual session databases into a master database. Each of the individual session databases (incoming databases) could have been derived from a different CBT application, so the collator produces a master database for each of these applications. The collator achieves this by differentiating between each of the 'CBT unique ID' fields that are present in the incoming databases. As a consequence, every CBT application using the monitoring system will have its own master database, and each of these databases would contain their own queries and methodologies for analysing the data.

The collator uses the ODBC DLL's discussed in 4.7 to facilitate the reading/writing and creation of any database types that have an ODBC driver. Each of the incoming databases could be in a different format (i.e. Access, Dbase, MsText, etc.). As a consequence, the first task the collator performs is to determine the type of database it is dealing with and initiate an ODBC connection to it prior to examining its contents. Each individual incoming database is examined in turn and deleted after successful integration into the master database.

The Collator application does not have to have prior knowledge of the CBT application before it receives an incoming database from it. If a new incoming database is received from a CBT application that it doesn't recognise, the Collator simply creates a new database for it based upon the structure of that first incoming database. Consecutive databases from that application are then merged into that new database.

Should a database be received that it can identify but the structure is different from the existing master database (through an error – perhaps a different version) then the master database is not updated and a notification is placed in an error log for examination by the training manager.

5.3 Collation Configuration File

The collator application reads a standard windows 'ini' file when it is started up. This enables the collator to be configured externally. A detailed explanation of this file is included in Appendix E. The file is used to specify things such as:

- Input directory (for the incoming session databases)
- Output directory (where to edit/create the master databases)
- Location of error log text file
- Output database format (i.e. Access, Dbase etc.)

5.4 Choice of database formats

The collator is capable of detecting and reading any database format supported by ODBC, assuming all the correct ODBC DMS drivers have been installed. In addition, the collator can also be set to create or append a master database of any type. This flexibility ensures that the system will operate seamlessly with a customers existing DMS. In addition, it gives the customer, or the training manager, the option to use the database facilities of their choice in order to perform queries and interpret the data.

However, there are significant differences between proprietary database formats, not from a functional perspective as this is enclosed by ODBC, but rather in the way the data is physically stored, retrieved and maintained in the files. These differences have a dramatic affect both on file-sizes and on access speeds. In practise, the smallest file size and highest access speed is the most desirable scenario. This is because these factors are directly proportional to the bandwidth taken up by the reported data, and the time taken to collate the individual files into the master database. This issue is investigate fully later in the text.

5.5 *Visualisation of data by the training manager*

The overall objective of the training manager is to measure their organisation's training effectiveness and obtain information that helps them calculate return on investment. It is, therefore, important to have some visualisation method to help the training manager to analyse the master database, with ease, and to determine the progress of individuals or groups and to allow them to assess the effectiveness of the curriculum. The use of ODBC opens up the potential of using the powerful reporting and visualisation tools already provided by commercial database vendors.

Many of these databases offer very easy ways of fast tracking the procedure to create informative reports. For example, Microsoft Access database now provides wizards that assist a user, who can be quite unfamiliar with databases, in creating graphical or tabular reports. These sort of tools are extremely important, as training managers are just as unlikely as CBT authors to have the technical skill to program in SQL. In addition, many companies already have an established MIS, perhaps for personnel or maybe even for holding training information, and have set ways of displaying data in a particular format.

As previously described, an advantage of CBT over conventional instruction is that it provides efficient possibilities for monitoring and tracking of student and course performance, that would have previously been impossible. This factor, as new students and courses are introduced to the system, will lead to a large amount of data in the master databases. Data visualisation methods may need to be employed due to the amount of data held in the TIS and the strategic importance of being ably to analyse it quickly and efficiently.

6 Validation of the Monitoring System

The monitoring system was validated in order to establish its merits in different areas. Firstly, it was a design objective to ensure the tools provided by the monitoring system could be easily integrated, by non-programmers, into the majority of authoring tools in widespread use today. In addition the monitoring system had to be proved to be usable, effective and efficient by being transparent to the student and using the minimum of bandwidth over computer networks whilst working with students in a distributed environment. This is especially important, as any monitoring system should add value to the CBT experience, and not detract from that value by being slow, cumbersome and interfering.

The ability of the monitoring system to be able to cope with the wide variety of CBT monitoring requirements is also significant, in addition to the scalability of the system as a whole. Finally, the benefits to the training manager had to be measured to establish if the system could provide the training manager with the ability to track what students have seen and done, and to measure training effectiveness and return on investment. In order to establish the merits of the system, and address these considerations, the monitoring system was subjected to several trials both in industry and academia.

It was possible to provide some semi-quantitative results from the trainees on their perceptions of the TIS. On the other hand, the small number of training managers and authors resulted in more nebulous feedback which is summarised at various points.

6.1 *Trials at CBL Technology*

The monitoring system was initially validated during internal trials at the collaborating company, CBL Technology. These trials involved setting up the TIS on the company network and incorporating courseware into the system. The courseware selected for the trial was titled "Display Screen Equipment Health and Safety" (DSE). The DSE CBT course consisted of several modules of instruction with a final risk assessment exercise at the end. The objective of the risk assessment was to ascertain if the student had their workstation ergonomically arranged in order to prevent injury. It was particularly desirable for this CBT application to be monitored as the Health and Safety (Display Screen and Equipment) Regulations 1992 require companies to give guidance on basic ergonomic practice. It is therefore a legal obligation for companies to record that this training has taken place, and to assess their staff risk.

The requirement from the training manager's perspective was two fold, firstly to record proof that each student had visited and interacted with all the mandatory sections, and secondly to record any problems people had with their workstations so that the appropriate action could be taken to rectify the situation. Figure 33 shows some introductory screenshots of the Health and Safety application, and demonstrates that the students are clearly informed of the fact that they have a legal responsibility to complete the course.

The course itself is split into four main sections, three of which are used to deliver the training, and the fourth to quiz the student about their current situation in order to determine their risk factor. The TIS developed by this project offered the ideal

environment and opportunity to record student's exposure to the material and the results of the risk assessment for analysis by the training manager.

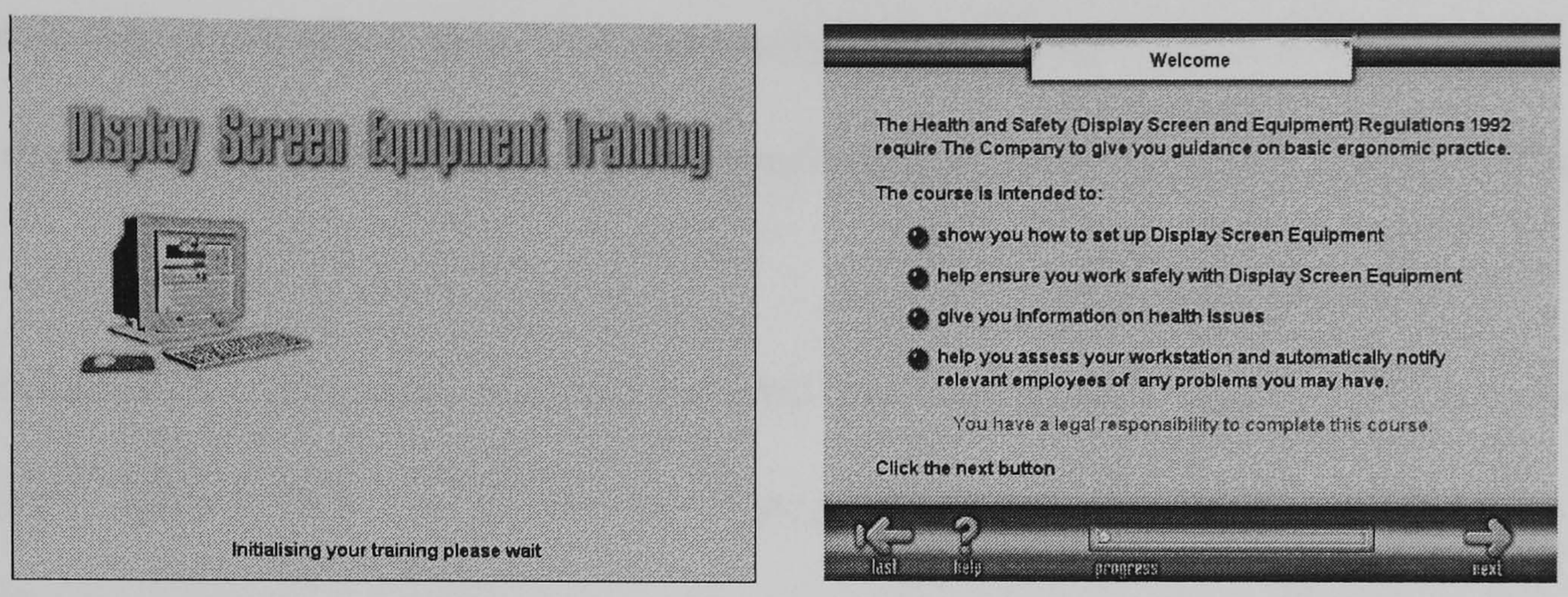


Figure 33: Screenshots Monitored Health and Safety Training [7]

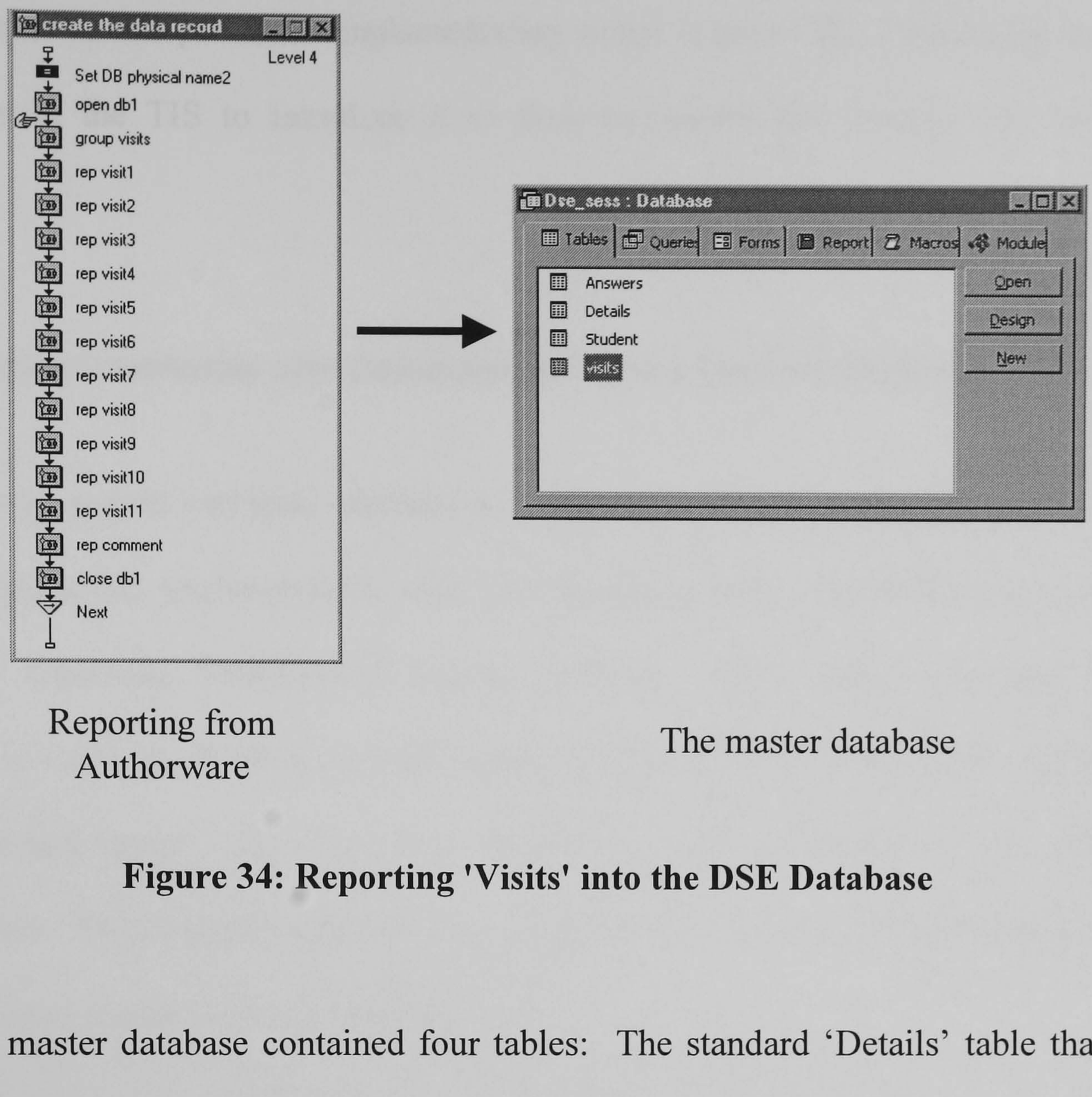
The Health and Safety CBT had previously been developed in Macromedia Authorware together with its own proprietary, crude reporting system. The main impetus for incorporating this software into the newly developed generic monitoring system was to enable the reporting system to work over the Internet and use the advanced features that the new system provided. In addition, CBL Technology Ltd., who intend to expand this product range, desired not to be restricted to a specific authoring tool whilst retaining TIS compatibility throughout the whole product range. There was therefore a distinct advantage to utilising the generic monitoring system which could be incorporated into the majority of authoring systems.

The non-programming authors who integrated the CBT application into the TIS found it an easy task. The data that was of interest had already been stored in variables within the authoring tool, so it was possible to report these into the TIS through the use of the Generic Monitoring Icon (described in 4.4.1). The Health and Safety training was tested on the company intranet and made available to around 40 employees. The TIS system was found

to successfully identify which students had completed the mandatory material, and identify those students with problems in the risk assessment. The fact that the TIS automatically tracked the IP addresses of the student's computers was found to be especially useful in this particular application, as it was extremely easy to locate the students location when action needed to be taken in the event of a workstation receiving a poor risk assessment.

The collaborating company commented that the real power of the TIS was that the monitoring system did not dictate how they should develop their applications. For example, the authors have the option to produce an intranet or LAN solution using any of the major authoring tools available. In addition, authors found that the system allowed them to define the structure of how interactions are monitored whilst offering them advanced features that had not previously been available. This was extremely important, and a great time saver, when integrating an existing CBT application previously developed for a proprietary reporting system.

The monitoring system was seen as a truly flexible way to incorporate monitoring functionality into applications, offering advanced monitoring features whilst shielding the complexity from non-programming CBT authors. Figure 34 shows how the generic monitoring icon was implemented to report user comments and mark the pages they had visited.



Reporting from Authorware

The master database

Figure 34: Reporting 'Visits' into the DSE Database

The DSE master database contained four tables: The standard 'Details' table that holds location and identification information, a 'Answers' table to hold the responses to the risk assessment, the 'visits' table, and finally a table called 'student'. The 'student' table was used to hold extra information related to individual students, such as "floor number", "location code". This information was stored in a separate table because this particular version of the monitoring system did not allow authors to create user-defined fields within the mandatory 'Details' table (see 4.4).

In later versions it was unnecessary to hold this type of personal data in a separate table. Further details on how data was reported out of the DSE CBT application, and how this data was collated and analysed in report format, is very similar to a further implementation at Cable & Wireless Communication PLC which is described in depth in section 6.2. The

greatest value of this particular implementation is that it gave CBL Technology sufficient confidence in the TIS to introduce it to their customers and establish its credibility externally.

6.2 *Trials at customer site Cable and Wireless Communications PLC*

The reporting system was then validated at Cable & Wireless Communication PLC where a CBT solution was implemented to train their telephone sales personnel in the use of their Integrated Customer Management System (ICMS). This system provided training managers at Cable & Wireless with the means to analyse which parts of the training their employees had viewed. The application (shown in Figure 35) consists of four individual test modules. The students undertook these modules after the initial ICMS training and all four of the test modules were monitored.

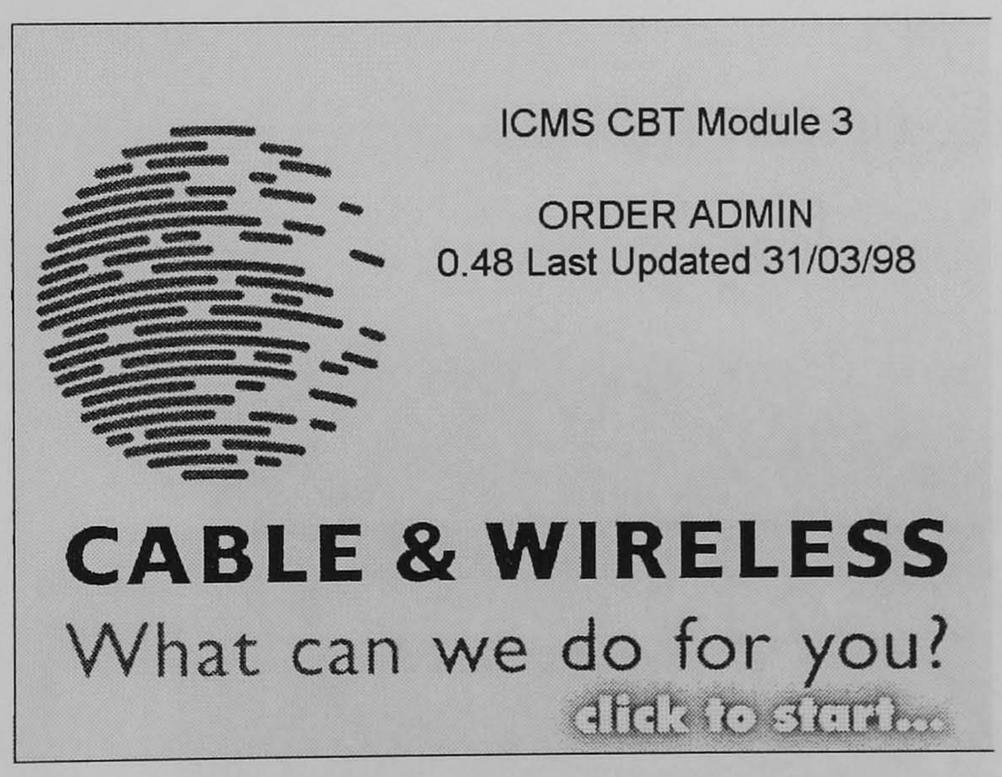


Figure 35: ICMS CBT application [8]

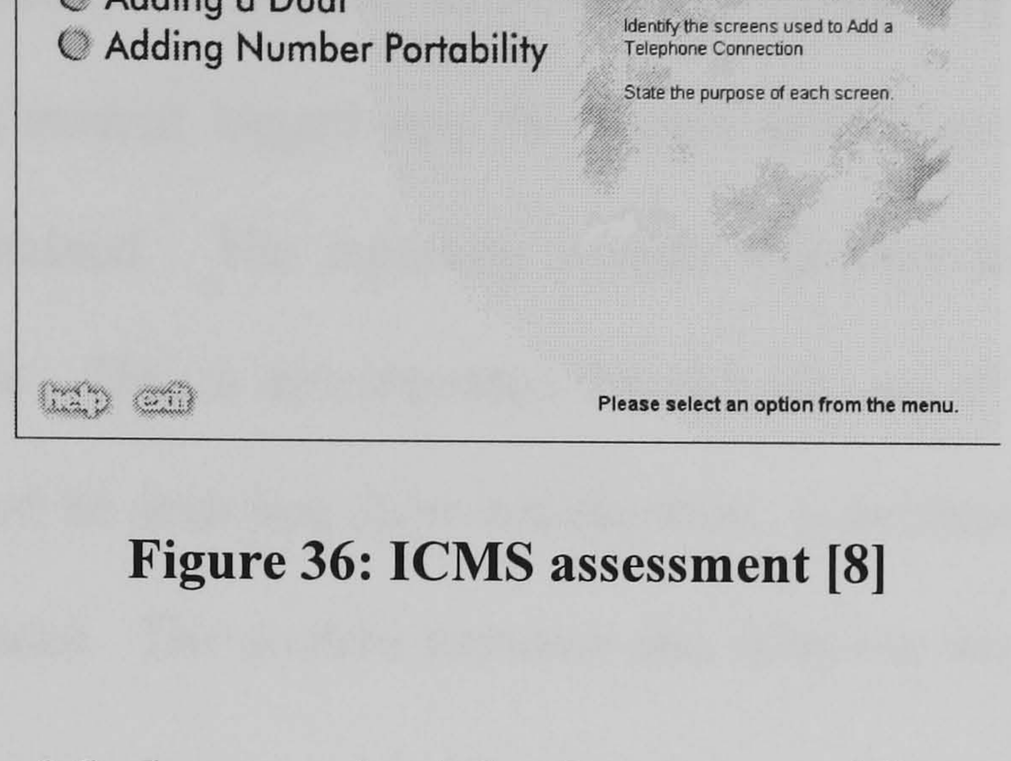


Figure 36: ICMS assessment [8]

The main menu (Figure 36) shows each of the four tests; adding a Telephone Connection (TelCo), adding a Cable Television Connection (CATV), adding both a Telephone and a Cable Television Connection (Dual), and adding Number Portability. The reporting system was applied to each of these four tests. Each test involved a simulation of the ICMS and every data item entered into the system was recorded in variables. These variables were then reported using the reporting DLL. Figure 37 shows the simulation screen for adding a telephone connection (TELCO). We shall now examine how the authors actually implemented the reporting system in a little more depth.

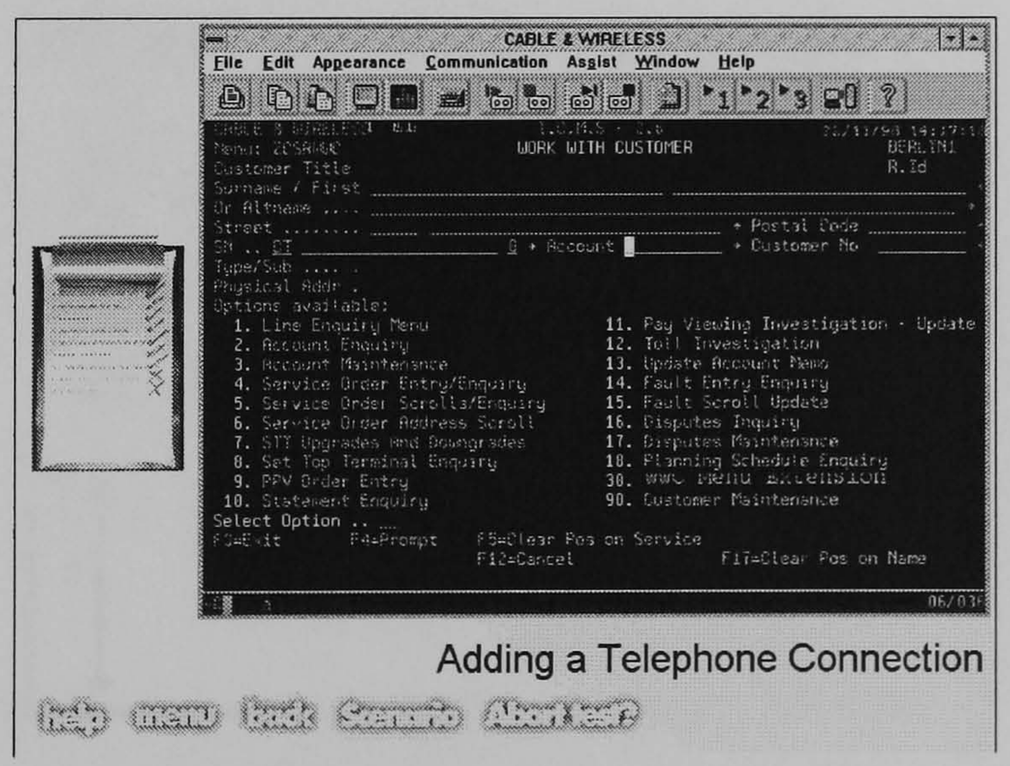


Figure 37: ICMS simulation [8]

The resulting activities of the students in the ICMS test suite produced a single database. The results for each of the four tests were grouped into separate tables. The application was written in Macromedia Authorware, so the generic-monitoring icon described in 4.4.1 was utilised. When a student logged into the system their User-Sales-Rep number and their surname was obtained. The reporting system was then initialised by opening a database for the session. This is demonstrated through the use of the generic monitoring icon in Figure 38. It can be seen that upon initialisation, a database is opened, and a new table "attend" was created. The student surname and sales-rep are reported into this new table.

The student then did one of the four tests. At the exit point for a particular test the performance/status variables for that test are reported into a specific table. Figure 39 shows how the variables containing the results for the TELCO test were reported using the 'Group' and 'Report' commands. The reporting system allows the command "Report" to be shortened to "Rep" to quicken implementation. At then end of the training session the database was closed using the "Close" command as illustrated in Figure 40.

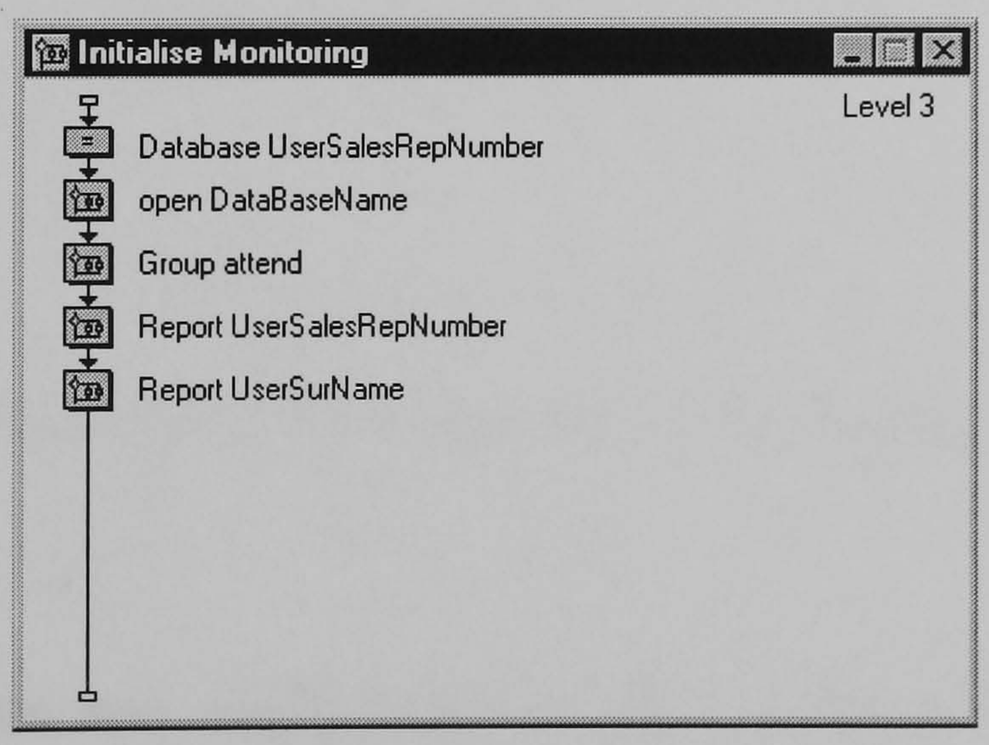


Figure 38: ICMS TELCO, initialising the reporting system

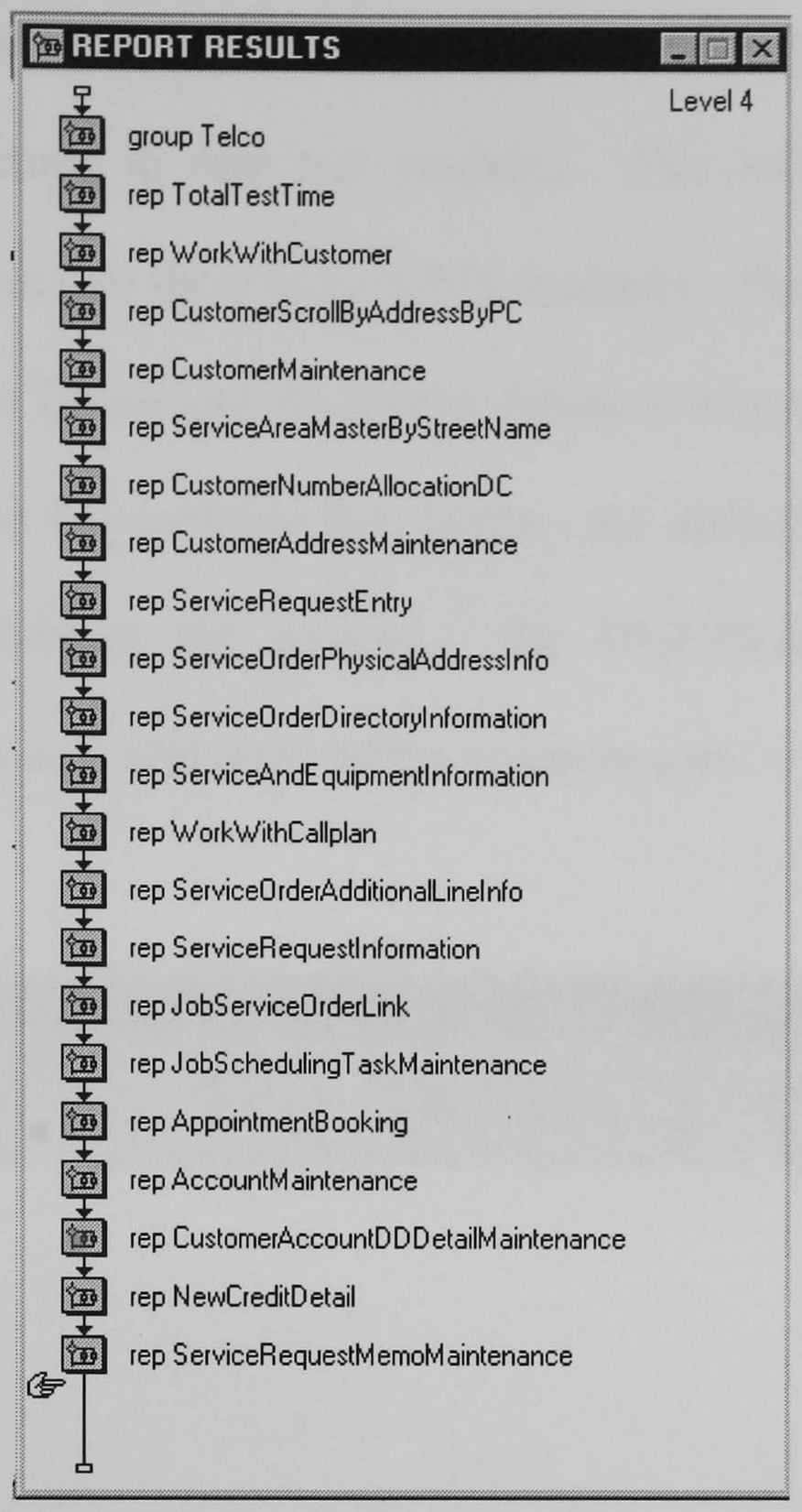


Figure 39: ICMS reporting results for TELCO test

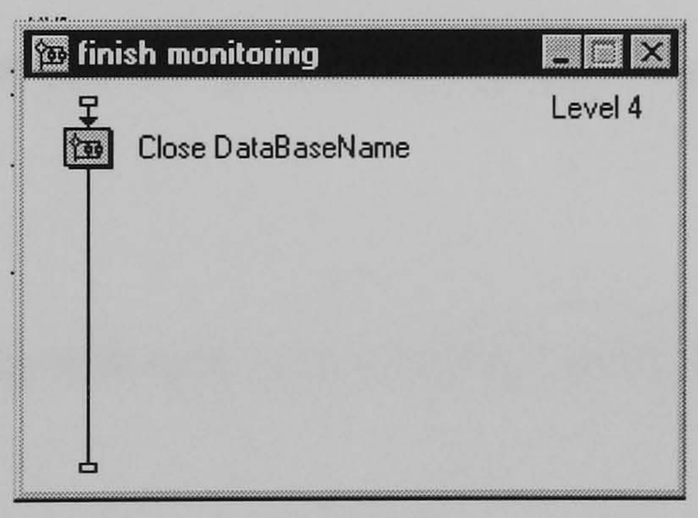


Figure 40: ICMS closing the session database

The collator application was implemented as part of the ICMS monitoring system (described in 5.2). The monitoring system provided training managers at Cable &

Wireless with the means to analyse which parts of the training their employees had viewed and how they had performed in four test modules. The collator successfully merged individual session databases into the master ICMS database. The incoming databases were in Microsoft text format or Dbase and the master database was maintained in Access. The database (shown in Figure 41) contains five tables; the attendance table (storing details about students who undertook the course), the DETAILS table (storing standard information about the session), and three tables containing the results of the tests.

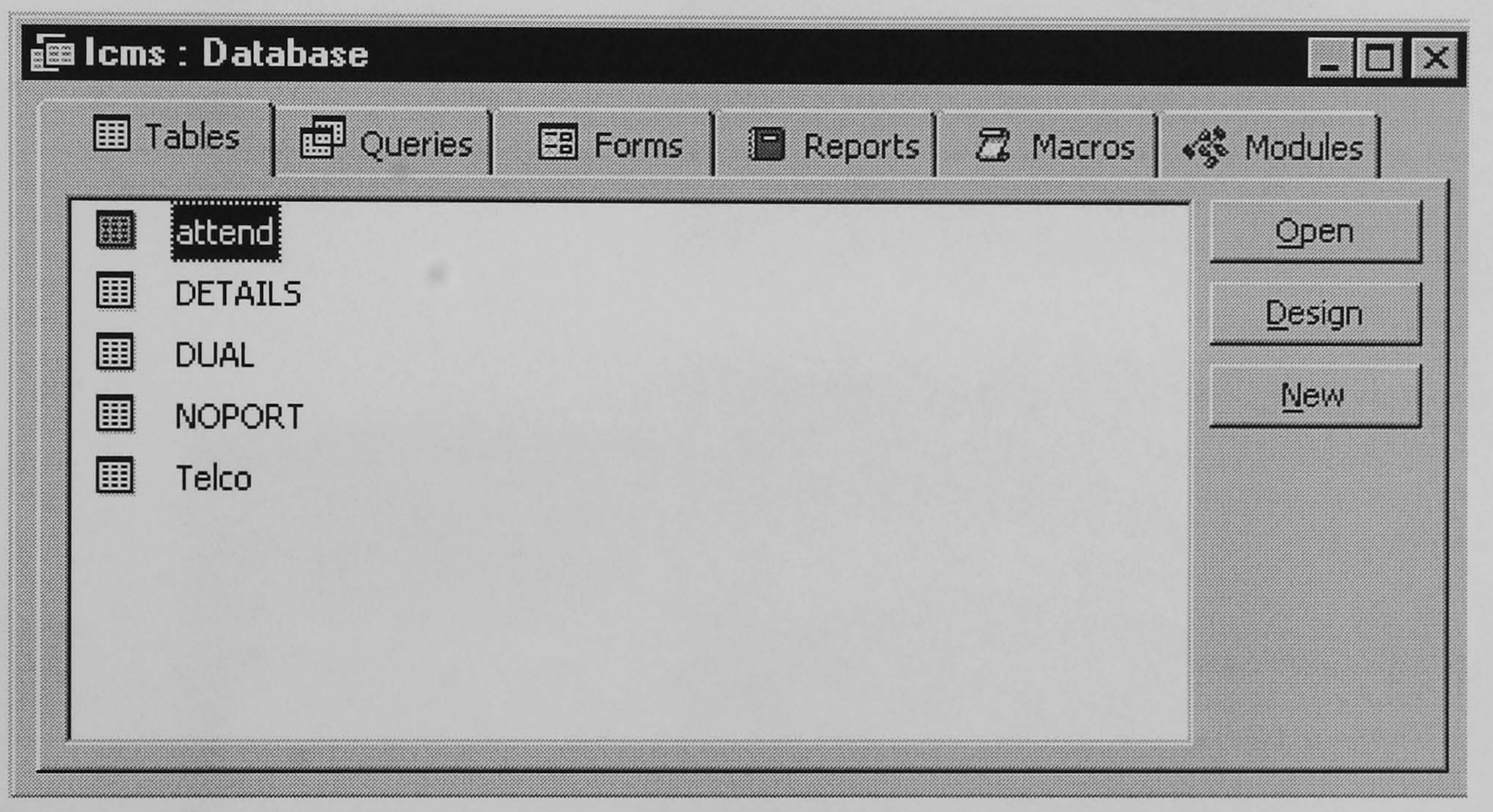


Figure 41: ICMS master results database

The requirement of the training manager was to obtain three reports from this data:

- Accumulative results for a specific user for a specific test
- Latest results for a specific user for a specific test
- Latest results for all users who sat a specified test

These three reports were constructed in Access with a small amount of Visual Basic programming. Forms were constructed to allow the training manager to enter the details in a user-friendly manner to facilitate the generation of the specific reports. These forms are illustrated in Figure 42.

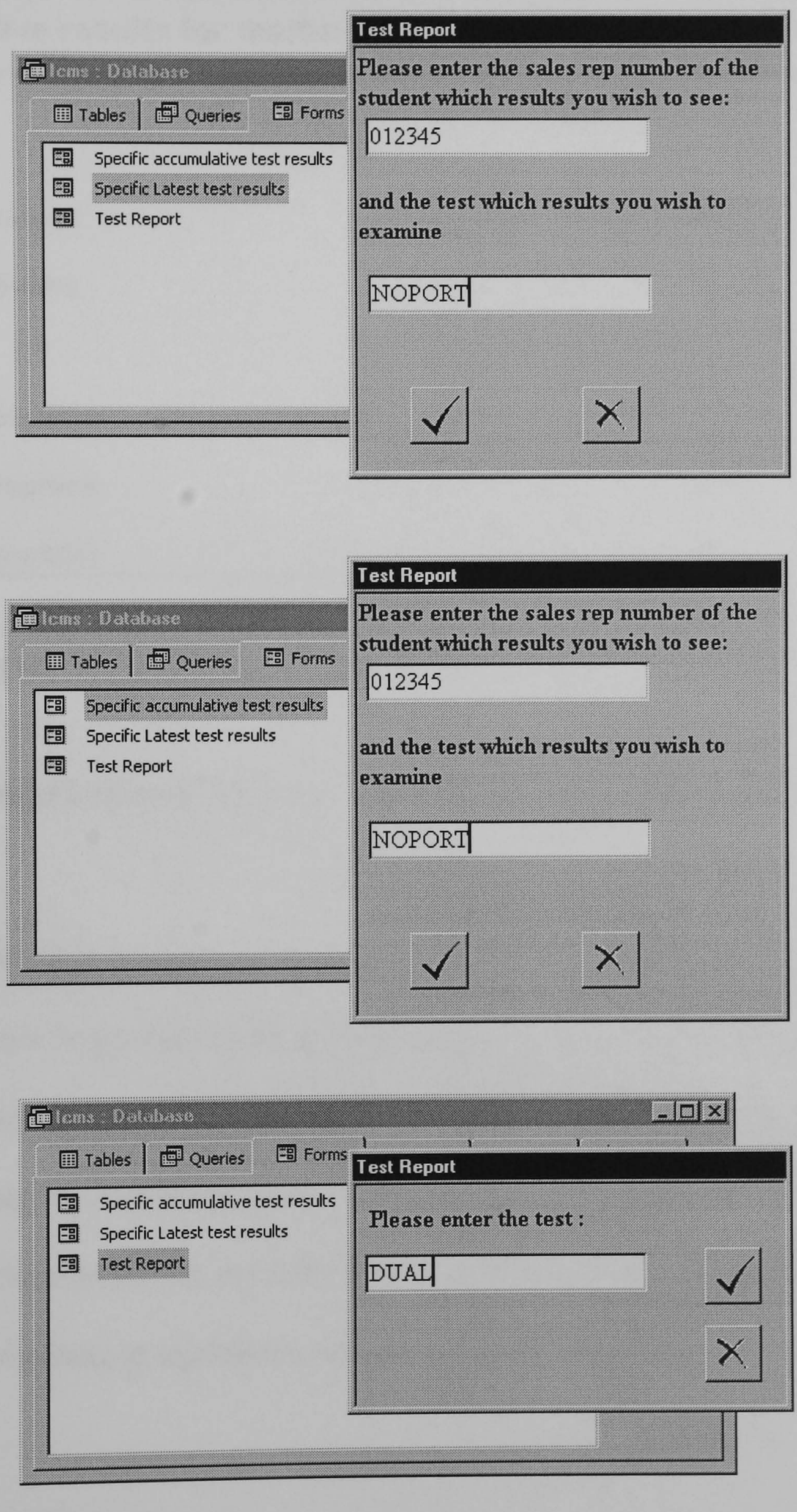


Figure 42: ICMS report generation forms

Through the use of these forms the training manager successfully produced tabular reports for both individual students and complete tests. Figure 43 shows part of a report for an individual who undertook the 'DUAL' exam.

Accumulative results for martin (Sales Rep no. = 123456) in test DUAL:

Result	Value
AccountMaintenance	6
AppointmentBooking	9
CATVSE	1
CATVServiceOrderEntr	8
CATVSEviewingServic	3
CustomerAccountDDD	0

Figure 43: ICMS part of accumulative report for test DUAL

6.3 Internet and Intranet Trials at The University of Huddersfield

A trial was undertaken at the University of Huddersfield in order to test how the system would perform in a large scale training environment. This was crucial in order to test how the system coped when used to train large numbers of students in many diverse locations and who are studying at different times. This is required because the true advantages of a training monitoring system are not fully realised unless it can be shown to work on such a large scale and resulting in significant amounts of student data being processed.

The trial was accomplished by using the TIS to monitor a computer-based learning session that forms part of an undergraduate degree course at the University of Huddersfield. The system monitored an application that students accessed from a number of different computer laboratories at various times. Some of them even accessed the applications over the Internet from home. The large amount of data that this exercise created was then analysed and presented in such a way that it was of best use to the module tutor and the student.

The computer-based application that was be used was an introductory module to Computer Networks, an integral part of the undergraduate course. The students were fully aware of the monitoring functions, and extremely interested in the trial, since they were students on multimedia pathways who are being taught how to develop multimedia applications. The CBT module itself consisted of five sections, each teaching the student about a specific network topic area, and one assessment section for the students to complete last. In addition, at the end of each section was a mini question relevant to that specific section's subject matter. The structure of the CBT application is shown in Figure 44, where only two section of content and the assessment section are shown for clarity.

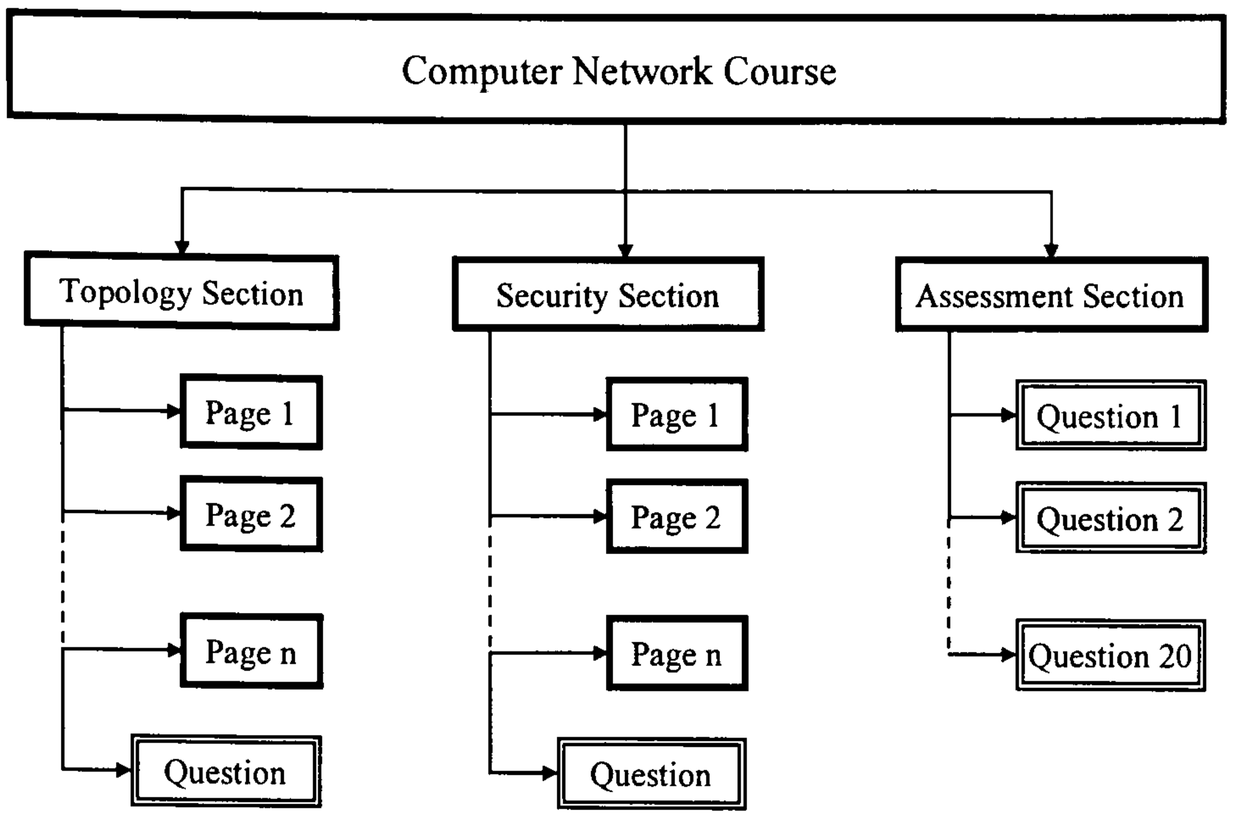


Figure 44: Structure of Computer Networks CBT application

The application was developed in Macromedia Authorware, and then prepared for WWW delivery using Macromedia's streaming software 'Web Packager' in order to allow deployment over the university web-site. The students then view the CBT application embedded within a WWW browser such as Netscape Navigator. Figure 45 shows the title screen and the main menu and the login script for the CBT application and demonstrates how the student views the application from within a WWW browser. The WWW browser requires an extra software component (called a plug-in) to enable it to understand the streamed data and embed the courseware.

The objective of the group lecturer, the training manager, was to clearly identify how much material students had absorbed and which students were struggling with work and which students were performing well. To achieve this the monitoring system was deployed to record which sections the student had viewed, the extent at which students explored sections (along with the time taken), how they responded to assessment questions and how long they took to answer each individual question.

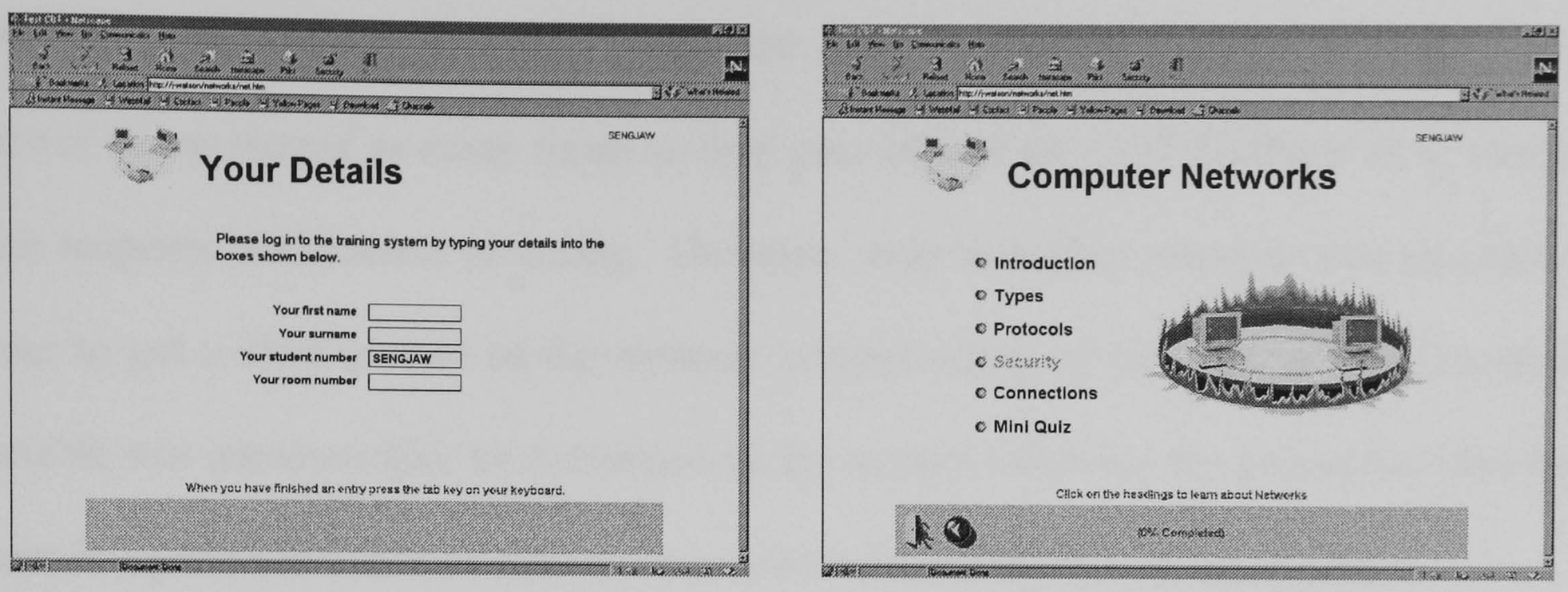


Figure 45: Screenshots from Computer Networks application

The data to be reported out of the CBT application was contained in four types of variables. To report visits, "Visits_SectionName" and "Visits_Max_SectionName" was used for each of the sections of CBT required to be tracked. Students are only allowed to navigate pages, within each section, in a sequential manner. As a consequence, the CBT application only required two variables to indicate what a student had viewed in a particular section: firstly, a variable containing the maximum number of pages the student could possibly view (e.g. Visits_Max_Security), and secondly, another variable indicating how many pages the student has actually viewed. This method was used to reduce the amount of information reported, decreasing network bandwidth usage and time taken to collate the results for analysis.

To report assessment responses, "Q_SectionName", and "Q_First_Answer_Array[]" type variables were used. These were used to hold the student responses from each of the mini-questions (delivered at the end of each section) and the student responses from the main

assessment. The five responses from the mini-questions were held in separate variables (e.g. Q_Security), and the responses from the main assessment, 20 in total, was held in an array (Q_First_Answer_Array[]). During the main assessment, students were allowed to answer the questions as many times as they pleased and received feedback at to whether their response was correct or wrong. However, only their first response was recorded in order to get a clear picture of the students understanding of the material. Each reported variable was automatically time-stamped by the system, allowing the amount of time they spent in a particular section or answering a particular question to be calculated.

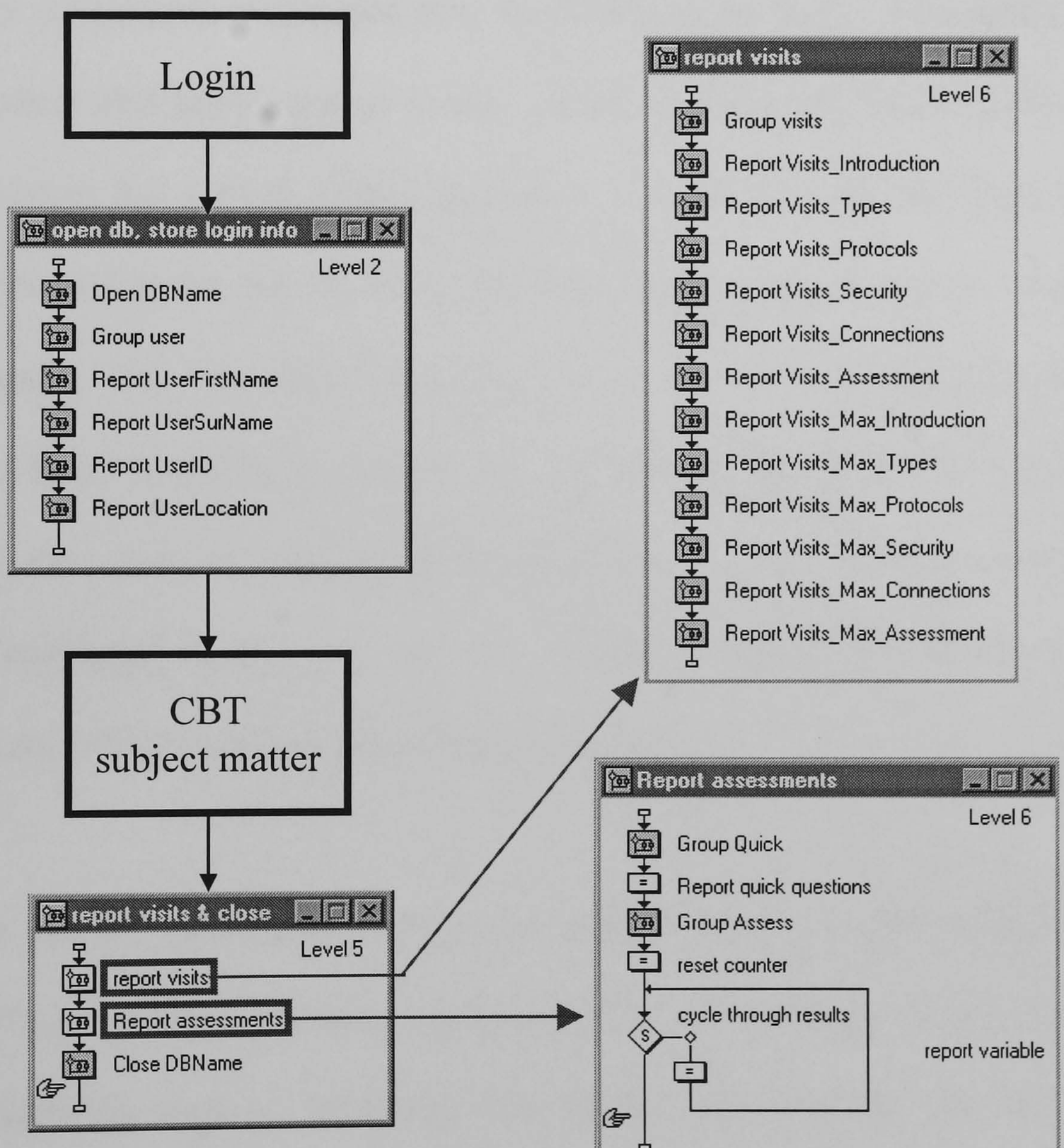


Figure 46: Overview of Monitoring System implementation

Figure 46 shows an overview of how the monitoring system was implemented in the CBT application. Firstly, upon entering the application the details of who the student is and which room they are training in is obtained from them via a login script (Figure 45). This information is further complemented by the TIS automatically retrieving the IP, unique identity, of the computer they are training on. In addition, the student is authenticated by the 'Username', which they used to log onto the computer they are using, being automatically retrieved from the computer network (as described in section 4.10).

The reported information is grouped into four categories: User – containing information about the student who has logged in, Visits – containing information about the sections and pages the student has visited, Quick Questions – containing the responses to the mini-questions delivered at the end of each section and Assess – containing the responses to the main assessment. The Computer Networks CBT application was placed on the university web-site and made available to students for a period of one week. During this time the students had the option to complete the instruction during a 2 hour allocated session, or during their own time. It was found that most students opted for the 2 hour session, where a total of 70 students underwent instruction simultaneously.

The collator system (described in 5.2) was implemented and successfully merged the results obtained from the individual student sessions into a single master database. The incoming databases were in Microsoft text format, the smallest and most bandwidth conserving database file format, and the master database was maintained in Access. The contents of this database were then queried in order to extract the performance information

that the training manager required. From these queries, several different data visualisation techniques, such as table reports and charts, were used to extract useful information for the tutor. For example, a chart was produced to show the performance of all the students in a particular question.

Figure 47 shows an example of this, where the tutor can instantly differentiate between students who have not got question 11 correct, those that have, and those that have not made an attempt to answer. This example was found to clearly indicate the level of acceptance of an individual question to the students as a whole, enabling the tutor to establish whether a question was too difficult, or too easy, and whether subject matter (pertaining to the question) had to be investigated.

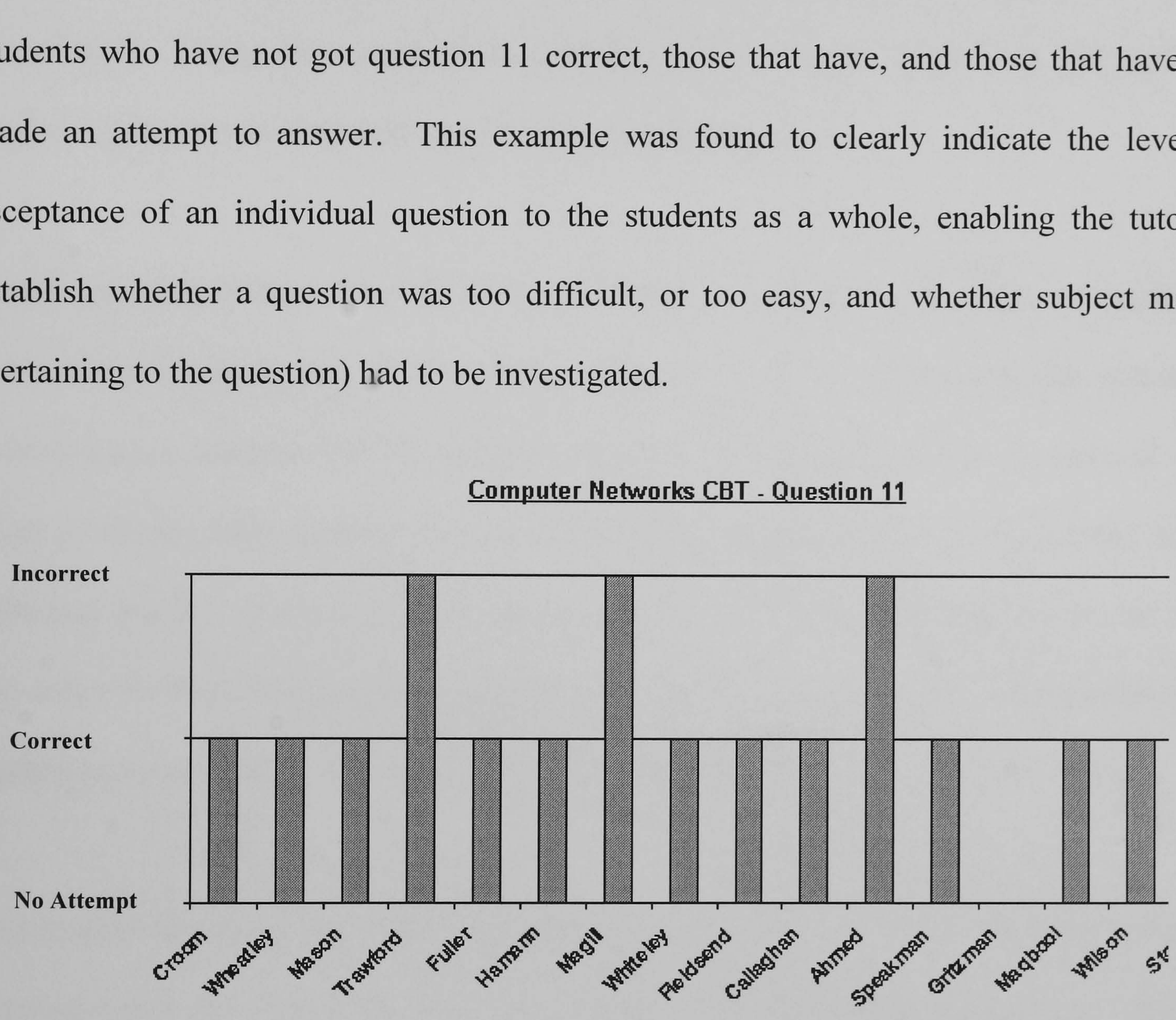


Figure 47: Bar-Graph visualisation of results from specific question

The chart in Figure 47 does not indicate the overall performance of an individual student. To establish the performance of an individual student and enable the tutor to see at a glance that student is doing well, or not, and which parts of the CBT application they are having difficulty with, the data visualisation technique shown in Figure 48 was used.

The monitoring system had recorded how much material each student had gone through in the CBT application (per section), and also the student's performance in the mini-question assessment, and the main assessment. All the assessment questions contained within the CBT application were associated with their relevant content section - the section that the student was required to study in order to gain the ability to answer that question successfully. Using these associations, and the student's results, it was a simple matter to create a query to summarise the results and produce this chart.

In this chart, the student "Mason" has his performance indicated by 5 circles, one circle for each section, or topic area, of the training. The size of a circles represents the amount of content/lecture-material that the student viewed in that particular section or chapter. The position of the circle indicates how well the student performed in the assessment linked with that section. If the student got all the questions for a particular section correct, then the centre for that circle is placed on 100 in the y axis. Conversely, the more questions the student got wrong, the further the centre of the circle moves down to 0 in the y axis.

The amount of training the student underwent, i.e. their exposure to the material, is directly proportional to the size of the circle. An accurate feel of how hard a student has tried, and how well they performed in the assessment, can be obtained at a glance. This visualisation technique is shown in Figure 48, which is the product of using Microsoft's Access 97 chart wizard on the master database.

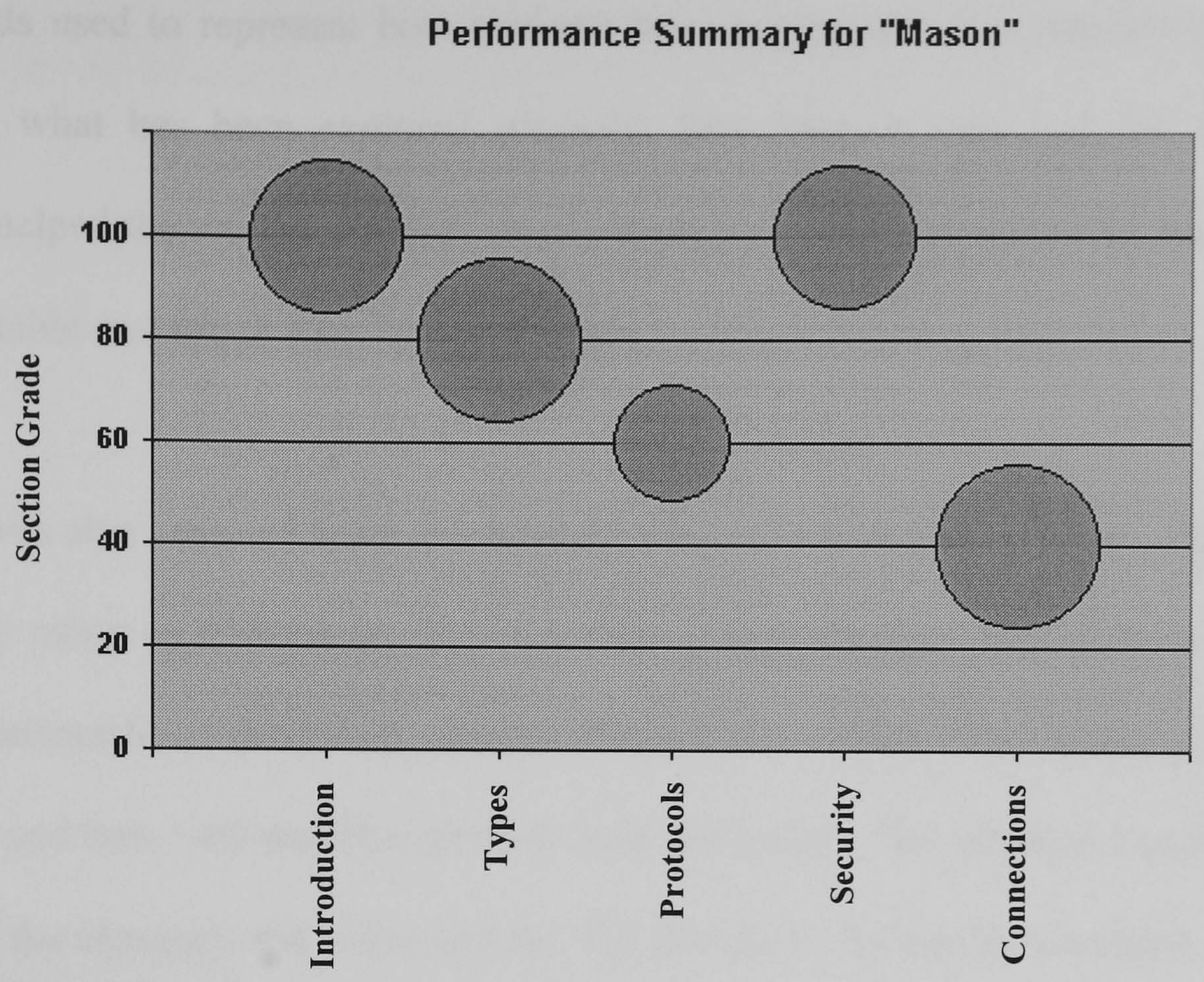


Figure 48: Visualisation of overall performance for a specific student

In this example, the student "Mason" who has only obtained 60% in the assessment related to the chapter on network protocols, has only a small circle for this chapter. It can therefore be concluded that the reduced mark could be attributed to the student "Mason" not completing the material on the chapter on network protocols. However, this student has got a large circle linked to the chapter on network connections, yet only obtained 40% in the assessment related to this chapter.

It can be concluded from this that the student has tried hard, but has some genuine problem with that subject area. The monitoring system was deemed to be highly successful by the group tutor who was able to keep track of students who were struggling with the work for one reason or another. The presentation of the results in the master database made it simple to identify these student in order to investigate and provide the appropriate support.

The methods used to represent how students have progressed, (i.e. monitoring what has been seen, what has been explored properly, how long it took and the response to questions) helped the tutor to identify clearly what kind of presentation the students found most acceptable and which they "glossed" over.

Feedback was also obtained from the students who undertook the training. This data was collected by using an anonymous feedback form (Appendix D). The form was designed to obtain information about two aspects of the trial. Firstly, the quality of the CBT application and how well they thought they had performed. This allowed an assessment to be made of the accuracy and value-add the TIS provided. Secondly, questions were asked to ascertain what the students thought of the TIS and their general impression of having their activities monitored.

From the feedback gained from the students on the forms, and at subsequent meetings, on the whole they enjoyed the training experience, not least because they could go over the work at their own pace and at a time and place that suited them. They also found the feedback of results useful, but it should be noted that a small minority saw the extent of the monitoring as an invasion of their privacy. This was the case even though it had been explained to the students that the monitored data would not be used by the module tutor for any assessment purpose.

6.4 *Relevancy of trials completed with Authorware*

The three trials discussed in sections 6.1, 6.2 and 6.3 involved CBT authored in Macromedia Authorware. A requirement of the monitoring system is that it should be capable of being used in any major authoring tool. This has not been validated in these particular tests but rather in sections 4.4.1 to 4.4.4 where the monitoring interface was validated with different authoring tools. The way in which the monitoring interface is implemented, irrespective of the particular authoring tool used, consists of an object which has a title dictating the monitoring action to be taken. In all authoring tools, this is a familiar component that meets the authoring tools analogy and with which the author is highly familiar with. For example, in Authorware the title of an icon on the flowline dictates the monitoring action to be performed, whereas in Director the title of a frame or an onscreen object dictates the monitoring action to be performed. The differences between the interfaces do not restrict the monitoring capability that can be incorporated into the CBT. It is therefore considered a superfluous exercise to repeat the trials completed with applications created in Authorware with applications created in other authoring tools in the light of each interface being validated independently and their inherent similarity.

7 Discussion of Results

This chapter examines the performance of the TIS and the value add gained through its use to the customer (training manager), the CBT author and the students. An analysis of current trends in CBT authoring is discussed, highlighting new technologies and how the tools and methodologies for authoring have changed in the last three years. Finally, comparative systems that have emerged since the beginning of this project are discussed and compared to the generic monitoring system.

7.1 TIS Performance

The Generic Training Information System provides a viable solution for the monitoring and tracking of activities in computer based training. It provides mechanisms for the three fundamental parts that must be present in any such system:

- Reporting
- Delivery of information over computer networks
- Collation and analysis/production of reports

The impetus behind the design of the reporting system was that of flexibility. There are a number of popular authoring languages being used to develop a huge variety of Computer Based Training (CBT) applications all which have different monitoring requirements. The reporting system was designed so that it could be integrated into any authoring language, and fulfil any reporting requirement. If the authors had been provided with a less flexible

system and were forced to use a rigid framework, there is no doubt that they would have become frustrated and the system would have been less accepted. The reporting system allowed the reporting of any variable from the CBT application, and allowed the author to structure this information how they saw fit.

This factor combined with the simple to use interface, made it extremely easy to integrate the reporting system into CBT applications. This was the case even if the CBT application had not specifically been designed with a monitoring system in mind. This flexibility and the ease with which it could be integrated into the courseware is most certainly the system's greatest selling point.

The collator application was deemed an extremely flexible way of collating the individual session databases together into a single entity for analysis. The major advantage of the collator application was that it needed very little setting up and maintenance. New CBT applications did not have to be introduced to the collator. The collator could efficiently process data from CBT applications of which it had had no prior experience.

The collator creates the master database based upon the format of the first incoming data it receives, so the collator application should need minimum alterations irrespective of the kind of monitoring needs of future applications. The flexibility of the collator was increased even further by its ability to automatically detect and read / write and create the majority of database formats through the use of ODBC.

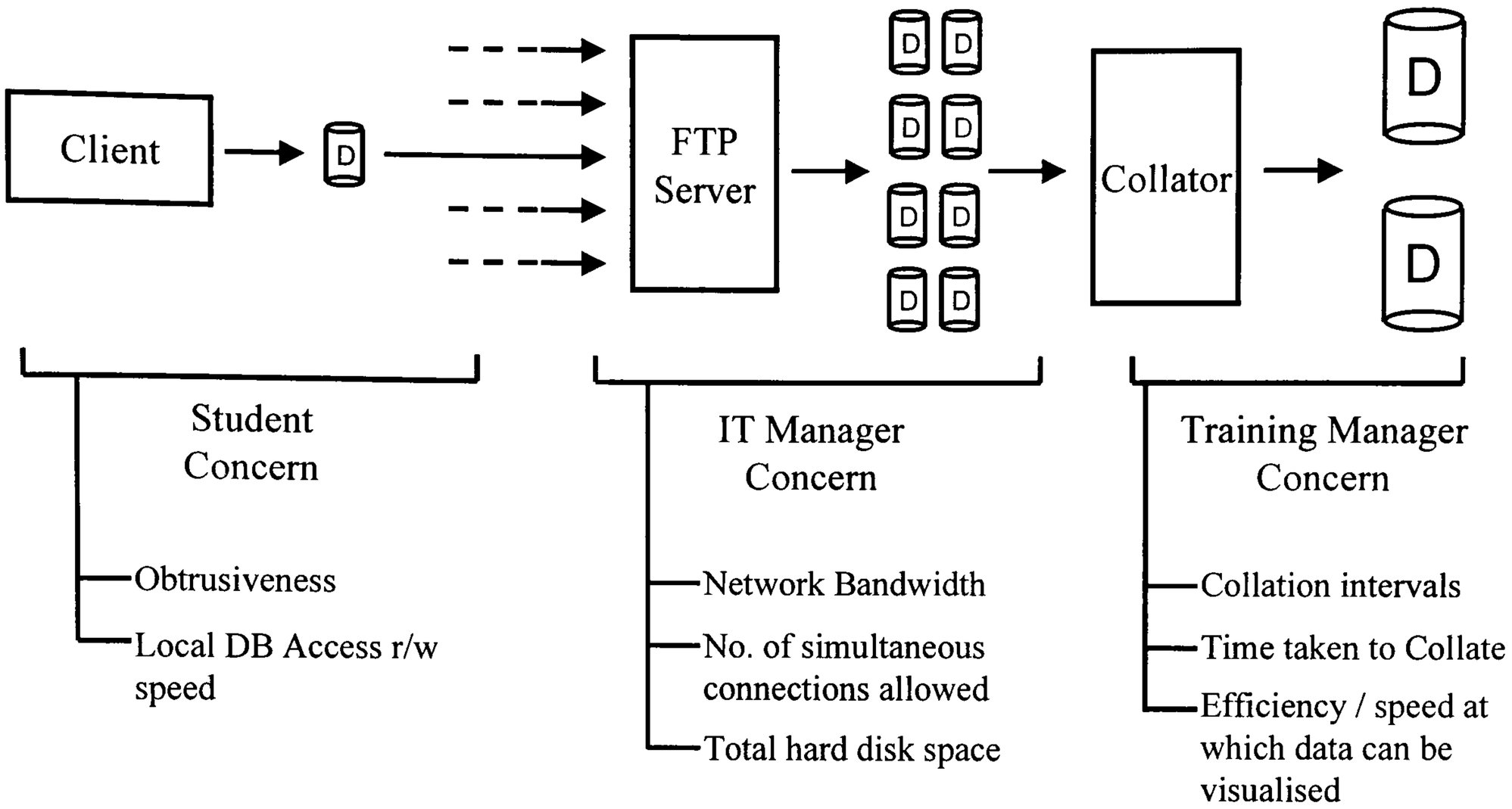


Figure 49: Deployment Issues

Apart from issues revolving around the authoring and integration of the system with the CBT, there are several quantitative factors that affect the overall deployment and performance of the monitoring system. These are demonstrated in Figure 49, where they are grouped into areas of concern. These groupings are only general guidelines, highlighting those factors that are of most concern for the student, the IT Manager and the Training Manager.

For example, the student may be as equally concerned as the Training Manager that their activities have been efficiently interpreted so that they can receive feedback, or recognition, for the work that they have completed. However, the main aspects from a students perspective is that any monitoring system should not distract them or detract from their learning experience. It is therefore extremely important for the system to be as transparent and unobtrusive to the student as possible. For example, login information and

user authentication should be done automatically and the monitoring system should not noticeably slow down the CBT.

The amount of interruption can depend on the time taken to read, write and send the local database files. This length of time was not only dependant on the amount of information reported, but also the choice of database file format. Considering only the monitoring system, network bandwidth, on the student side, is unlikely to be an issue. This is because the actual CBT content will be an order of magnitude greater than the demands of the monitoring system. However, at the administrators site, where hundreds, perhaps even thousands, of these small databases are being received, there could be a bottle-neck and network bandwidth may be an issue.

An IT manager would be interested in exactly how much network bandwidth would be consumed, how many simultaneous connections the FTP server is expected to take and also the total storage requirements for the data. These factors again depend on the amount of data reported, the type of database format used to contain the data, and also the total number of users and simultaneous users of the system.

The training manager is most concerned with how to manage the collator software and update the master databases so results can be analysed. The length of time it takes to collate data depends on how many individual session databases have been delivered since the last collation, and again their size and file format. In addition, the speed at which queries and reports can be performed on the master database is inversely proportional to its size. In summary, the concerns listed in Figure 49 are very much affected by the reported

data volume, the file format of reported data and the number of users and simultaneous users of the system. We shall now have a closer look at system performance in each of these areas.

7.2 Data Volume

The system provides great flexibility in terms of what the author may report. However, this can have adverse implications on file sizes as the amount of data reported is dependant on what the author or training manger dictates. It is important to keep the file sizes of the reported data to a minimum and it must be impressed upon the authoring team to only report information that is really necessary. File sizes can be decreased, and monitoring system performance increased, if the CBT summarises the data prior to outputting the information via the reporting system.

For example, a CBT application may have obtained 20 responses to a set of 20 questions. In this case, the most compact way to report student performance would be to simply report the total mark obtained in the test. However, this may not suit every application as there may be times when the actual answers given could be of interest to the training manager, and also other additional information.

The individual, local, database of reported data consists of both a pre-defined section and a user-defined section. The pre-defined section, containing information the system automatically obtained about the user and session, contains a minimal amount of information (less than a Kilobyte). Far more significant than this is the user-defined

portion of data which can range from zero to any amount determined by the author of the CBT application. However, every single piece of information reported is automatically time and date stamped by the system, adding a further 15bytes. For example, the "Display Screen Equipment Health and Safety" CBT (section 6.1) produced data records, for each session, of approximately 17 Kilobytes, 94% of which was user-defined data. However. this figure does not include overheads attributed to the particular file format selected, which shall be discussed next.

7.3 File formats

ODBC was selected for being an extremely flexible and interoperable way to report data directly into just about any database format. Through the use of ODBC, the reporting system leaves the choice of database file formats completely up to the IT manager or training manager. However, there are certain implications discovered during the trials that make the use of certain file formats more practicable than others. The most important criteria to examine, when choosing which file format to use, is how the file size and read/write access times will be affected.

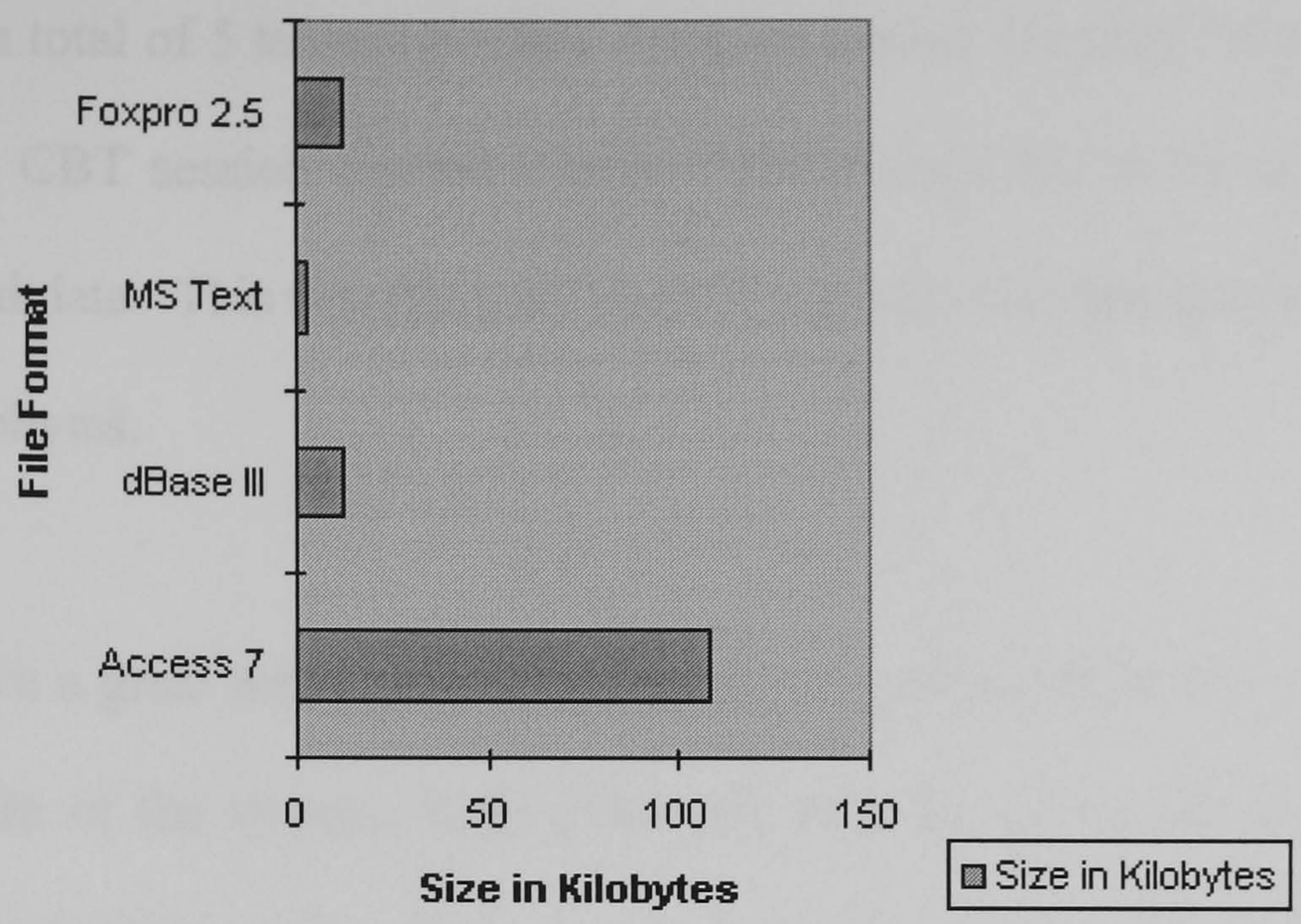


Figure 50: Comparison of ODBC File Format Sizes for fixed data

Figure 50 compares four different database files, each of which contain exactly the same data. The data itself is only 2 Kilobytes. It can be seen here that the use of certain fully fledged database file formats can introduce large overheads. The file size of reported data can therefore be reduced by the selection of a database format that takes less disk space. The impetus in storing the data in a database format at the student side was to cater for future functionality, allowing the author to interrogate or manipulate the file, during runtime, as a fully fledged relational database.

However, any database file carries overheads, some considerable, compared to that of a proprietary flat file format. For example, an empty Access (version 7) database was found to take up a surprising 120 Kilobytes of disc space. The equivalent for a flat file is of course zero Kilobytes. This was not just a feature of Access 7, other database formats such as dBase and FoxPro were found to have unacceptable overheads. These overheads are not fixed, and increase radically when you start to add tables. For example, in the trial at Cable and Wireless Communications PLC (section 6.2), the ICMS CBT output student

activities using a total of 5 tables (4 tables + the mandatory 'Details' table). It was found that each ICMS CBT session created a huge 730 Kilobyte file when using Access 7 to store the reported data. This was the case, even though the data that had been reported was less than 10 Kilobytes.

These issues have a great influence on the concerns listed in Figure 49. Considering first the reporting side of the system, large Files will take an significant amount of time to upload to the administrators site. For example it would take over 2 minutes to upload the 730 Kilobyte access database produced by the ICMS CBT over a modem connection (this assumes an FTP upload rate of 6 Kilobytes a second, which is considered a good transfer rate when using a 56.6 KBPS modem connected to a telephone line). This would provide the student with an unacceptable delay. However, if the student is connected via a LAN or an Intranet, the upload time would be reduced considerably, depending on network traffic conditions, to perhaps only a second. There are, however, other disadvantages to having such large file sizes which shall be discussed later when considering the effects at on administrators side.

Large file sizes are obviously unacceptable, so tests were performed to find a compromise between the advantage and flexibility of having a database and the size of the resulting file. The 'Microsoft ODBC Text Database' format was found to have the smallest file-size of any of the databases without any significant increase in overhead if numerous tables were used to report the data. In addition to file format selection influencing file size of the reported data, different file formats were also found to influence the speed at which data

could be reported. In fact, the speed at which variables could be reported out of the CBT application was found to be inversely proportional to the file size.

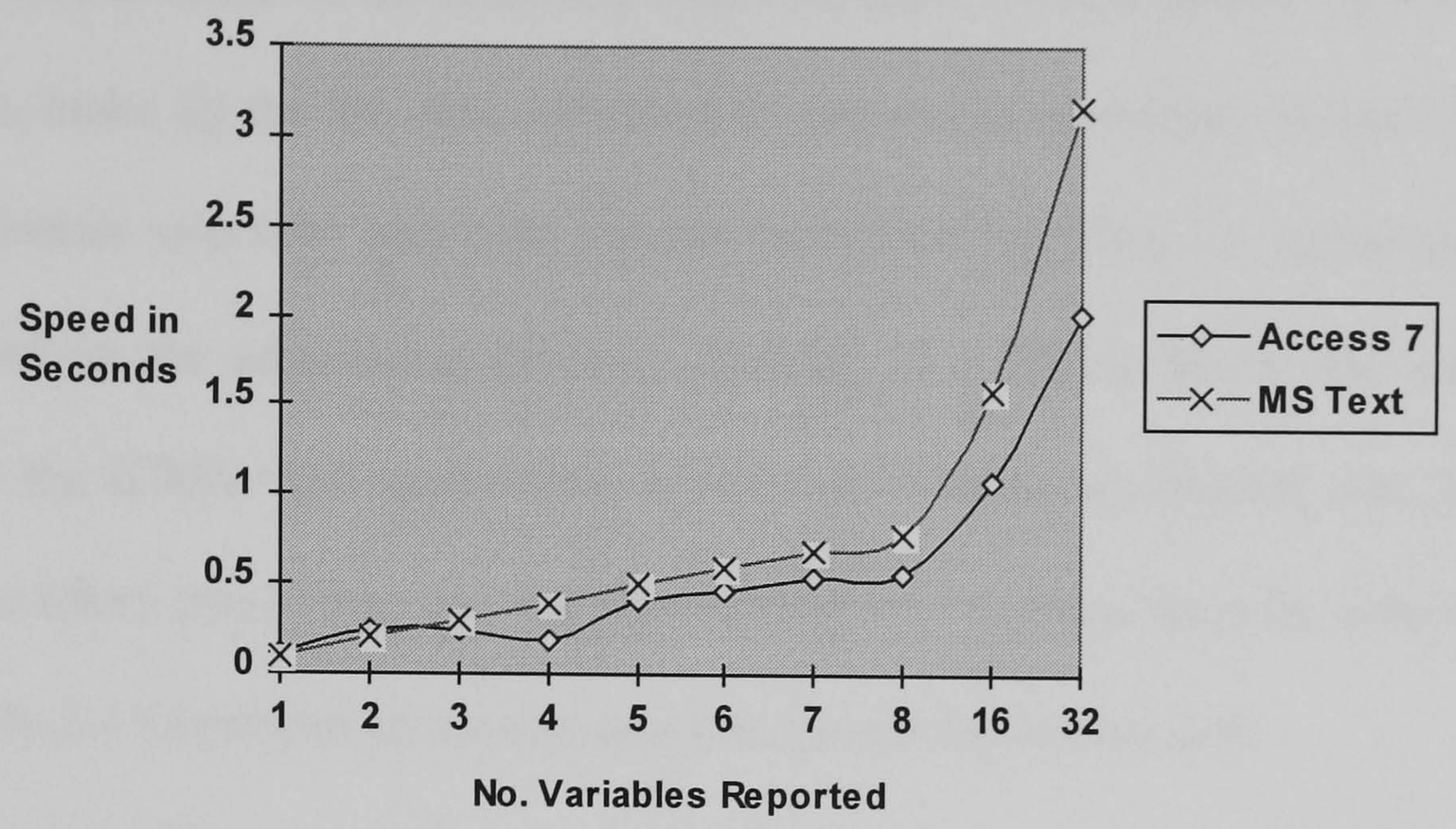


Figure 51: Reporting delay for different variable numbers

The 730 Kilobyte Access 7 file produced by the ICMS CBT application was instead 11 Kilobytes when reported in Microsoft ODBC Text Database format. However, the time taken to report data into file was found to take significantly longer in the Microsoft ODBC Text Database format. Figure 51 shows the length of time and interruption experienced by the student when reporting different numbers of variables from a CBT application. There are two graphs, one for an Access 7 database and another for an MS Text Database.

It can be seen that writing into the small MS Text Database takes a third longer than writing into the Access 7 database when writing 32 variables. The time taken to report the information was found to be proportional to the number of variables, and volume of data, that is being reported. However, the time taken to report information in MS Text file format was found to climb quite steeply compared to that of the Access file format.

It is important to minimise this time as much as possible in order to be unobtrusive to the learning environment. However, the speed increase experienced by using the Access 7 file format does not, in terms of the relatively small amounts of information reported during a CBT session, make up for the unacceptable file size and consequently longer FTP upload time. File format selection, and data volume, also have an affect on performance of the collator based on the administrators side. Storage of a 730 Kilobyte file, such as that produced by the ICMS CBT application, is not a problem at the student side, but if 2000 sessions have taken place since the last time results were collated then the data would take approximately 1.4 Gigabytes of storage space at the administrators side.

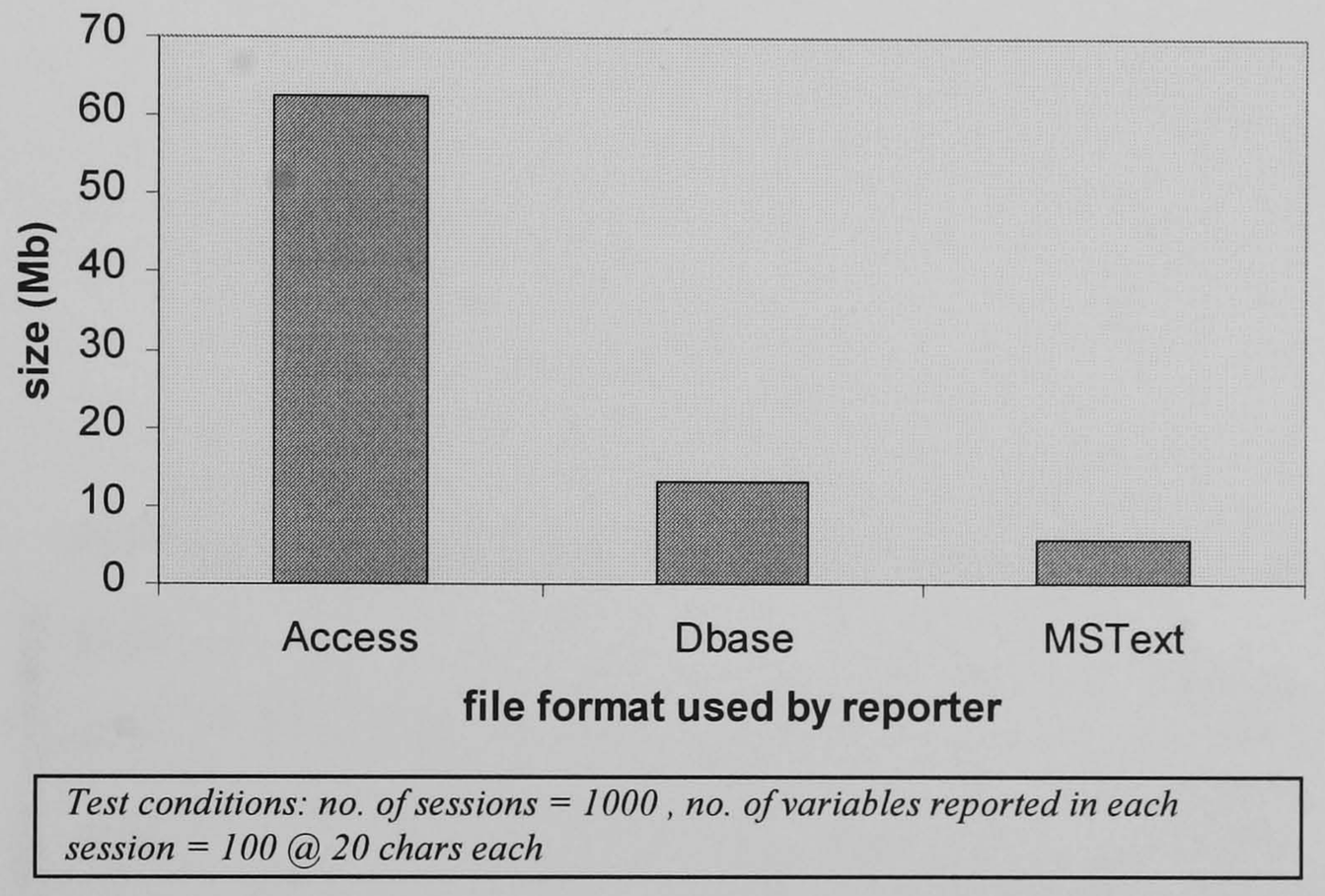
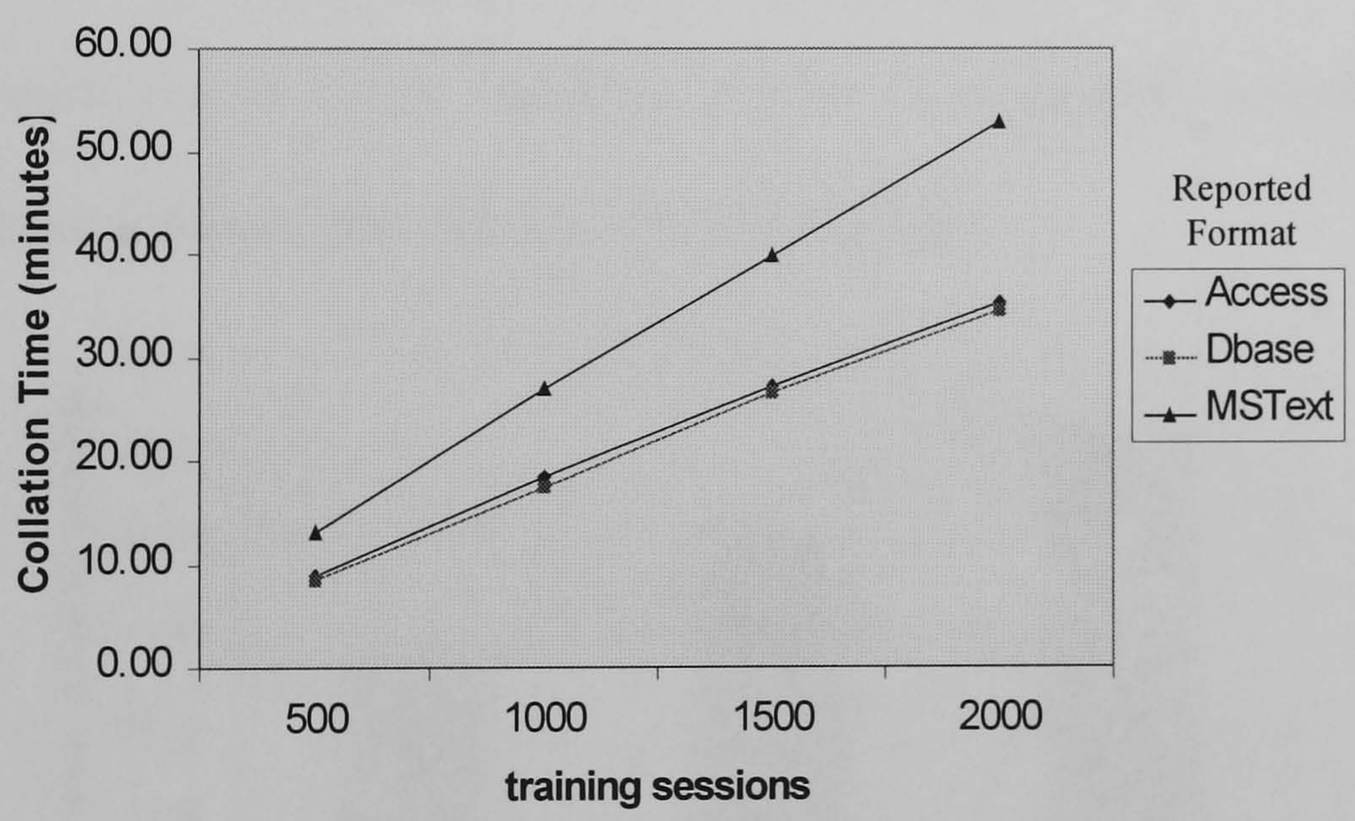


Figure 52: File Format influencing administrator-side storage requirements

Figure 52 shows the results of an investigation into how the file format of the reported data, in three popular database formats, influences the administrator side storage requirements. It is apparent that Access takes a massive 62.6 Megabytes of disk space to store what is effectively, at 1000 sessions of 100 variables at 20 bytes each, 1.9 Megabytes. The other file formats are far less at 13.1 Megabytes for Dbase and 5.81 Megabytes for MS text. It can be concluded from this that MS text database format is the most desirable

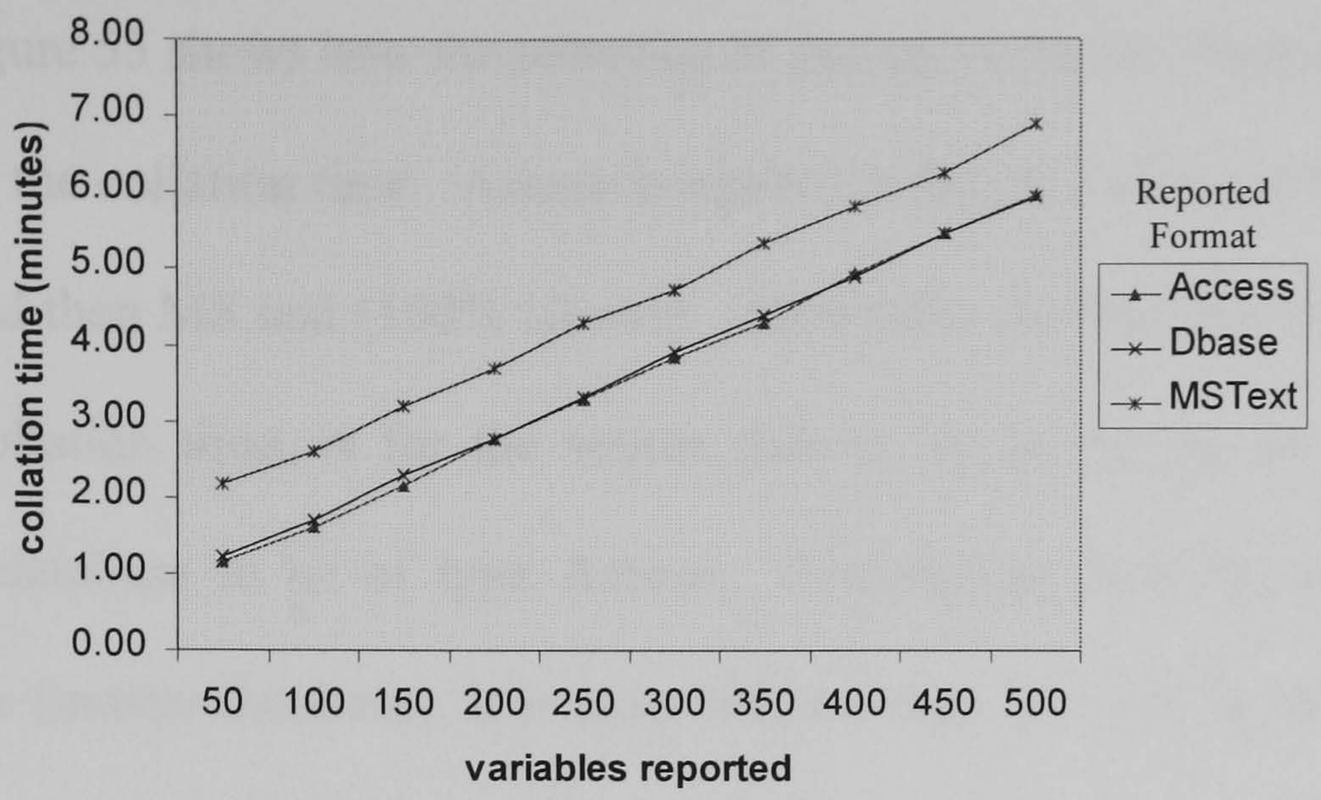
format to store the data in prior to the collation process, certainly from the perspective of minimising upload times and server side storage requirements.

Another important factor to consider is the collation time and final size of the collated database (master database). These factors are dependant on the number of sessions collated, and their size, and also on the source and destination file formats. Collation time was examined for different source database formats, different numbers of sessions and also different data volumes. Figure 53 demonstrates how the time taken to collate results increases linearly with the number of sessions received since the last collation process took place.



Test conditions: no. of variables reported in each session = 100 @ 20 chars each, master database was in access file format

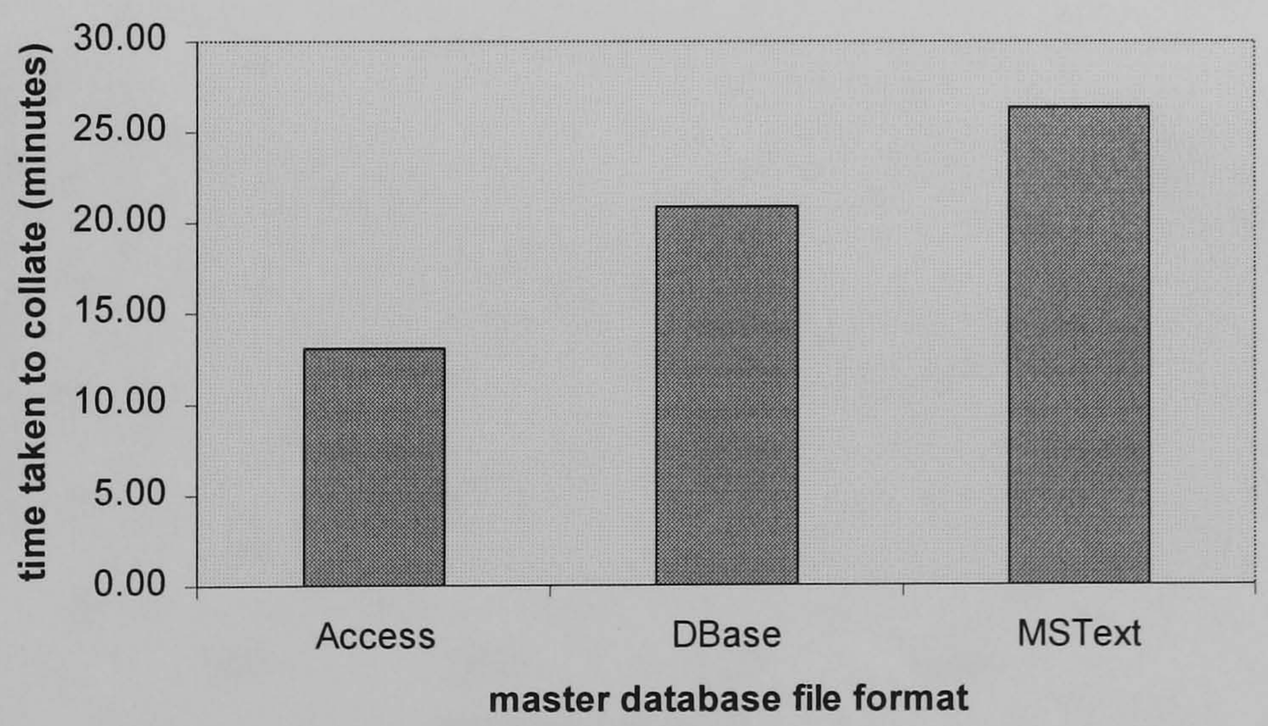
Figure 53: Collation time Vs no. sessions and reported file format



Test conditions: no. of sessions = 100, each reported variable was 20 chars in size, master database was in access file format

Figure 54: Collation time Vs data volume and reported file format

Access and Dbase file formats were processed by the collator at approximately the same speed and far more quickly than MS text format. Figure 54 confirms that this as the case, even if the increase in total input data is due to the number of variables reported, instead of the number of sessions. The graphs in Figure 53 and Figure 54 were recorded with the collator set to output into an Access database. Further investigations revealed that the file format of the output database also influenced collation time.

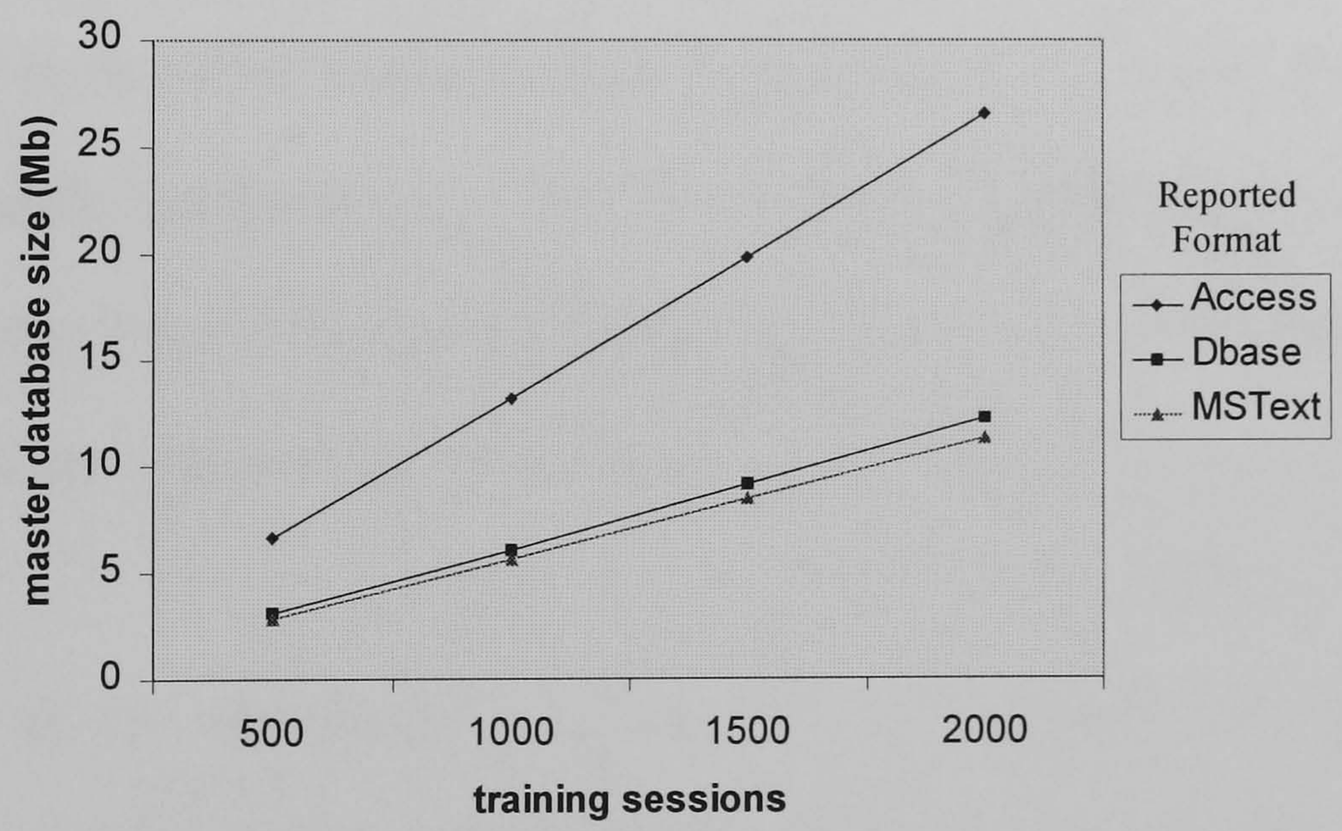


Test conditions: no. of sessions = 500 (MSTextformat), no. of variables reported in each session = 100 @ 20 chars each

Figure 55: Collation Time Vs File format of destination (master) database

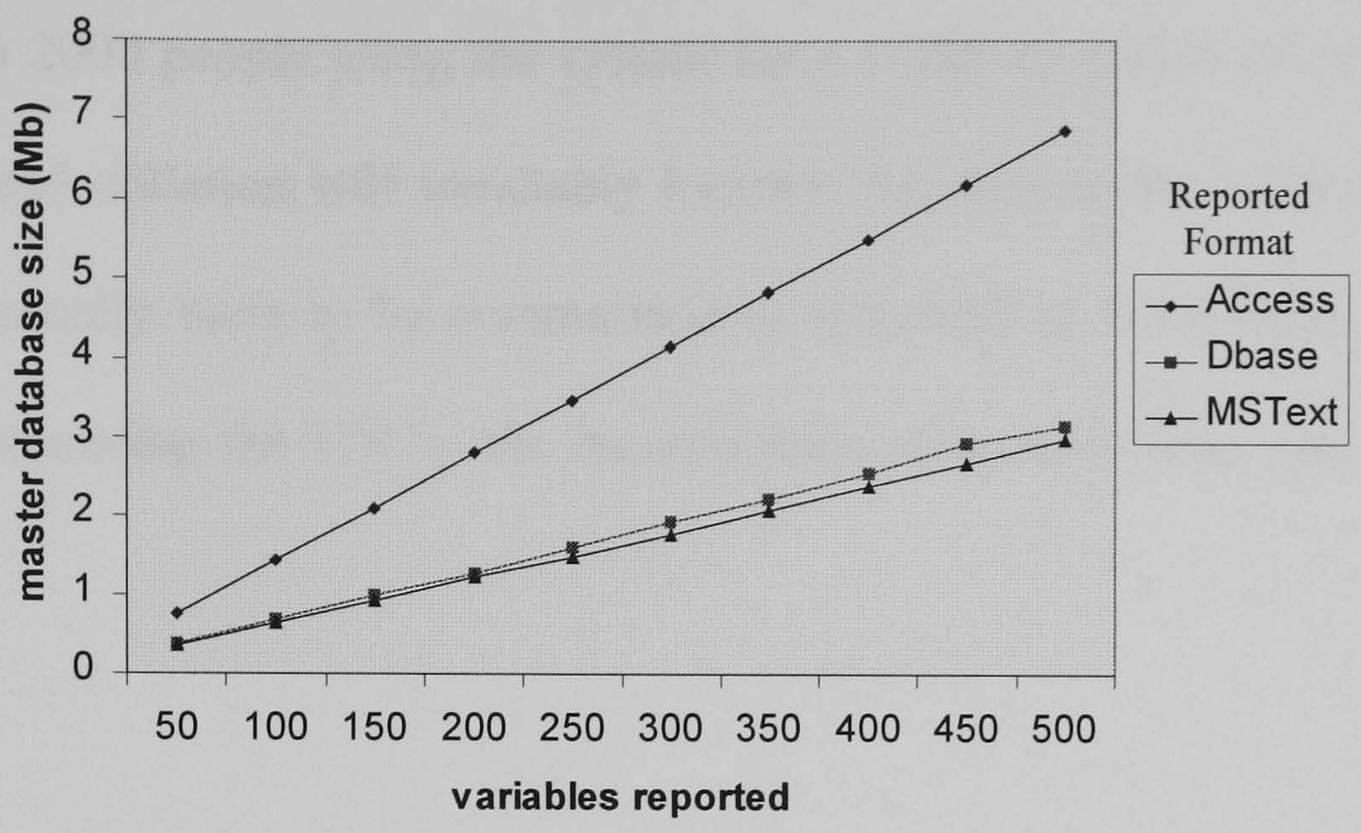
For example, Figure 55 shows how the selection of output, or master database format, does indeed influence the collation time. Access is again the fastest choice, followed by Dbase (60% slower) and then MS text (100% slower). Therefore, the best combination, in order to reduce the collation time, is for the source data to be in Access or Dbase, and the selected output database to be of type Access. Considering now the size of the final collated database (master database), it is quite evident that this will be influenced by the total amount of input data – both the number of sessions and the size of each session.

The size of the master database was examined for different source database formats, different numbers of sessions and also different data volumes. Figure 56 shows that the size of the master database increases linearly with the number of sessions received since the last collation process took place. This is also the case if the size of the sessions is increased (Figure 57). It was found that if the source database format was of type Access a considerably larger master database would be produced.



Test conditions: no. of variables reported in each session = 100 @ 20 chars each, master database was in access file format

Figure 56: Master Database Size Vs Number of Sessions



Test conditions: no. of sessions = 100, each reported variable was 20 chars in size, master database was in access file format

Figure 57: Master Database Size Vs Data Volume

In conclusion, system performance is very much influenced by the type of database formats that are implemented, and also the overall amount of data reported. However, factors such as speed and file size are of far more importance on the student, or client, side of the system because the weakest link in the chain is the time required to upload the data to the server. It is, therefore, crucial to keep the reported file size down to a minimum by choosing the correct database format. Ms text file format was found to be the best ODBC option for reporting data out from CBT applications, providing the smallest file sizes and least upload time and interruption to the student.

The conditions on the administrators side are far less restrictive, as increased storage capacity and computing power make file sizes and time taken to collate increasingly insignificant. For example, even with the slowest combination of options (MS text input and an MS text master database) the collator could collate 500 sessions in less than 30

minutes. Assuming that the collator was run by the training manager once a day and that a maximum of 25% of all users produced a single session (of 2 Kilobytes) per day, this would equate to 2000 people using the system for a collation period of only 30 minutes. However, repeated collation will inevitably increase the size of the master database, and data would eventually have to be summarised or removed at some stage. Future work could include enhancing the TIS by the incorporation of an automatic way to accomplish this.

7.4 The Delivery Mechanism

The delivery mechanism for the reported data was essentially File Transfer Protocol (FTP). This option was selected because FTP is widely accepted as a reliable and safe method for delivering files on an intranet, or the Internet. The thrust of the whole design of the monitoring system was to use open standards and avoid the use of obscure file formats, or communication methods that would limit its compatibility with existing systems or its acceptability by customers.

The use of FTP was found to be the most successful way of transferring data from the client (student) to the server (administrator site). If a file failed in its transmission then the monitoring system simply added data to the that file during the next session, before trying to re-transmit the whole lot at the end of that session. The speed of transfer depends on the speed of the underlying link, the traffic of all other users on that link and also the overall concurrent load on the FTP server. The physical connection of the server to the network is likely to be of much higher capacity than that of the individual clients. Therefore, any

bandwidth limitation will probably be on the off-ramp to the students computer. For example a 56Kbps modem will never transfer information as fast as an ISDN connection.

It is, however, feasible that a bottle-neck could be formed by the sheer volume of traffic encountered by receiving session databases from numerous training locations. In addition, 50 students uploading data to a single FTP site simultaneously will effectively load the computer that the server application is running on. In fact, it is usual for IT managers to restrict concurrent FTP connections to about 30 users. These issues are not a great threat as it is unlikely, even with the TIS delivered on an enterprise scale, for their to be more than 30 concurrent connections. This is especially so with the relatively small amount of data reported out of the average CBT application and the small duration of connection times. In addition, if the FTP server can be set to limit the number of concurrent connections in order to achieve a minimum performance level. If this level has been reached the reporting system will fail to connect and, instead, will simply store the information locally and attempt to re-send the information at the end of the next training session. The TIS will simply transcend above the connectivity limitations of the FTP server. Should this method not be sufficient, the TIS could be set up to report into more than one database within a single session. The CBT courseware or test modules could fragmented their reported information into sensible groups and deliver these groups to alternative locations and different FTP servers. However, this solution may not be ideal as the cross analysis of data between the separate master databases would be extremely difficult.

7.5 System Trials

The validation process consisted of the monitoring system being tested in several training scenarios. A small-scale trial was undertaken at CBL Technology Ltd where the monitoring system was applied to Health and Safety CBT for deployment on the Internet/intranet. This trial acted as a prelude to much larger trials performed at Cable & Wireless Communications PLC, where a customer service CBT application was deployed on a LAN, and also at the University of Huddersfield, where first year undergraduate lectures were deployed on a intranet and the Internet.

These trials demonstrated how easily authors, non-programmers, could use the tools provided by the generic monitoring system to incorporate monitoring functionality into their applications. In addition, the CBT applications tested were quite different from each other, ranging from corporate health and safety training, Integrated Customer Management System simulation and academic lectures. This measured how well the system would cope with the wide variety of applications that CBT is being applied to today. An equally important factor, in addition to the flexibility the monitoring system offered in terms of its application, was the flexibility it afforded the author in the selection of the authoring development tool.

At the beginning of this project, a large amount of research was undertaken into the wide range of multimedia / training authoring tools on the market (section 3.1). As a consequence, the interface to the monitoring system was designed so that it was compatible with the vast majority of these tools. Experiments were conducted with the

most popular tools and interfaces were developed for use with Authorware, Director, ToolBook Instructor, ToolBook Assistant and a variety of different programming languages.

All these interfaces provided a high level of detachment (black box) from the internal workings of the system, allowing the author to use the monitoring system via tools he/she would normally already be familiar with. This flexibility was provided at the request of the collaborative company, CBL Technology Ltd., who develop products using a wide variety of development tools ranging from Authorware and Director to low level programming languages such as C++. CBL Technology Ltd. did not wish to be tied into using a specific authoring tool simply because of the TIS system that they had adopted. For example, there are plans for the next version of the Display Screen Equipment (DSE) Health and Safety CBT application, which has been integrated into the TIS, to incorporate extensive use of virtual reality. As a consequence, it may not be possible, or cost effective, to continue development of this product in Macromedia Authorware and a different authoring tool may need to be employed. If this is the case, the flexibility of the TIS will allow students who train in CBT developed in any authoring package to have their results recorded into the same database.

The analysis of the monitoring system performance, during the validation trials, led to several issues, and many important lessons about the monitoring of computer-based learning material, that were then fed back into the design of the monitoring system in an attempt to further improve its performance. For example, the large scale tests at Cable and Wireless and also at the University of Huddersfield led to research into which database

formats were best in order to increase access speed and reduce network bandwidth. These are factors that are essential in maintaining system transparency to the student.

Lessons were also learnt in how best to monitor computer-based training regarding transparency and obtaining the desired outcome for both the student and the training manager. It was very evident when studying how best to monitor interactive software, in order to provide objective data for the evaluation of that software and users, that the choice of precisely what is to be recorded is crucial. The flexibility provided by the reporting system, although much needed due to the diverse applications that CBT is being applied to, makes it all too easy to generate vast quantities of redundant low level data. This can be highly detrimental to network bandwidth, and make collation and analysis of the data difficult and time consuming. It is, therefore, important for inexperienced authors to liaison regularly with the course training manager in order to ensure that his/her objectives are met whilst reporting only the minimum of information. Choosing the correct grain of reporting for interactions is, therefore, of central importance, as it affects system bandwidth, transparency to the student and the ease at which the required monitoring outcome can be obtained. It is also equally important to ensure that events are recorded in sufficient detail although this problem is far less common.

There were also other factors that became apparent, during the trials, that could have some bearing on how the system is used and what should, and should not, be monitored. On the whole students enjoyed, and found extremely useful, the feedback the TIS provided. However there were a minority of students that saw the monitoring as an invasion of their privacy. In fact some students flatly refused to identify themselves to the system.

It is proper to assume that the person who is being monitored does have rights. The Association of Test Publishers [19] provides guidelines on what rights a "test taker" actually has. They suggest that a person should voluntarily agree to a test and that they should receive some information about the purpose of the test and how the results will be used. Further, they suggest that it is the test takers right to have the results of the test kept confidential except for the test taker, the experts who are authorised to give and interpret the test, and the individuals who are involved in the decision making process for which the test was administered. In addition to this, UK law, under the data protection act, specifies procedures for the holding and releasing of personal data that is held on computers.

It is indeed possible that the TIS could be used to invade peoples privacy, and perhaps even facilitate a criminal offence. A CBT author must therefore ensure that their tests are well-designed, used for their intended purposes, and do not ask questions that invade a persons privacy. Also, the full implications of the data protection legislation act must be understood before the implementation of the monitoring system [31 chapter 2.4]. Robin Chater, director of the Personnel Policy Research Unit states that "One of the greatest victims of modern computing power is personal privacy."[21]. Privacy issues associated with tests have been around long before the advent of CBT and computers, but the possibilities that computers provide for monitoring peoples actions, in every aspect of today's life, is unsurpassed and bound to make these issues far more pertinent.

One of the ways to address this issue is to monitor students on what they accomplish, rather than how they accomplish it. It does however depend on the monitoring outcome, as

it may be as equally important to record how a person does something, as well as what they have done. For example, how a student does something may be especially important in a computer-based simulation, or when obtaining information on how to improve a CBT application by examining how it is used.

British Telecom PLC (BT) have experienced similar problems with their staff not accepting CBT monitoring. BT has recently provided 3000 of its staff with technical CBT. Staff do not actually train on the web, but download materials via the Internet as and when they want them [25]. Staff are expected to connect back to the Internet in order to send their completed work, registering their training and enabling managers to monitor progress. A major issue for the BT project has been getting their users to reconnect. BT is currently experimenting with different techniques in order to encourage their staff to re-connect. Some of these techniques include issuing certificates, because people like to receive recognition, and not recognising that the training took place at all unless the work is uploaded.

The TIS is an extremely powerful tool and can be used to monitor students in many different locations and who are studying at different times. The system brings great cost and efficiency benefits to stand alone CBT, or conventional ways of monitoring, for several reasons. Firstly, monitoring training is repetitive as it is important to monitor and assess each person in the same way as any other. Secondly, the faster feedback and results are given to someone who has their performance monitored, the better they learn from it.

This feedback is also vital to assess the CBT itself and ascertain if the learning concepts employed are successful on the students. However, the true advantages of the training monitoring system are only fully realised when the training is deployed on a large scale, resulting in significant amounts of crucial data being processed. In addition, the use of a monitoring system applied to distributed training, or in fact any distributed-computer application, has considerable potential in establishing pay-per-view system. However, although the TIS goes some way to authenticate users by automatically obtaining login details from the network, and tracking which workstations are being used, further work would need to be undertaken to encrypt the reported data and increase security. As the system currently stands it is open to abuse by users who may try to "defeat the system" by manually altering the databases.

The TIS has a whole range of implications and advantages to all people involved in CBT. The system increases the efficiency and training effectiveness to the customer, allowing instant feedback on the performance of large numbers of distributed students and feedback on curricula effectiveness. This will give customers increased effectiveness and save them time and money. The system gives CBT developers a single methodology to report information out of diverse CBT applications using an easy to use, innovative and familiar, interface aimed at their own skill level, and using any popular authoring tool. Further, the student will benefit from impartial, fast and accurate feedback on their performance which will inevitably enhance their learning experience.

7.6 New Trends for Multimedia Authoring

Technology concerning multimedia production and delivery over computer networks is still relatively immature. As a consequence, since the beginning of this project, there have been many developments in how multimedia content, such as that used in CBT, is created and delivered. CBT material has been developed for many years using the authoring tools described in this thesis. Authoring tools allow a wide range of people, often with little programming skills, to assemble multimedia components, such as graphics, text, video, sound and animation into robust applications.

The authoring tools themselves provide the methodology for production and presentation of these applications, in their own way, using their own proprietary standards over a LAN or intranet/Internet. Hence the impetus (in section 4.4) to investigate the different ways which these proprietary authoring system could talk to the monitoring system, and the development of different monitoring interfaces. There are few, if any, standards between these different proprietary authoring systems making it extremely difficult to integrate content produced by different tools from different vendors. In fact, one of the greatest selling points of the TIS developed by this project is that it overcomes this problem by allowing CBT applications developed by different tools, from different vendors, to report into the same database.

However, industry has realised that great advantages can be gained by the integration of media produced in different ways, using different tools, by different vendors. This is because different tools are simply better at doing different things, with respect to graphics

processing, bandwidth conservation and other features. Microsoft Corporation has recently begun a new strategy where they intend to integrate support into the Windows operating system for two recently developed formats, the Advanced Streaming Format (ASF) and the Advanced Authoring Format (AAF). AAF is an open, industry-developed format that enables exchange of rich media among digital production tools and content creation applications, such as authoring tools [20].

Microsoft are hoping that ASF and AAF will become the new defacto multimedia file formats. This is debatable, but the standards will most certainly replace the extremely popular Audi Video Interleaved (AVI) file format. However, by far the best example of multimedia integration that is in wide use today is the WWW protocol used by the Internet.

The thrust to perform research in the area of 'monitoring' of CBT came from the massive connectivity potential the Internet provided in the area of large scale distributed training. However, the proliferation of the Internet, and popularity of intranets and extranets has led to different strategies being used in the development and integration of multimedia content. Three years ago the WWW was not capable of much interaction, it was more of a presentational method for text and graphics incorporating poor interaction. In addition, there were few high level tools available to enable non-programmers to author WWW content. The only way to produce CBT with acceptable interaction was through the use of conventional authoring tools such as Authorware, Director and ToolBook. Unfortunately, at that time these authoring tools offered little in the way of connectivity, online delivery and exploitation of the Internet.

Quite recently, active web technology such as Java applets, Java beans, Java Script, Active X, Flash and XML have emerged and are becoming widely adopted. These technologies have radically increased the capabilities, and interaction, of web delivered content. In addition, the introduction of high-level, easy to use, web integration tools, such as Dreamweaver, has opened up web content design to non-programmers and perhaps even to people with little or no IT skills.

Authoring tool vendors have not stood still, and have introduced intranet/Internet connectivity to conventional authoring tools through the use of plugins and the ability to store media externally. This is demonstrated in Figure 58 where it can be seen that at the WWW functionality, and ease at which it can be authored, has increased over the past three years along with the connectivity and Internet awareness of conventional authoring tools.

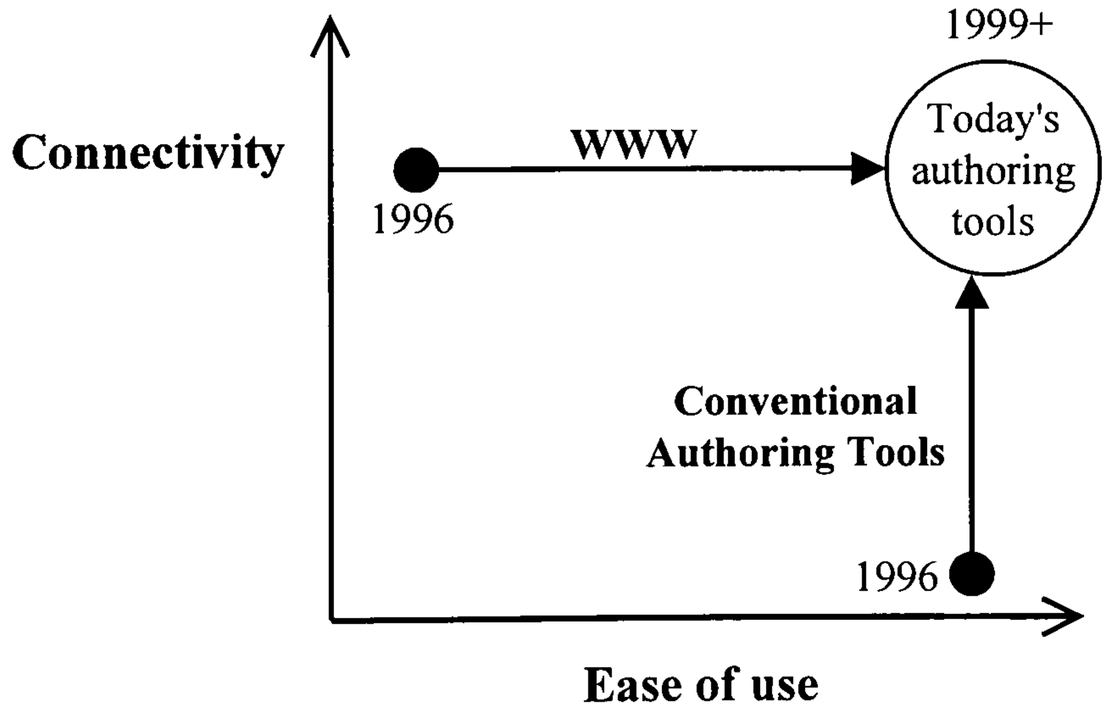


Figure 58: Evolution of authoring tools

This leads to the interesting question, of "What will happen next?" Currently more and more CBT is being deployed on company intranets, and the Internet. This suggests that as the WWW increases further in interactive potential and functionality, and reduces in

authoring difficulty, conventional authoring tools may be made redundant. However, experience suggests, obtained through analysing the changes in Multimedia and web technology over the last three years, it is extremely difficult to predict precisely what this emerging melting pot of hotchpotch technology will mature into.

7.7 Comparative products analysis

At the time this research project was conceived, a thorough search was performed for comparative software and nothing was found to be similar to the TIS designed here. The proprietary bespoke monitoring systems that were available, at that time, did not come close to delivering the flexibility, re-usability and connectivity that the TIS developed in this project has provided. However, various programs that could be thought of as comparative have since been developed, or older systems updated to be so, and they shall now be considered.

There appear to be two categories of these products, one class of products designed, in the most, by universities (WebCT, Topclass, Blackboard) to help lecturers to display teaching material on the web and perform assessment, and another class of products developed by authoring tool vendors (Pathware attain, Asymetrix Librarian) that provide a wide range of facilities leaning towards Computer Managed Instruction (CMI). Many of these products also integrate several Internet technologies to allow collaboration via e-mail or discussion lists between lecturers and students. The comparative products available commercially are listed in Table 12 with their web addresses.

Product Name	Website
Blackboard	http://www.blackboardllc.com/
Pathware Attain	http://www.macromedia.com/support/pathware/
Toolbook Librarian	http://www.asymetrix.com/products/librarian/
WebCT	http://www.webct.com/webct/intro/
TopClass	http://www.wbtsystems.com/
Lotus Learning Space	http://www2.lotus.com/
Question Mark	http://www.qmark.com

Table 12: Comparative products to generic TIS

The 'web tutoring' systems, developed mostly by universities, are different from the generic TIS in that they are aimed solely at Computer Aided Assessment (CAA) and web authoring. They provide a proprietary interface which must be used in order to construct courseware that uses their reporting system. Authors wishing to use these assessment/reporting tools will not be able to deploy their training over a LAN and are tied into using the specific authoring tool that the product provides. Similarly, Asymetrix Toolbook Librarian is designed specifically for deployment over the intranet/Internet and specifically for use with their own authoring tools (ToolBook Instructor and ToolBook Assistant). CBT applications developed without Asymetrix authoring tools must use be hot-wired into the Librarian system using Asymetrix's proprietary, complicated, OLX protocol.

Macromedia Pathware Attain is probably the product most similar to the generic TIS. Reporting facilities in Pathware are integrated and hooked directly into Macromedia's authoring tool, Authorware, but can still be accessed through the use of a Dynamic Link Library (DLL) by applications developed using alternative authoring tools. Pathware, like many of these tools, go further by providing collaboration, notice-boards and course management/scheduling utilities.

The main advantage of the TIS developed in this project is that it is solely a monitoring system and has many strengths and features in this area which the other programs lack. For example, the generic TIS system is not tied to any specific Authoring tool. This was a strict requirement by the collaborating company, CBL Technology Ltd, due to the wide variety of tools they used to author their training. Most CBT developers do not wish to put all their eggs in one basket by tying themselves down to using a specific development tool, and certainly not because of the monitoring system they decide to adopt. In addition, the comparative systems all dictate a tight specification of how interactions should be reported in the training environment.

In contrast, the generic TIS allows the author to report any variable he/she wants and to group data into structures that best suit that specific training application. This affords more flexibility during authoring and increases systems suitability for a wide variety of monitoring applications, from Health and Safety training to flight simulation. The generic TIS can use either the LAN or the Internet to transport information between the student and the administrative site, and courseware can be deployed on an ordinary LAN or over the Internet. This is possible because all courseware vendors now have mechanism to place their material on the Internet (Figure 58: Evolution of authoring tools). The generic TIS is a flexible monitoring system, going beyond providing just assessment recording, allowing authors to monitor every user interaction/visit/click to a very fine detail, in any multimedia or training application.

The popularity and proliferation of CBT and the advent of distance learning has led to organisations being established to set standards on how to create interoperable content and courseware management systems. The two largest players are the Aviation Industry Computer-Based Training Committee (AICC) that is favoured by industry, and IMS, favoured by academia.

AICC is an international association of technology-based training professionals that have developed guidelines for aviation industry in the development, delivery, and evaluation of CBT and related training technologies. This system has not only been adopted by the aviation industry but also by many other CBT content providers and users across industry. AICC guidelines and recommendations no. 6 and no. 10 [22] define the communication between TIS systems and CBT courseware for both LAN and Web-based systems. The latest versions of some authoring tools and comparative courseware monitoring systems are beginning to provide some AICC compatibility (namely Macromedia with Authorware and Pathware), and it could be that these standards will one day become widely accepted by industry.

IMS (formally Instructional Management Systems project, but now just IMS) originated in higher education and provides similar standards for Web-based courseware. The development of these sort of specifications are of supreme importance to distributed learning, allowing interoperability of compatible courseware between different course management systems and TIS's.

8 Conclusion

Teaching employees efficiently and effectively is a major concern for any company that wants to stay competitive. On a broader scale, maintaining a highly skilled workforce has the greatest influence on the competitiveness of the United Kingdom. The greatest asset of a company, or in fact this country, is the expertise and skill of its employees. However, rapid technological change not only means that the number of courses that employees have to undergo is increasing but also their sophistication and complexity. If technology is the cause of the problem by increasing training requirement, then the logical solution is to apply this technology to the training.

Companies have, for some time, been looking towards using Computer Based Training (CBT) as an alternative to traditional 'classroom' instruction, in order to increase effectiveness and reduce expenditure. Implementing a CBT solution has additional advantages by allowing employees to train when and where they want. This factor is especially important when considering the structure of today's companies who have offices all over the world, with many people using hot-desks or working from home. The sophistication of today's CBT applications have been pushed by industry's complex and diverse requirements and pulled by new technology founded by the multimedia revolution.

Alongside the multimedia revolution has been the massive increase in the use of, and proliferation of, computer networks. This connectivity, combined with the increased use of CBT, has provided massive potential for tracking and analysing training usage and monitoring student performance. Yet, the production of CBT, and in fact any multimedia application, is an extremely diverse procedure and is completed by people (authors) of

varying skill sets. This factor, combined with the huge variety of authoring tools, all with different Application Programmable Interfaces (APIs), created an urgent need for a single generic monitoring solution. However, these same factors were also the major stumbling blocks that had prevented such a solution, up until now, from being developed. These key problems, amongst several others, needed to be analysed before such a system could be designed, not least the balancing of contradicting user needs factors: users having different needs according to their perspectives as training systems managers, authors/developers, students and IT managers.

This research has done just that, and addressed many of the complex issues to do with monitoring of courseware over today's computer networks. As a direct result, many of the hurdles that companies encountered when attempting to incorporate monitoring into their CBT applications, or in fact any other multimedia application, have now been overcome. This research has made it possible for authors, with no additional skills beyond that needed to master their high level GUI authoring tools that they are already familiar with, to seamlessly incorporate monitoring functionality into their applications. Further, training managers can now track CBT modules which could have been delivered anywhere in the world and have been created with very different authoring / programming tools, by using a single system.

Although instructional multimedia applications have been around for over a decade, the area of today's interactive CBT applications, and the development processes used to create them, is still in a very immature state. With the aims of the research focussed squarely on developing a generic monitoring methodology specifically tailored to this complex

environment, we deemed it extremely important to examine this area thoroughly. In response to this, we began by investigating the processes used by multimedia developers, when creating CBT applications, and identified the current methodologies, techniques and tools that are used. This analysis demonstrated that there are a prolific number of tools used in the generation of CBT applications, and that these tools, and the people who use them, are quite different from those found in the conventional programming environment. CBT applications are now rarely programmed, but rather authored using an extremely high level programming language, or authoring tool.

This shift from programming to authoring has occurred because the best people to develop training are inevitably subject matter experts, rather than programmers. This raised two vital issues which a CBT monitoring methodology needed to address. Firstly, to be widely adopted, any monitoring system must be able to interface to the wide range of tools that are used to create today's CBT applications. The collaborative company considered this as vital, as many CBT developers use several tools due to the fact that some are better than others at doing different things. In addition, we would be exposing company to high commercial risk if the monitoring system restricted their choice of development tools to that of a single vendor. Secondly, different authoring systems provide alternative analogies and offer dissimilar programming processes that must be followed in order to build a CBT application. It is therefore necessary for the monitoring system interface directly to these analogies and thus require little additional authoring skill to utilise.

Many different ways were explored to enable CBT authors to integrate monitoring functionality into their applications using no additional technical or programming skills

beyond the authoring tool that they are already familiar with and would have previously been obliged to master. To test the feasibility of this idea, a Windows Dynamic Link Library (DLL) was written in order to extend the functionality of authoring tools to include monitoring capabilities. Interfaces were then built, for each of the authoring analogies, in order to communicate with that DLL. These interfaces were tested in the most popular authoring development environments in use today; Authorware, Director, ToolBook Instructor and ToolBook Assistant.

It was concluded that the interface with the monitoring system (the DLL combined with the authoring analogy interface) could indeed provide a single interface to all courseware irrespective of the development method. This type of interface does not deviate from the general theme of all authoring tools: Allowing authors, with limited programming skills, to develop their applications completely detached from the underlying technology. Another key aspect of the CBT environment are the requirements of the training manager.

The training manager's perspective is quite different from that of the author: Authors being concerned at how easily the system can be integrated into their productions, and training managers being more concerned on overall functionality and return on investment. CBT is ever more popular and is being applied to many more areas of training than ever before. Each of these areas can have very different and individual monitoring requirements. It was therefore of great importance, from the training manager's perspective, that the system could meet all the requirements of today's, and future, CBT applications.

If the system forced a rigid pre-defined structure of what can be reported there could be limitations as to its use. As a consequence, we explored different techniques to fully monitor any user interaction/visit/click to a detail defined by the author, rather than the system itself. Employing this tactic meant that we had to design a system able to cope with any data structure that a CBT application could possible throw at it. However, this flexibility would ensure compatibility with all future CBT enterprises and also ease the procedure required to integrate existing, legacy, courseware.

An equally important issue for the training manager is the ability to analyse results and to actually maintain the system. The training manager needs to be able to ascertain who has trained in what, how well students are performing and how well courses are being received by the students on the whole. To facilitate this, a monitoring system must provide a way of retrieving the individual data, describing each training session, from the distributed training environment, collate that data into a single entity and then provide tools to analyse and visualise that data.

To incorporate this functionality into our prototype monitoring system, an application was specifically written that took the numerous data records created by individual training sessions and collated them into one large database that could be queried by the training manager. This application used Open Database Connectivity (ODBC) in order that it may write the data in any popular database format. This allowed the training manager to select their preference of which Database Management System (DMS) they wished to store, analyse and view the results with.

All modern day DMS' provide a rich array of tools and wizards that make it extremely easy to visualise data such as that contained in the training database. The concept behind using an existing DMS to visualise the data, opposed to using specifically developed proprietary software, is inline with keeping the monitoring system as generic and flexible as possible. Similarly, File Transfer Protocol (FTP) was selected as the way to transfer data over computer networks.

FTP was chosen because it is an efficient and robust way of transferring data that is widely known about and readily accepted by customers. We saw this as important way to satisfy the needs of the IT manager. An IT manager, who looks after the day to day running of computers and computer networks, views any system that they have to integrate into their services as a potential threat. For example, if we had developed a highly sophisticated way to report data over networks, tailor made specifically for CBT applications, that used proprietary, previously unknown protocols the IT manager would be extremely hesitant at allowing the monitoring system to reside on his/her computer systems. We decided, upon investigating network technology, that any system would be far more readily accepted if accepted standards, such as FTP and ODBC, were used wherever possible.

In addition, many network security devices that shield company networks from the outside, such as Firewalls, actively disallow traffic that does not follow standard protocols. This would make any system that broke the rule extremely tedious to install: Each individual firewall having to be separately configured to accept the non-standard traffic. Another issue that we addressed was transparency to the student. Monitoring systems should enhance the students experience rather than detract from it. A badly designed monitoring

system is one that distracts the user from their training. For example, it is quite possible for an external system, such as a monitoring system, that is linked to the CBT application could slow down or even halt that application.

In this research we investigated several anticipated distractions and came up with solutions for them that are demonstrated in the prototype. For example, in the prototype, login scripts were avoided, where possible, by automatically identifying students through their IP address or from a previous network login. Also, if an event prevents data being transmitted over the network, the data is held locally and transmitted at the next opportunity. Investigations were also carried out to determine the optimum file format in order to reduce transmission time, and hence interruption experienced by the student. Ideally the students should only be aware that their activities are being monitored at the discretion of the training manager or CBT authors.

The prototype system was subjected to three independent trials: Firstly at the collaborating company, CBL technology, that in turn led to larger trials at both Cable and Wireless Communications PLC and the University of Huddersfield. These trials gave us the opportunity to examine the performance of the prototype in relation to the customer (training manger) requirements, the CBT developer (author) requirements and the requirements of the students who received the training. The positive feedback from these trials have demonstrated that we have indeed addressed the major aspects of monitoring CBT in today's society.

Our approach has been to investigate all aspects of, and roles associated with, computer based distance learning and the monitoring of material over today's computer networks. This allowed us to define, and empirically test, a monitoring methodology that could meet the requirements of this complex environment. The area of CBT, and especially that of monitoring of CBT, is still a very immature and rapidly changing environment. It is, therefore, of prime importance that CBT monitoring systems apply the methodologies that have evolved during this research: facilitation of monitoring integration by non-programming authors, the freedom to select any authoring tool and the ability for modules developed with different tools to co-exist within a single environment.

This research has demonstrated, with the production of a prototype TIS, that a generic flexible monitoring solution can be applied to this rapidly changing, diverse, environment. As companies are increasingly burdened by legislation, the monitoring and tracking of training is becoming a more important part any training solution. The ability to record this type of training information is also essential to prove that the money allocated to a training budget is being used in the most effective manner.

With the increasing adoption of CBT solutions, the proliferation of networked computers and use of distance learning, the factors researched by this project are clearly going to play a key and co-ordinating role in the enhancing the effectiveness and administration of training. This research has shown traditional CBT can be much improved by exploiting computer network technology to monitor student activities, and that a generic CBT monitoring methodology, addressing the complex needs of the CBT industry, can indeed be achieved.

9 Further Work

The fundamental objective of the monitoring system was to retrieve data from CBT applications into a master database for analysis. The generic monitoring system achieved this by writing to a local file, transmitting it over the intranet/Internet, and employing a separate application (the collator) to then merge the data into a database. Since this project was started there have been some major innovations which provide more efficient alternatives to the steps taken by the generic monitoring system we developed. This chapter examines some of these innovations and discusses different ways of enhancing the system.

9.1 Internet Database Technologies

Both the reporting system and the collator employed ODBC to read and write databases. There are now substitute technologies, such as JDBC (Java Database Connectivity) and Microsoft Active Server, that facilitate the manipulation of server-side databases over the intranet/Internet. If technology like this was implemented the delivery mechanism and collation steps would no longer be necessary. Furthermore, it would be unnecessary for any data to be written to the local machine. Figure 59 shows such a system employing Microsoft Active Server to write directly to the master database. The reporting interface would be the same, but the reporting DLL would send SQL commands over the internet for processing by the remote Active Server instead of using ODBC to produce a local database.

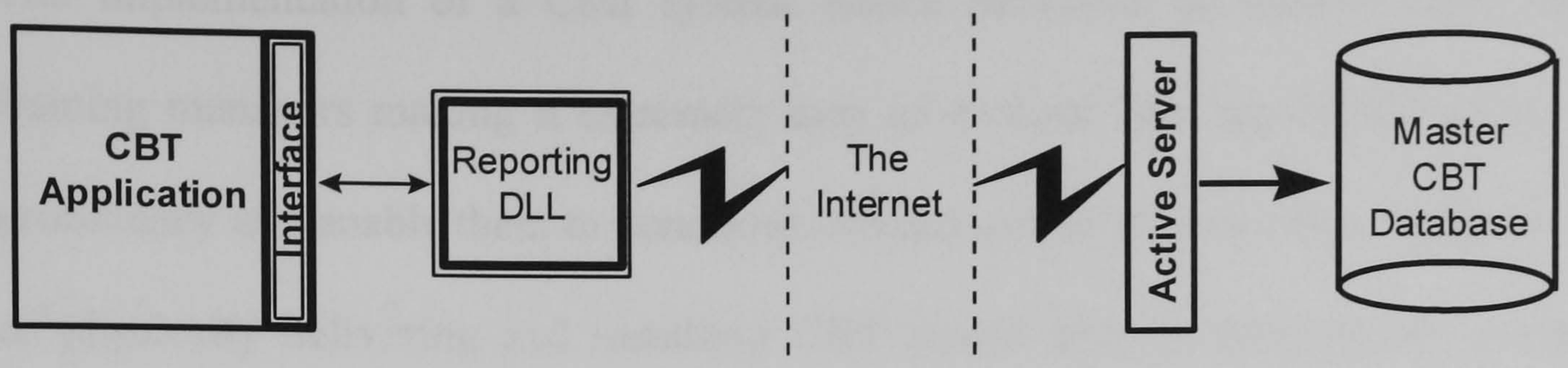


Figure 59: Reporting using Active Server

An additional advantage of this improved system is that online queries can be performed on the remote database allowing immediate feedback for both the students themselves and administrators. Currently, technologies such as Active Server and Java are being used increasingly in the design of CBT systems deployed on the Internet [35]

9.2 Computer Managed Instruction

The technical advantages of CBT goes far beyond that of facilitating monitoring. The generic monitoring system developed as a result of this research only partially exploits what is achievable. The most complete way of exploiting CBT would be to encapsulate the monitoring system as part of a Computer Managed Instruction system. A CMI system would provide a complete training environment, an environment within which both courses and student profiles could be manipulated, providing facilities for the deployment and management of CBT. The objective of a CMI system would be to organise curricula, assign the curricula to students, route students through courseware components of the curricula, and track what the students do in them. A CMI system would reduce staff downtime by allowing geographically separated staff to train when and where they wanted in a accountable, controlled environment.

The implementation of a CMI system would introduce invaluable rapid feedback to training managers making it extremely easy to evaluate training effectiveness, employee proficiency and enable them to hunt down subject matter experts when required. The costs of physically delivering and installing CBT would also be dramatically reduced. The functionality of a CMI system can be broken down into 6 distinct sequential parts which are shown in Figure 60. A CMI system would contain an application or tool (a course builder) to assemble courses and activities into curricula. The system would also facilitate the enrolment of students to this curricula and handle the delivery, monitoring, scheduling of courses and the production of end reports.

CMI is currently being exploited by all the major software vendors of authoring tools and standards such as AICC and IMS are specifying how to design interoperable courseware and CMI systems. A key part of a monitoring system, and that of a CMI system is the reporting of performance and the visualisation of the results.

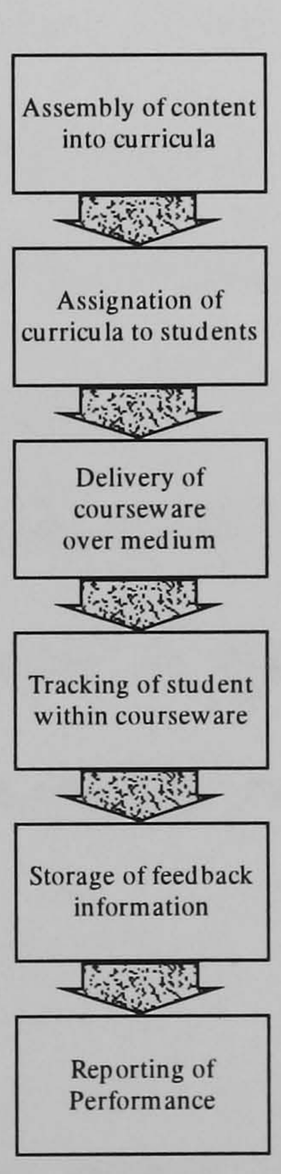


Figure 60: Functions of a CMI system

9.3 Pay Per View

Although training managers and companies are looking increasingly towards using CBT to fulfil their training needs there are still some negative aspects of a CBT solution that give some companies concern. One of these is that CBT, if it is to be effective, is expensive to develop. For example, it may take 300 man hours to produce a single hour of user training. Obviously, this cost is very much offset if the training is applied to large numbers of people. However, to some companies, the high development costs for bespoke CBT applications could make the solution prohibitively expensive. This is especially the case if a company has not previously implemented a CBT solution and, as a consequence, is more sceptical of CBT and hence perceives the expenditure as a risk.

Monitoring systems, such as the generic TIS developed here, will play an important part in addressing this problem. It is possible, through the use of monitoring systems and networked CBT, to levy a small charge on people as they view the training. This would be a much easier 'pill' for prospective, especially smaller, CBT customers to swallow than the high costs involved in an outright purchase. The TIS that has been developed as a result of this research project, can track exactly what a user has seen, and record how long they have spent viewing individual pages and could, in theory, facilitate a pay-per-view system. However, further work is required on security and the system, as it currently stands, is open to abuse by the user.

For example, the Windows operating system multitasking environment provides an opportunity to alter the contents of a session database prior to it being sent. In addition,

there is no guarantee that the information has not been intercepted and altered during the FTP delivery process. This issue is a current topic being discussed by the WWW consortium, in order to define a standard way to allow secure e-commerce over the WWW. There are currently many encryption / authentication methodologies available that make the transfer of private data over public computer networks extremely safe. One of these is Pretty Good Privacy (PGP) which uses a distributed public key method. In addition, there are many essential steps to be taken to integrate any software application into an e-commerce environment [34, page 1291]. At the very least the TIS developed by this project should incorporate an encryption algorithm, such as PGP, prior to being used as a pay-per-view system.

9.4 Advanced Data Visualisation Techniques

As technology advances and CBT and distance learning become more widely accepted the number of courses and students that have to be introduced to a TIS, or CMI system, will inevitable increase. It is therefore a challenge to introduce a way to display the vast amount of information contained in training databases in a simple, innovative, and easy to visualise manner, for perusal by the training manager. Humans are able to analyse information and make informed decisions from it, but lack the ability to absorb, retain, and process, the vast amounts of ever increasing data that can be created by a TIS. Such a task is more suited to a computer. The problem of a person reacting to this data is therefore one of communicating with such a computer, and the algorithms employed to summarise the data[23].

Virtual-reality is perhaps the solution in providing a way to manipulate and visualise vast quantities of data using intuitive man-machine interfaces. As a consequence, experiments have been conducted, by this project, at implementing Virtual Reality to facilitate visualisation of data and relationships between data elements. A separate C++ application was developed in order to investigate the practicalities of this research area. The application queried the training database and created a virtual world to represent the data by automatically writing Virtual Reality Modelling Language (VRML) code.

Experiments were then conducted by exploring the use of colour, columns and circles to represent data. It was found that if the training manager orientated themselves correctly, they were able to absorb large amounts of information by performing a 'fly-by' over the field of columns. Also, if the training manager flies over the terrain in one direction individual student performance can be assessed, and if the training manager flies over the terrain in the other direction, overall course suitability, and acceptance by the students, can be assessed. Figure 61 demonstrates how data contained in the master database produced during the TIS validation (section 6.3) can be represented in a virtual world.

This example was created by applying the C++ VRML builder application to the same master database that was used to create the chart shown in Figure 48. However, this visualisation technique displays all the student results as spheres. Student activities and responses to all questions have again been summarised into 5 autonomous chapters: with each chapter represented by a sphere instead of a circle. Each student has 5 spheres that together represent his/her total training activity within the courseware. The size of a sphere represents the amount of content/lecture-material that student viewed in that

particular chapter. The colour shows how well the student has performed in questions linked with that particular chapter by varying between red (all incorrect) and green (all correct).

This technique allows the tutor to instantly visualise individual performance, or more generally, chapter suitability, for the whole of the class, at a quick glance. This example demonstrates how virtual reality can be used to visualise very large amounts of data extremely quickly. However, at this stage, the virtual environment has not been exploited to its full potential. The work completed so far is in early stages and does not benefit fully from using the interactivity that determines what is considered by some people as the only true Virtual Reality.

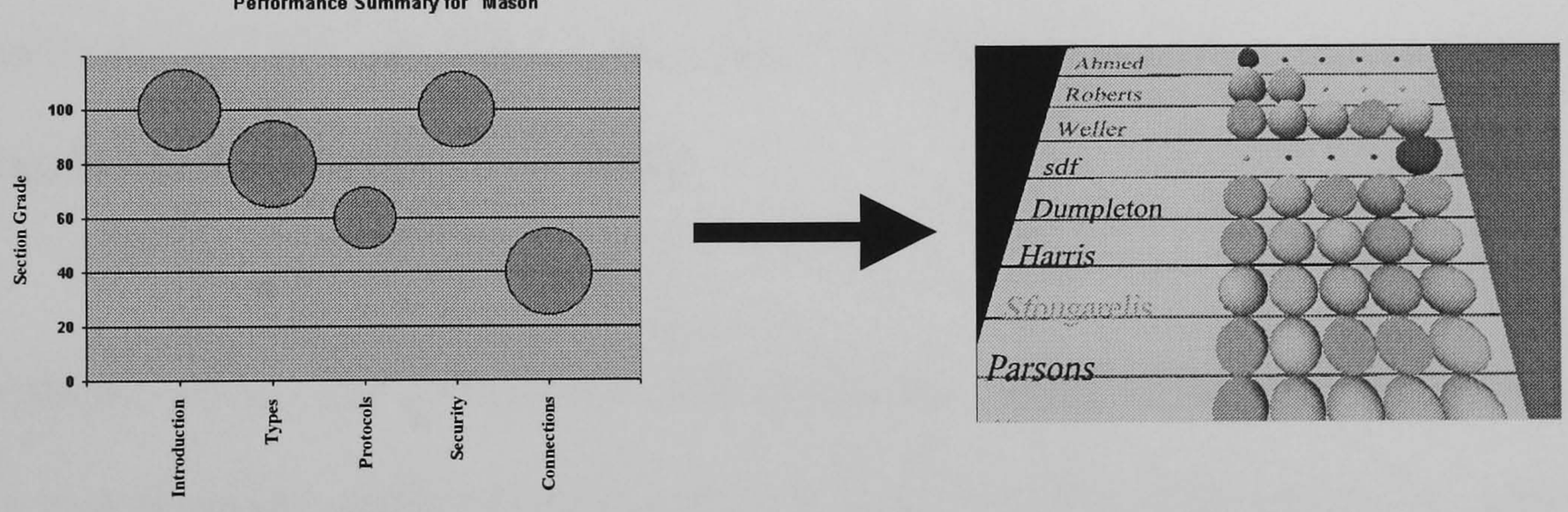


Figure 61: Representing data using a Virtual World

VRML gives the opportunity to use attributes which are far more palatable to the human brain than conventional methods of data visualisation. Further research will need to be undertaken into the use of spatial awareness, time, shapes, and sounds to enhance the understanding of complex data structures. In addition, interactivity between the training manager and the objects in the virtual worlds must also be explored. The use of Virtual

Reality will allow training managers to instantly evaluate training effectiveness, employee proficiency and enable them to hunt down and contact subject matter experts when required. Using virtual reality to visualise data is not new, but little work has so far been done in integrating virtual reality with a TIS.

9.5 Collaborative Environments

The thrust of this research has been to enhance CBT by facilitating monitoring of student and courseware performance over computer networks. This led to several investigations and the development of a highly flexible TIS. The TIS provides a way to transfer information about the use of a CBT application to a training manager. However, it does not allow the transfer of information between one CBT application and another. Alongside the advantages of CBT and distance learning, is the disadvantage that the student is likely to endure an isolated learning experience.

Peer communication and group work is an important aspect of many learning processes and is a tool regularly used by lecturers to enhance conventional instruction. In addition, a collaborative learning environment allows students to exert more control over their own learning [36]. Today's CBT applications have not yet addressed this shortfall that impinges on the students learning experience. For example a very successful and important CBT application designed by CBL Technology Ltd, the collaborating company, is one which simulated crowd control and is used in the training of police officers. At the moment this is a standalone application, but with networked collaboration it would be possible to simulate a number of different physical locations such that a crowd control

action taken by an officer at one location affected what happened at another location as the crowd moves around. In this way teams of officers could simulate real situations in real-time under the control of a master station which could trigger certain events to happen.

The field of collaborative computing encompasses the use of computers to support co-ordination and co-operation of two or more people who attempt to perform a task or solve a problem together [24]. There are third party software applications such as e-mail, video conferencing and Microsoft's NetMeeting that facilitate collaboration over computer networks. However, these tools are either too distracting, not interactive enough and not tailored for seamless integration into a training environment. A significant problem encountered when integrating outside programs with a CBT application is often the limited programming expertise of the author. However, the majority of these problems have been overcome through this project and the development of the generic TIS.

As a consequence, the existing TIS could be designed to allow collaboration between students and the majority of CBT developers would be able to use this new tool to provide both monitoring and collaboration, vastly improving the students perceived and unconscious learning experience. This will have a direct positive impact on the quality of tuition that CBT can supply.

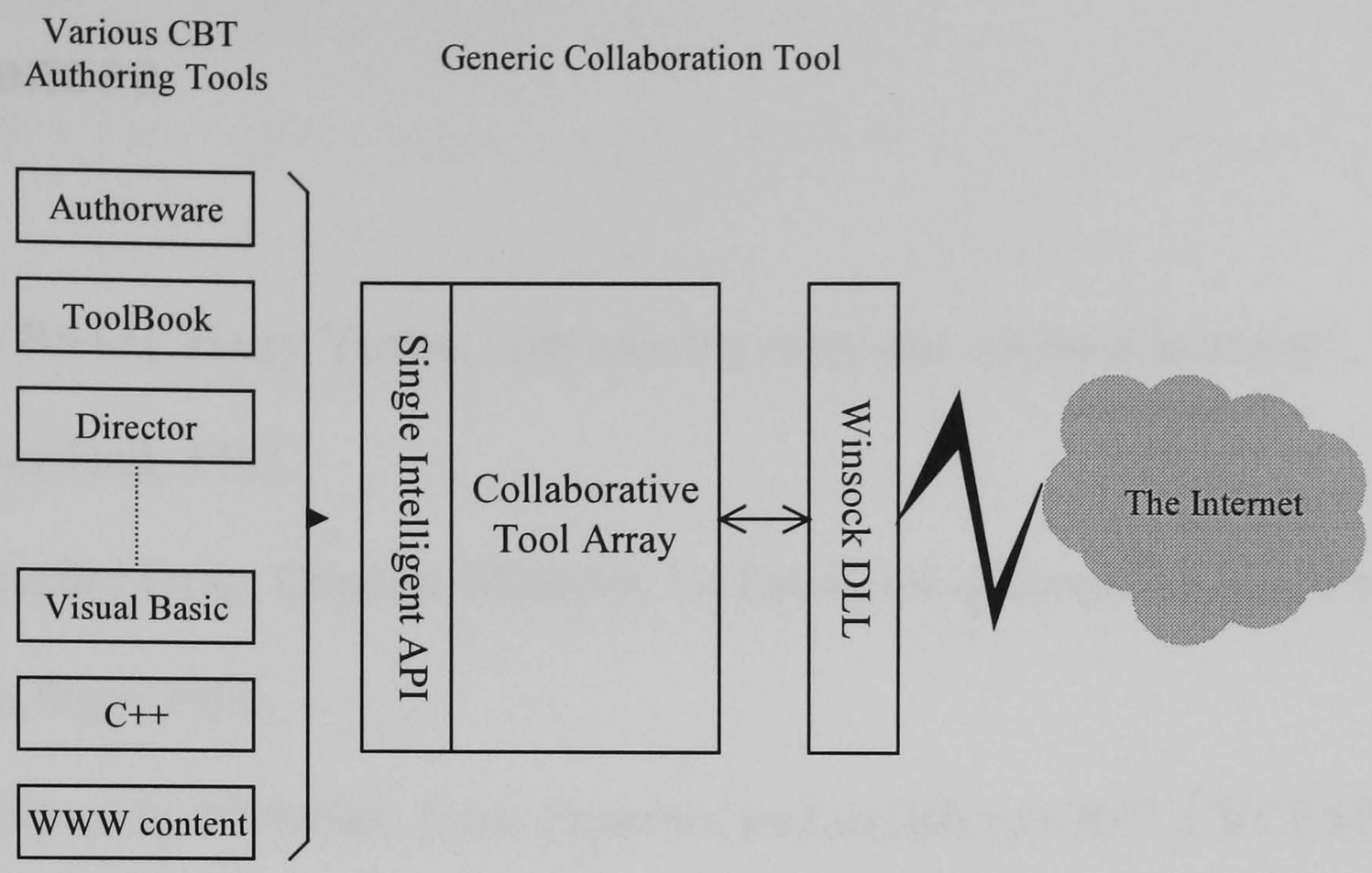


Figure 62: Overview of Collaborative Tool Array

There are also some proprietary systems that exist that incorporate purpose built high-speed communication between bespoke training applications. However, these systems are bespoke hardware developments and do not address the market requirement for a generic communication tool that can be integrated, by someone with limited programming experience, into the large variety of authoring tools now used to develop multimedia and CBT applications.

10 References

[1] Philip Barker, Harry Yeates, "*Introducing computer assisted learning*", Prentice/Hall, 1985.

[2] Christopher Dean, Quentin Whitlock, "*A handbook of computer based training*", Kogan Page, 1992.

[3] J. D. Edwards, "*Benefits, Uses, Potential and pitfalls of CBT*", CVCP/USDU, February 1993.

[4] AICC CMI Subcommitte, "*CMI Guidlines for Interoperability AICC*", document no. CMI001, 01/07/1997

[5] William Buchanan, "*Mastering the Internet*", Macmillan Press Ltd, 1997

[6] Http://www.nw.com/zone/WWW/top.html, "*Network Wizards Internet Domain Survey*"

[7] CBL Technology Ltd, "*Display Screen Equipment Health and Safety Training CBT*"

[8] CBL Technology Ltd, "*Integrated Customer Management System CBT*"

[9] Solis Design, http://www.solis.com/wpstd.html, "*Standards Based Computer Managed Instruction: Impetus for Change*"

[10] Thomas McLaren, CBT Solutions, "*1997 Authoring Tool Roundup*", http://www.cbtsolutions.com/html/9709_mcl.htm

[11] Don Binns, "*Inside Windows NT*", Microsoft Developer Network Journal, July 1997

[12] Charles Petzold, "*Programming Windows 3.1*", Microsoft Press, 3rd Edition, 1992

[13] Microsoft, "Developer Network", Microsoft Developer Network CD-ROM, October

1996.

[14] Alan R. Feuer, "*MFC Programming*", Addison Wesley Developers press, 1997

[15] Erik Holsinger, *"How Multimedia Works"*, Ziff-Davis Press, 1994

[16] Ron White, *"How Computers Work"*, Ziff-Davis Press, 1997

[17] CBT Solutions magazine, *"1997 CBT Report"*, SB Communications, 1997

[18] IDC Research, http://netware.novell.com/discover/compete/mswinnt/topten.htm, "*NOS Market Share 1997*"

[19] Association of Test Publishers, "*Questions about Tests*", http://www.testpublishers.org/question.htm

[20] Wash Redmond, "*New Strategy Solves Needs for Media Streaming, Media Authoring*", Microsoft Corporation (PressPass), 1998

[21] Robin Chater, "*Uses and misuses of personal data in employee/employer relationships*", http://www.open.gov.uk/ppru.htm, 1999

[22] AICC CMI Subcommittee, "*CMI Guidelines for Interoperability AICC*", July 1997

[23] Jason Watson, David Taylor, Stephen Lockwood, "*Visualisation of Data from an Intranet Training Systems using Virtual Reality Modelling Language (VRML)*", Proceedings of the International Conference on Information Visualisation, 1999

[24] N.S. Borenstein, "*Computational Mail as Network Infrastructure for Computer-Supported Cooperative Work*", Proceedings of the ACM Conference on CSCW, Toronto, Canada, 1992

[25] "*Novell and Microsoft Skills go online at BT*", Computer Weekly, 27 May 1999, p49

[26] John Villamil-Casanova and Louis Molina, *"Multimedia: An Introduction"*, Macmillan Computer Publishing, 1997

[27] Muriel Jourdan, Nabil Layaida and Cecile Roisin, *"Authoring Techniques for Temporal Scenarios of Multimedia Documents"*, published in the "Handbook of Internet and Multimedia Systems and Applications", IEEE Press, 1998

[28] Robert S. Tannenbaum, *"Theoretical Foundations of Multimedia"*,W.H. Freeman and Company, 1998

[29] Francis Botto, "*Dictionary of Multimedia and Internet Applications: a guide for developers and users*", John Wiley & Sons, 1999

[30] Jessica Keyes, "*The Ultimate Multimedia Handbook*", McGraw-Hill, 1997

[31] Liz Barnett, David Brunner, Pat Maier, Adam Warren, "*Technology in Teaching & Learning, a guide for academics*", Interactive Learning Centre, University of Southampton, 1996

[32] Adrian Boucher, "*Who pays the costs? Who gets the benefits?*", Conference Proceedings: Embedding Technology into Teaching, CTISS Publications, 1996

[33] Marc H. Brown, Marc A. Najork, "*Distributed Active Objects*", Computer Networks and ISDN Systems 28, 1996

[34] Paul-Andre Pays, Fabrice de Comarmond, "*An Intermediation on Payment System Technology*" , Computer Networks and ISDN Systems 28, 1996

[35] Jay M. Lightfoot, "*A blueprint for using the World Wide Web as an Interactive Teaching Tool*", Journal of Education Technology Systems, vol. 27(4) 325-335, 1998-99

[36] D. Boud, "*Assessment and the Promotion of Academic Values*", Studies in Higher Education, pp. 101-111,1990

[37] D.K. Lidtke and D. Moursund, "*Computers in Schools: past, present, and how we can change the future*", Communications of the ACM, pp 84-87, May 1993

[38] Andrew S. Tanenbaum, "*Computer Networks*", Prentice-Hall inc., 1996

[39] J. Watson, D. Knowles, P. Smith, D. Taylor, "*Delivery, Scheduling and Monitoring of Computer-Based Learning Material Over Computer Networks*", Computer Based Learning in Science International Conference, 4th July - 8th July 1997, De Montfort University, Leicester, UK.

[40] J. Watson, D. Knowles, P. Smith, D. Taylor, "*Designing Next Generation Computer-Based Training Products Which Exploit the Internet*", Teaching Company Directorate Multimedia Seminar, 15th October 1997, University College of North Wales, Bangor.

11 Appendices

11.1 Appendix A - Computer Based Learning in Science International Conference, 4th July – 8th July 1997, De Montfort University, Leicester, UK.

Delivery, Scheduling and Monitoring of Computer-Based Learning Material Over Computer Networks

J. Watson*, D. Knowles*, P. Smith*, D. Taylor**

ABSTRACT

Computer-Based Learning (CBL) and Computer-Based Training (CBT) applications have been around for many years. These applications are now responding to the advent of high-speeds global computer networks and changes in learning/working practices by offering CBL/CBT on demand over the Internet as well as sophisticated monitoring and scheduling functions.

KEY WORDS:

delivery, scheduling, monitoring

BACKGROUND

CBL Technology Ltd are a company working at the forefront of computer-based training (CBT) design and production. They have many customers in all areas of education, industry and commerce including the RAF, Rolls Royce, the Bank of England and are associated with a number of projects in the HEFCE Teaching and Learning Technology Programme (TLTP).

At CBL creative CBT designers use Authorware to develop CBT material which makes extensive use of multi-media, with typical products being tens or hundreds of megabytes in size. Currently the only way to ship any demonstrations, prototypes or finished products to customers is by writing custom CDROMs, which is time consuming and prohibitively expensive when dealing with large numbers of students. This also results in a stand-alone, insular product which tends to be used by one person at a time with no automatic facility for scheduling series of modules or recording of who has seen what.

Through this project, carried out in collaboration with the School of Engineering at the University of Huddersfield, CBL Technology are exploiting the rapid expansion and proliferation of computer network technology to develop more sophisticated next generation CBT products and delivery mechanisms. They are using the Internet to distribute products to customer contact points and shipping a Delivery Framework which will permit delivery to individual students over local computer networks. This framework also enables scheduling of CBT material at the correct time and recording of who has seen what material and what kind of progress was made.

This project is basically tackling two weak links in the product cycle which need to be strengthened if companies like CBL Technology are to improve turnaround times, cut development costs and provide new functions which customers are demanding.
The first weak link concerns the initial delivery of demonstration, prototype and completed material to the customer site, done traditionally by expensive, specially written CDROMs which cannot be edited once written, and of which one customer requires 1,500 copies. Most customers have their own computer networks (intranets), which are linked to the Internet and are capable of downloading material quickly from a CBL site or even running material in real-time over the network via Shockwave Internet browser extensions. Through this project CBL Technology are also utilising the Internet to communicate with customers sites across the world, eliminating expensive media copying and delivery costs by downloading material to customer sites over the Internet. Development times are also improved since prototype material can be delivered and assessed by the customer much more quickly and new/updated versions delivered almost immediately without needing to write new CDROMs.
The second weak link concerns delivery to the individual students at a customer site. Students may be spread over a number of different laboratories, at different sites with some "on the road" or working from home. Traditionally this would involve the purchase of a number of CDROMs and software licenses making CBT look prohibitively expensive and with no means of monitoring who has seen what. By exploiting customer computer networks (intranets) for delivery around sites and creating a new Delivery Framework this programme will be able to provide a number of new network-based CBT facilities; specifically CBT scheduling and student monitoring, which will make their products more attractive to both existing and new customers. Computer network technology and these types of facilities also open up the prospect of group working and collaborative CBT.

SOFTWARE SPECIFICATIONS

The additional software required to exploit these developments in computer network technology and to solve the problems described falls into three categories:

A Delivery Framework

The delivery framework facilitates delivery of courses to customer contact points and installation of the courses for access by all staff at customer sites. The delivery framework is accessed via the Internet and when installed it sets up the Management Information System (MIS) database structures and places the files directing the scheduling algorithms in the correct place

Courseware Monitoring

The monitoring facilities fall into three categories:

- Obtaining student identification details: Students have to logon to their company/university network before they can begin running courseware, and it is therefore unnecessary to ask them to register again with each piece of courseware used.
- Monitoring student progress and performance: In different situation specifications will require that all kinds of different information be monitored and reported. Some applications will be interested in navigation routes, in key responses to questions, or in response times or total execution times etc; the possible list is endless. The monitoring

functions therefore need to enable monitoring and reporting of any type of variable from within the courseware with a minimum of inconvenience to the author.

- Collation of data and MIS information: The data file created each time a student accesses a piece of courseware is only one piece in a much larger jigsaw puzzle. There is a requirement for a specially written collation utility to merge all files into one large database which will provide information to a training manager about the effectiveness of the courseware and who is having any problems.

Courseware Scheduling

The scheduling functions ensure that a student sees the correct modules at the correct times and in the correct sequence. These functions therefore provide a capability for a training manager to schedule courseware appropriately.

Scheduling may take place on a purely temporal basis, informing anyone who is shown in the database as not having completed the current module that a new module is imminent and that every effort should be made to complete the current module. Alternatively scheduling can take place on a pedagogical basis so that students can progress at their own pace and such that they are directed to skip modules or take a supplementary diet of material if the MIS system sees that they are either coasting through the material or struggling.

EXAMPLES OF USAGE

So far the project has developed courseware monitoring functions from the specifications outlined above. A creative CBT author can use the utilities to include additional facilities without an in-depth knowledge of how they actually work (figure 1)

Obtaining Student Details

A single icon, included from a provided monitoring library is used to identify personal student details. Currently the assumption is that the student has his/her own dedicated PC and that the network is an intranet with Internet protocol TCP/IP loaded. This icon therefore uses a function from the winsock dynamic link library (dll), integrated into a custom written dll, to obtain the IP address of the machine which is used as a unique student identifier, although this system may be enhanced later.

Monitoring and Reporting

The system implemented for monitoring user progress and performance is based on specially created Authorware Group icons which can be customised by authors. By inserting the icon into the normal program flow and renaming it accordingly the icon will either manipulate databases or write information about a variable related to the icon name to a database of progress and performance records. These functions have been implemented in this flexible and transparent manner so that an author can pass any information such as usage time, responses to questions, response times, navigation routes to the database. They utilise the Authorware Open Database Connectivity (ODBC) facilities with Standard Query Language (SQL) calls to a "text" database. One icon is used to initialise the monitoring database for the current session. If the database is present it is cleared or if this is the first session with this piece of CBT and there is no database

currently available a new one is created. A second icon is used at the end of the application to close the database and transmit it to a central collection point. This is currently done simply by writing the database file to the local network server, but later version are intended to be able to use the Internet to transmit these to any central collection points at either CBL or the company training headquarters.
Within the application itself the author can use a third icon to report information about any variable to the database.

Although most databases created will be quite complex, figure 2 shows a simple example record created by the monitoring utilities described above:

- The first field records the IP address of the student
- The second field is the time that the application was started
- The third field indicates that the student got a key question correct
- The fourth field indicates that the application was completed fully
- The fifth field is the time spent by the user on the application.

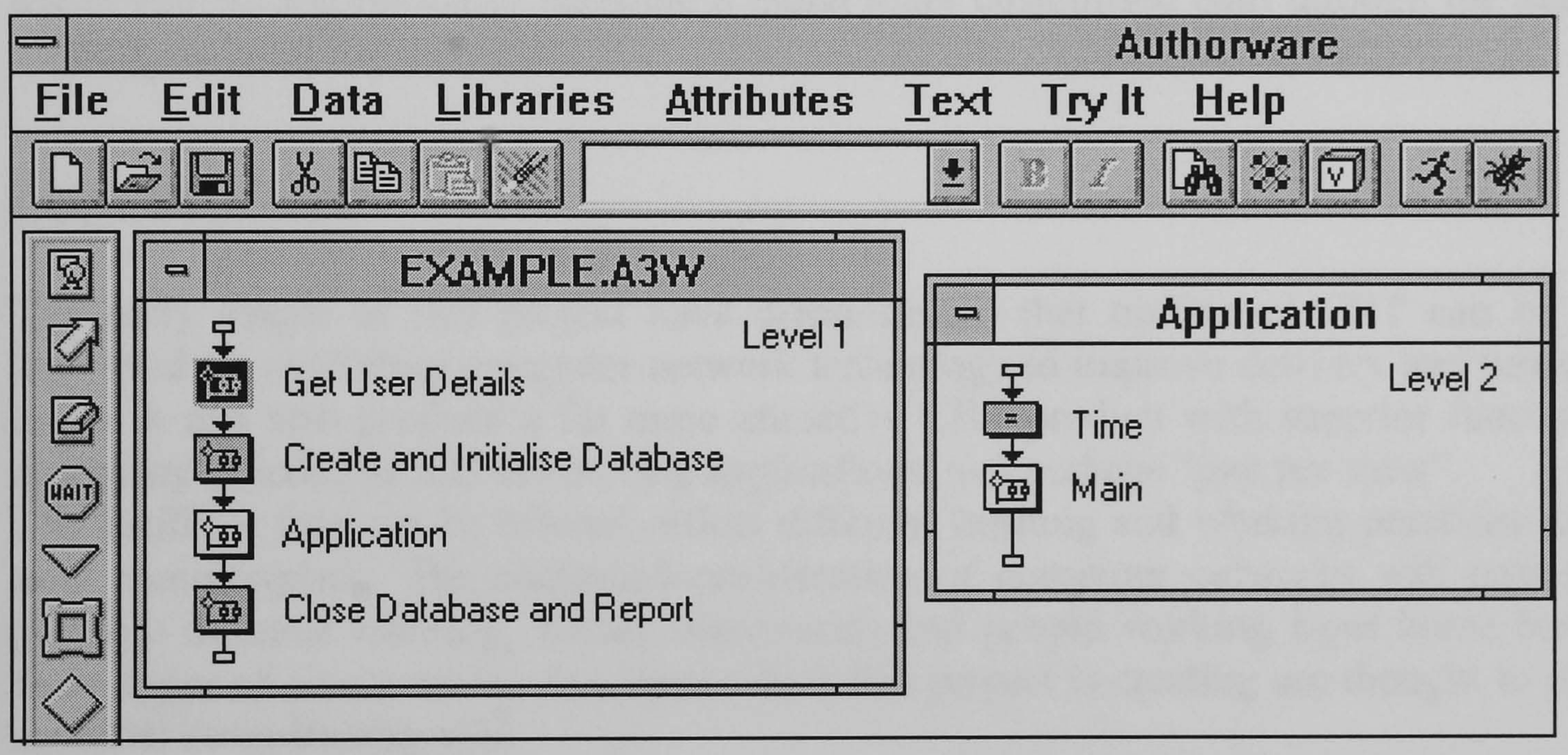


Figure 1. Example usage of monitoring functions

MIS

A management information system (MIS) to process all of the monitoring results is currently under development. The MIS firstly collates all of the individual databases which arrive at the collection point. It then unifies them and provides an interrogation facility, via an Authorware user interface, which allows a training manager or CBL to diagnose the information. This information can be used by CBL as feedback into the development cycle or by a training manager/tutor to assess the success of the training.
By collating number of databases similar to those in figure 2 a tutor/training manager could determine:

- What proportion of students have completed the module

- Whether anyone had any problems with the material
- Average training times
- Information as to when students find it convenient to use the material.

161.112.68.2	**12.37**	**Y**	**C**	**46**

Figure 2. Typical Database Record Created

Scheduling

The framework in which all these functions are delivered to the user includes simple scheduling functions which ensure that the correct module is seen by the correct user at the correct time.

Future development include the use of the training database to identify personal training needs and AI algorithms to schedule a much more customised path through the available training material for each user.

CONCLUSIONS

The early stages of this project have demonstrated that traditional CBT can be much improved by exploiting computer network technology to improve delivery and turnaround times. It can also produce a far more attractive CBT product with superior functionality and many scheduling and monitoring applications, and perhaps "pay per view".

The facilities that can be offered reflect different learning and working practices that are now commonplace. The continued proliferation of computer networks will continue to promote distance learning, virtual laboratories and people working from home bases. In these types of situations the functions which this project is creating are thought to play an essential co-ordinating role.

**Dr. David Taylor
Division of Electronic and Communications,
School of Engineering,
University of Huddersfield,
Queensgate,
Huddersfield,
HD1 3DH.
Email: d.taylor@hud.ac.uk

*J. Watson, *D. Knowles, *P. Smith,
CBL Technology,
St Katherines House,
Prime Enterprise Park,
Mansfield Road,
Derby,
DE1 3TQ.

Designing Next Generation Computer-Based Training Products Which Exploit the Internet

J. Watson*, D. Knowles*, P. Smith*, D. Taylor**

ABSTRACT

CBL Technology Ltd are a company working at the forefront of computer-based training (CBT) design and production. They have many customers in all areas of education, industry and commerce including the RAF, Rolls Royce, the Bank of England and are associated with a number of projects in the HEFCE Teaching and Learning Technology Programme (TLTP).

At CBL creative CBT designers use Authorware to develop CBT material which makes extensive use of multimedia, with typical products being tens or hundreds of megabytes in size. Currently the only way to ship any demonstrations, prototypes or finished products to customers is by writing custom CDROMs, which is time consuming and prohibitively expensive when dealing with large numbers of students. This also results in a stand-alone, insular product which tends to be used by one person at a time with no automatic facility for scheduling series of modules or recording of who has seen what.

Through this Teaching Company Programme, carried out in collaboration with the School of Engineering at the University of Huddersfield, CBL Technology are exploiting the rapid expansion and proliferation of computer network technology to develop more sophisticated next generation CBT products and delivery mechanisms. They are using the Internet to distribute products to customer contact points and shipping a Delivery Framework which will permit delivery to individual students over the Internet. This framework also enables scheduling of CBT material at the correct time and recording of who has seen what material and what kind of progress was made.

KEY WORDS:
CBT/Internet/monitoring

BACKGROUND

CBL Technology Ltd wish to exploit the rapid expansion and proliferation of computer network technology to develop more sophisticated next generation CBT products and delivery mechanism. Through this teaching company programme the company intend to use the Internet to distribute products to customers and to develop a Delivery Framework which will permit scheduling of CBT material at the correct time and recording of who has seen what material. In the future these facilities may also facilitate group working over customer computer network and a "pay per view" invoicing facilities.

Introducing CBL Technology Ltd

CBL Technology Ltd is the UK's largest bespoke computer based training and multimedia analysis, design and production company, currently employing 70 full-time development staff operating within an ISO 9001-registered quality system from offices in Derby and Bristol.

Formed in 1990, originally to produce Computer Based Training (CBT) material for Higher Education, CBL Technology then expanded steadily into the design and production of bespoke training courseware. This is an area in which they have gained both considerable experience and the respect of mainly blue-chip commercial, industrial and defence clients. Armed with this experience, they have diversified into other areas of multimedia and electronic communications for which their background has given them a sound footing for the analysis, design, media selection, production, project management and implementation issues involved.

The Company has experience and expertise in the following key areas which are directly relevant to multimedia development:

- Training needs analysis and design (including SAT and ISD)
- Software engineering (Windows, Mac, Novell & Unix; Visual Basic & C++, C, Delphi, Pascal)
- Software analysis and design (SSADM, SDM)
- Graphic design and production (3D Studio, Animator Studio, Illustrator, PhotoShop, Painter)
- Multimedia authoring (Authorware, Director, Rapid)
- Multimedia production (Premiere, MPEG, Video for Windows, Sound Forge, CD Mastering)
- Virtual Reality (SuperScape)
- Project management (PRINCE, STRADIS)

Advantages of Computer Based Training (CBT)

CBT has now been proved, over several years and in many diverse applications, to have a valid role in training. As companies become leaner and budget, particularly those allocated for training, become tighter, CBT is likely to be examined increasingly by trainers and managers as an alternative to more traditional methods of instruction. This factor, coupled with the decreasing costs of PC hard ware and software, both of which are advancing in power, capabilities and ease of use, will all serve to make CBT look increasingly attractive. In general, the advantages of using CBT are as follows (O'Neil 1981):

- Those which predominantly reduce costs by:
  - ♦ Reducing training time
  - ♦ Reducing reliance on trained staff
  - ♦ Reducing the need for expensive real equipment

- Those which predominantly increase effectiveness by:
  - ♦ Consistent high-quality instruction on a large scale
  - ♦ High-quality training on remote sites
  - ♦ Hands-on, performance-oriented instruction
  - ♦ Individualisation of instruction

- Those which allow training, in controlled conditions, for events or situations that:
  - ♦ Cannot Normally be observed
  - ♦ Happen too quickly for the student to assimilate
  - ♦ Occur infrequently or Unpredictably
  - ♦ Are potentially hazardous

Using Networks to Enhance Computer Based Training

There are four main areas that network technology can applied to in CBL's production cycle:

- Point of Delivery

  CBT Material could be supplied at low initial cost and invoiced depending on usage as part of a pay-per-view system. This would be especially attractive to smaller companies reluctant to pay a high initial outlay and those who are sceptical about CBL techniques. Demonstration or prototype material can be delivered immediately and as many times as required without additional cost. Similarly, developers can receive real time customer response quickly without having to visit on site. Completed material can be downloaded directly to customer or run in real time over the network (lending itself to pay per view). This would eliminate the expense and lead time encountered producing and dispatching CDROMS.

- Customer Support

  The medium of the Internet would allow CBL Technology to go a step beyond providing only learning material. A complete training service could be provided that included real time on-line customer support and progress monitoring. Companies could free time by transferring complete responsibility for staff training over to CBL Technology. Students are successfully taught by standard CBL methods, but actual expertise must be learned by doing: through individual and group work on open ended projects in an information-rich, tool-rich environment. This could be applied via collaborative tutorials and computer conferencing. Available technology provides almost unlimited possibilities of collaboration and interaction between students and/or tutors. The effectiveness of any training must be assessed for 3 reasons. Firstly, the customer needs to know that it has received value for money, secondly the customer has to monitor individual progress of people exposed to the material, Thirdly the CBL developer must respond by tailoring future material to achieve best effect. The Internet or a local Intranet provides the medium over which this data can be accumulated.

- Broadening of Customer Base

  The Internet presents the opportunity to reach a vast audience of individual people many of whom will have training requirements. CBL Material could be produced for general training requirements (e.g. self help manuals, assertiveness training, software package guides) and sold to the general public as pay per view. People can be trained at a time and place that suits them. On the Internet geography is not a boundary. Opportunities could arise with higher educational institutions such as the Open University or large international companies. Within such a vast customer base many training requirements would be similar. Emphasis can be placed on developing general prototypes that fulfil these requirements. These prototypes can then be tailored with minimum effort to meet specific customer needs.

- Advertising

  CBL's home page represents an excellent opportunity to reach people who have already expressed an interest in CBT. It should be used to encourage speculative customers by presenting case studies, offering demonstrations and giving access to pay per view software. The Internet is a fantastic marketing tool. By monitoring who accesses CBL's web page it would be possible to trace the market scope, perform surveys and follow up speculative customers with e-mails. It is necessary to inform people that CBL's web page and service exists. Messages can be placed on global bulletin boards and hyperlinks created from other related web sites or hypertexts.

This project is basically tackling two weak links in the product cycle which need to be strengthened if companies like CBL Technology are to improve turnaround times, cut development costs and provide functions which customers are demanding.
The first weak link concerns the initial delivery of demonstration, prototype and completed material to the customer site, done traditionally by expensive, specially written CDROMs which cannot be edited once written, and of which one customer requires 1,500 copies. Most customers have their own computer networks (intranets), which are linked to the

Internet and are capable of downloading material quickly from a CBL site or even running material in real-time over the network via Shockwave Internet browser extensions. Through this project CBL Technology are also utilising the Internet to communicate with customers sites across the world, eliminating expensive media copying and delivery costs by downloading material to customer sites over the Internet. Development times are also improved since prototype material can be delivered and assessed by the customer much more quickly and new/updated versions delivered almost immediately without needing to write new CDROMs.

The second weak link concerns delivery to the individual students at a customer site. Students may be spread over a number of different laboratories, at different sites with some "on the road" or working from home. Traditionally this would involve the purchase of a number of CDROMs and software licenses making CBT look prohibitively expensive and with no means of monitoring who has seen what. By exploiting customer computer networks (intranets) for delivery around sites and creating a new Delivery Framework this programme will be able to provide a number of new network-based CBT facilities; specifically CBT scheduling and student monitoring, which will make their products more attractive to both existing and new customers. Computer network technology and these types of facilities also open up the prospect of group working and collaborative CBT.

SOFTWARE SPECIFICATIONS

The additional software required to exploit these developments in computer network technology and to solve the problems described falls into three categories:

A Delivery Framework

The delivery framework facilitates delivery of courses to customer contact points and installation of the courses for access by all staff at customer sites. The delivery framework is accessed via the Internet and when installed it sets up the Management Information System (MIS) database structures and places the files directing the scheduling algorithms in the correct place

Courseware Monitoring

The monitoring facilities fall into three categories:

- Obtaining student identification details: Students have to logon to their company/university network before they can begin running courseware, and it is therefore unnecessary to ask them to register again with each piece of courseware used.
- Monitoring student progress and performance: In different situation specifications will require that all kinds of different information be monitored and reported. Some applications will be interested in navigation routes, in key responses to questions, or in response times or total execution times etc; the possible list is endless. The monitoring functions therefore need to enable monitoring and reporting of any type of variable from within the courseware with a minimum of inconvenience to the author.
- Collation of data and MIS information: The data file created each time a student accesses a piece of courseware is only one piece in a much larger jigsaw puzzle. There is a requirement for a specially written collation utility to merge all files into one large database which will provide information to a training manager about the effectiveness of the courseware and who is having any problems.

Courseware Scheduling

The scheduling functions ensure that a student sees the correct modules at the correct times and in the correct sequence. These functions therefore provide a capability for a training manager to schedule courseware appropriately.
Scheduling may take place on a purely temporal basis, informing anyone who is shown in the database as not having completed the current module that a new module is imminent and that every effort should be made to complete the current module. Alternatively scheduling can take place on a pedagogical basis so that students can progress at their own pace and such that they are directed to skip modules or take a supplementary diet of material if the MIS system sees that they are either coasting through the material or struggling.

EXAMPLES OF USAGE

So far the project has developed courseware monitoring functions from the specifications outlined above. A creative CBT author can use the utilities to include additional facilities without an in-depth knowledge of how they actually work.

Monitoring and Reporting

With Authorware you construct a multimedia piece by assembling icons on a flowline. (figure 1)

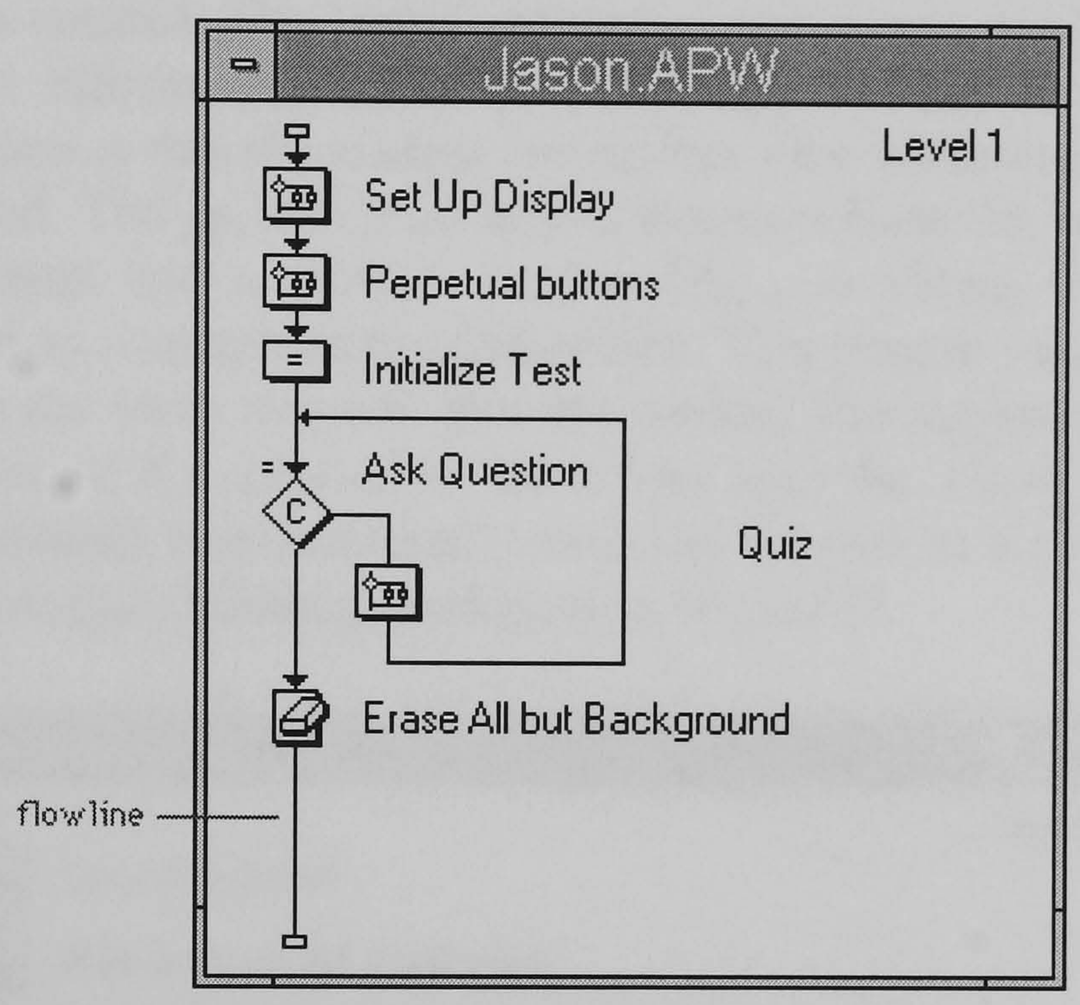


Figure 1. Example of an Authorware flowline

Icons contain the objects that you include in the piece. Different types of icons contain different types of objects, such as graphics, text, sound, digital video movies, or a set of instructions. The visual parts of the piece appear in the presentation window. This is where you arrange objects on screen and test-run the piece as you construct it.

The system implemented for monitoring user progress and performance is based on a single specially created Authorware Group icon which can be easily integrated into a CBT application and hence add monitoring functionality. By inserting the icon into the normal program flow and renaming it accordingly the icon will either automatically identify the host computer, manipulate databases or write information about a to a database of progress and performance records. All the monitoring functionality is accessed through the same generic Icon which has its functionality dictated by its title. This completely detaches the author from the inner workings of the icon's function and definitively defines the action of the icon at a glance. These monitoring functions have been implemented in this flexible and transparent manner so that an author can pass any information such as usage time, responses to questions, response times, navigation routes to the database. They utilise the

Microsoft Open Database Connectivity (ODBC) facilities with Standard Query Language (SQL) calls to a database.

The author uses the generic icon in the following manner:

The author has access to commands such as open, close, group, report. Firstly at the entrance to the CBT application the author uses the 'open' command to create a database to allow reporting of student performance. If the database is already present it is appended or if this is the first session with this piece of CBT and there is no database currently available a new one is created. The 'open' command also automatically identifies the host computer and records information about the CBT applications identity and start time. Currently the assumption is that the student has his/her own dedicated PC with the Internet protocol TCP/IP loaded. The generic icon uses a function from the winsock dynamic link library (DLL), integrated into a custom written DLL, to obtain the IP address of the machine which is used as a unique student identifier. This system will be later enhanced to automatically identify the local network that the student has logged into and extract their login details. At the exit of the application the author uses the 'close' command to log out of the database and transmit it automatically over the Internet to a central collection point at either CBL or the company training headquarters.(figure 2)

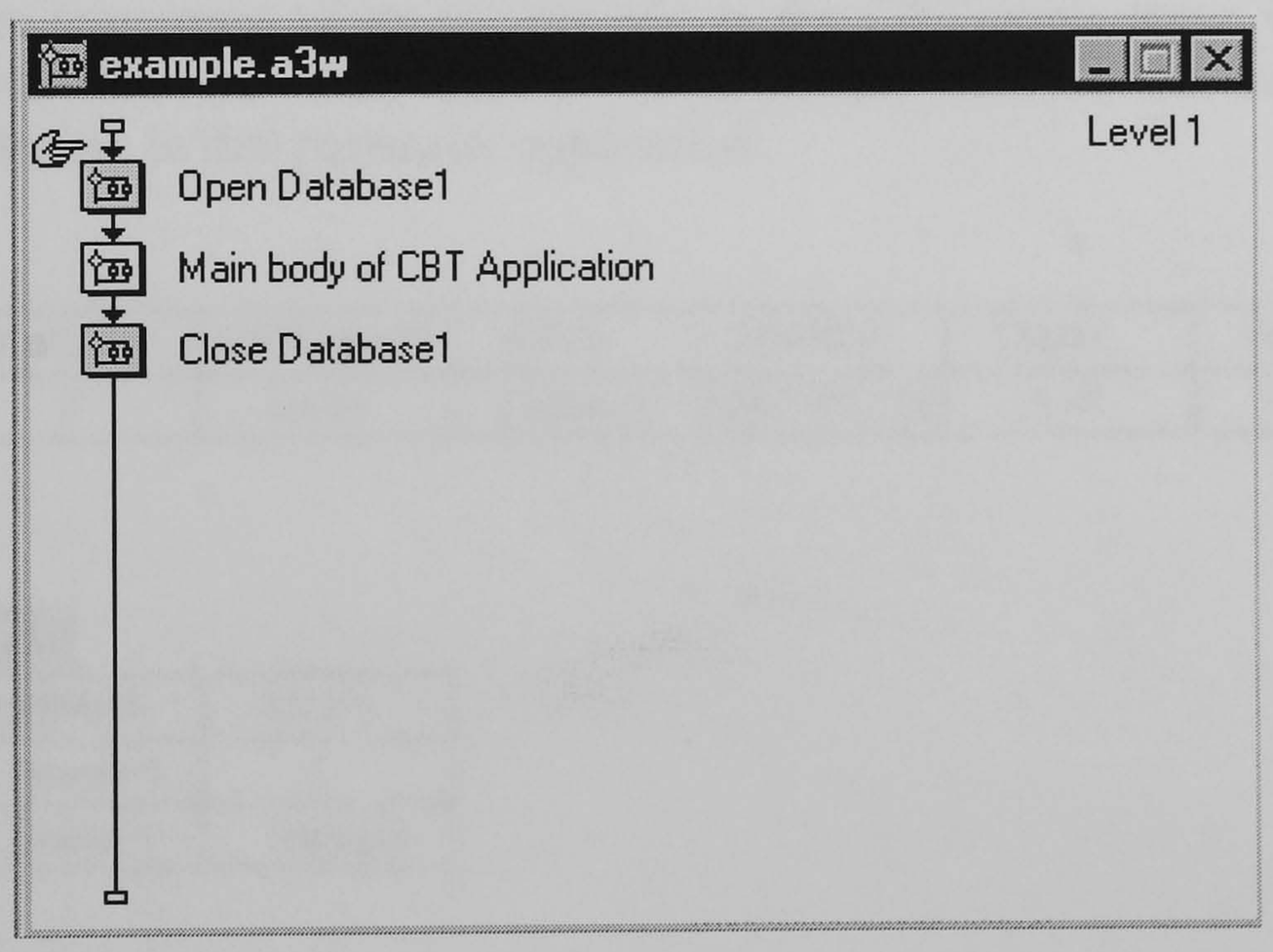


Figure 2. Example of how monitoring functions are added to the CBT

Once a database has been opened the author can add monitoring functionality to the CBT application using the commands 'group' and 'report'. Information is reported into the database using the 'report' command. The 'report' command can be used to output the name and value of any variable in Authorware. Variables can be grouped together in tables using the 'Group' command. (figure 3)

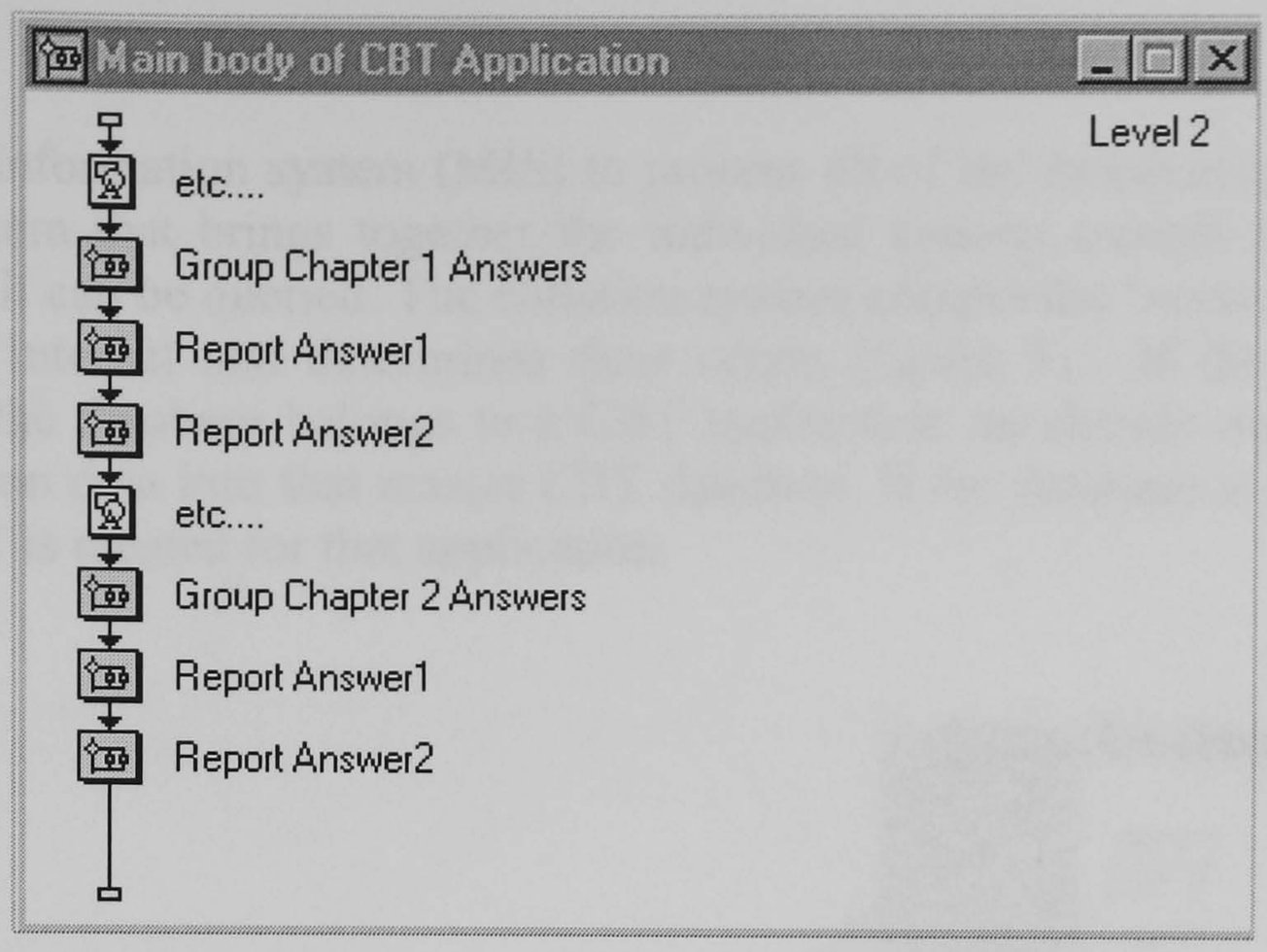


Figure 3. Example reporting data using the monitoring functions

Although most databases created will be quite complex, figure 4 shows a simple example of some records created by the monitoring utilities described above:

The database comprises of several tables. The 'Details' table is standard to all databases created by the system and contains information about the source CBT, the session, the username of the student, the host name and IP address and also when the session started and finished. The remaining tables are specific to the CBT application, their structure is standard and consistent but their content is dependant on how the author decides to implement monitoring in that particular application.

Details

CBT	SESSION	USERNAME	HOST	SOURCE	START	FINISH
Physics 1	0	Jason	Laptop 1	168.192.2.10	1630	1715

Chapter 1 Answers

SESSION	VARIABLE	VALUE
0	Answer1	5
0	Answer2	isotopic

Chapter 2 Answers

SESSION	VARIABLE	VALUE
0	Answer1	A
0	Answer2	C

Figure 4. Example of a typical database records

MIS

A management information system (MIS) to process all of the monitoring results consists of collation system that brings together the individual session records into one singular database so that it can be queried. The collation system accepts the 'session databases' that arrive over the Internet and determines their origin (figure 5). If the collator system recognises that the database belongs to a CBT application its already aware of it simply merges the session data into that master CBT database. If the database is not recognised a new master CBT is created for that application.

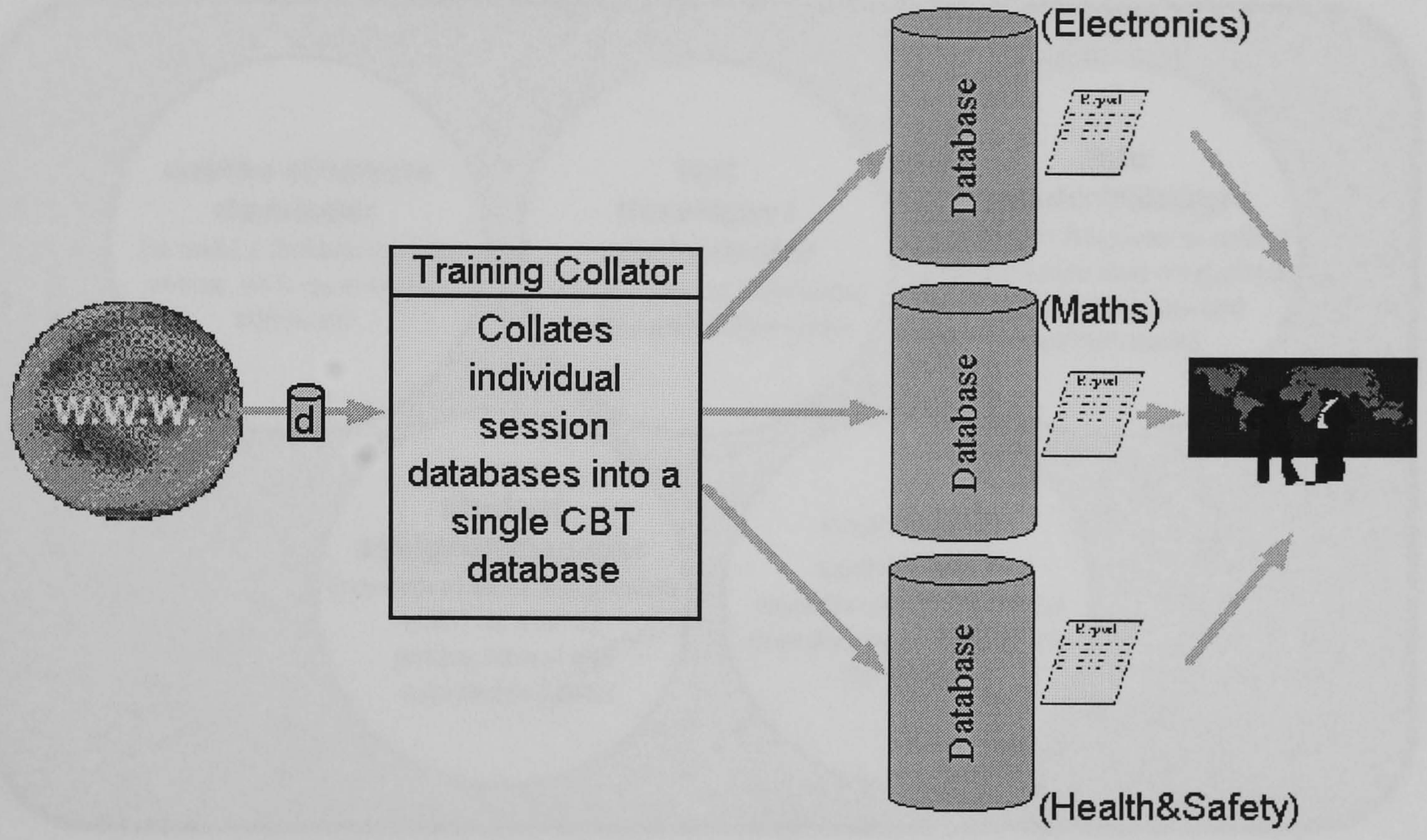


Figure 5. Illustration of the Collation system

Once a single database describing the use of the CBT application is can be queried to determine:

- What proportion of students have completed the module
- Whether anyone had any problems with the material
- Average training times
- Information as to when students find it convenient to use the material.

This information can be used by CBL as feedback into the development cycle or by a training manager/tutor to assess the success of the training.

Scheduling

The monitoring and collation system are one part of a big picture. Future work will involve designing a CBT scheduling system that integrates seamlessly with the monitoring system. The scheduling system will deliver CBT modules over the Internet and ensure that the correct module is seen by the correct user at the correct time. Future development also

includes the use of the training databases created by the monitoring system to identify personal training needs and AI algorithms to schedule a much more customised path through the available training material for each user. The scheduling system comprises of several individual components (figure 6). Firstly, the course/test developer is used to specify and introduce a course/test into the system. Students are introduce to the system via the enrolment manager which creates a training profile and sets curricula objectives. The student assignor/manager uses AI algorithms to deliver and administer the courses contained in the system to the student. Monitoring and performance related information is handled by the data collator/manager which can output status reports.

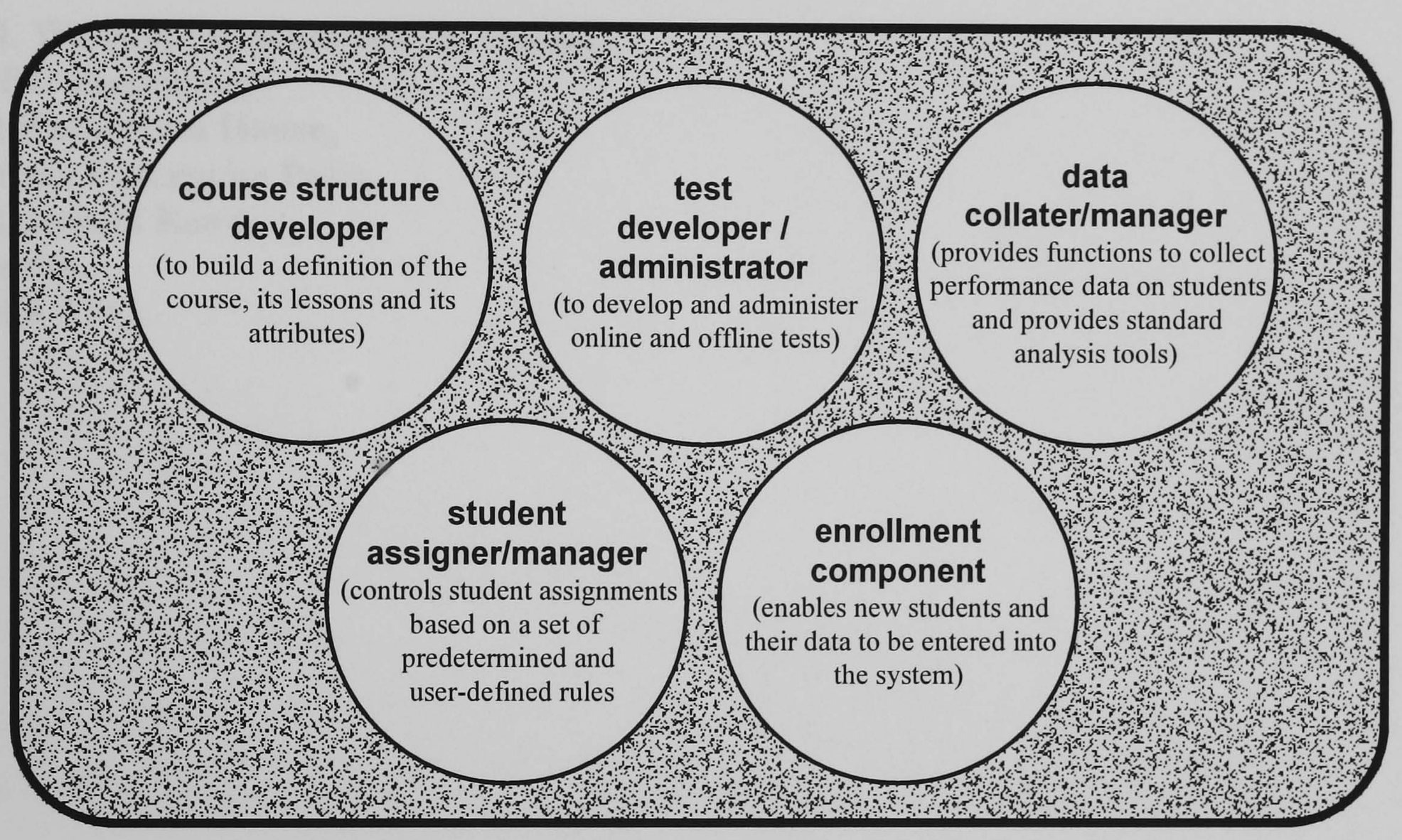


Figure 6. Illustration of the main components of a Scheduling System

CONCLUSIONS

The early stages of this project have demonstrated that traditional CBT can be much improved by exploiting computer network technology to improve delivery and turnaround times. It can also produce a far more attractive CBT product with superior functionality and many scheduling and monitoring applications, and perhaps "pay per view".
The facilities that can be offered reflect different learning and working practices that are now commonplace. The continued proliferation of computer networks will continue to promote distance learning, virtual laboratories and people working from home bases. In these types of situations the functions which this project is creating are thought to play an essential co-ordinating role.

**Dr. David Taylor
Division of Electronic and Communications,
School of Engineering,
University of Huddersfield,
Queensgate,
Huddersfield,
HD1 3DH.
Email: d.taylor@hud.ac.uk

*J. Watson, *D. Knowles, *P. Smith,
CBL Technology,
St Katherines House,
Prime Enterprise Park,
Mansfield Road,
Derby,
DE1 3TQ.

11.3 Appendix C – International Conference on Information Visualisation, 14th – 16th July 1999, SOAS, University of London.

Visualisation of Data from an Intranet Training Systems using Virtual Reality Modelling Language (VRML)

J. Watson*, D. Taylor, S.Lockwood
Dept. Electronic and Electrical Engineering,
University of Huddersfield

Abstract

This paper describes the design of a generic computer-based Training Information System (TIS) and visualisation of the data that it generates. Today's desktop training applications are delivered over the Internet to users who can exploit this flexibility and train when and where they like. However, the training manager or the author of the training material require a record of who has been trained, when they were trained and what kind of progress they made. The TIS co-ordinates this by remotely recording each training session, then sending a small summarising session database back to a central collection point for collation and subsequent analysis.

Because of the large amounts of data that this generates in large companies we are beginning to use Virtual Reality for analysis and presentation of training results. Described in this paper are the results from a session of monitored intranet training that was undertaken by approximately 60 students over a one week period.

Introduction

In today's cut-throat economic climate companies have to look more than ever for new ways of operating efficiently. When a company looks for ways of reducing its fixed expenditure it is inevitable that long-term investments such as the budget allocated for training will come under scrutiny first. So, as companies become leaner, and budgets become tighter, the money allocated for training becomes tighter. At the same time there is pressure to maintain a well-trained workforce and good ongoing training programmes are essential to the recruitment of well-qualified staff.

Computer Based Training (CBT) has now been proved, over several years and in many diverse applications, to be an extremely cost-effective way of training large numbers of people and is being examined increasingly by trainers and managers as a cost effective alternative to more traditional methods of instruction. In addition, today's company structures and work practices, where multinational companies have offices all over the world and many people hot-desk or work from home mean that computer-based instruction has many other additional advantages. For example, employees can train when they want, where they want and they can complete single modules using multiple sessions.

However, it is also extremely important for a company to maintain exact records of what training each of its employees has undergone and how well their employees have performed. This is especially so with the advent of new European legislation making companies more responsible than ever for maintaining up-to-date records of, for example, health and safety training records. A Management System can provide the instructor with numerous facilities, including selection of the most suitable instruction for a student at a particular time and the automatic testing of students both before and after instruction. The Management System creates a database to store information on the trainee learning process for each course and lesson [3]. The ability to automatically record and disseminate this type of training information is an essential ingredient of next generation computer-based training applications.

This paper describes a Training Information System (TIS), that has been developed by the University of Huddersfield and CBL Technology, and visualisation of the vast amounts of data that it generates.

CBL Technology is the training division of Vega Group and the UK's largest bespoke computer-based training and multimedia/VR analysis, design and production company, currently employing 70 full-time development staff operating within an ISO 9001-registered quality system from offices in Derby and Bristol [5].

The TIS can monitor the usage of a particular training application by any number of users anywhere in the world and can be supplied on CDROM or actually run over the Internet. Each time a training session is completed anywhere on the on the internet, or the company intranet, a record is added to the training database which can be queried from anywhere on the network by anyone with access rights. The system provides a training manager with the ability track student progress in curricula and determine which training objectives have been met. Rapid technological change means that the number of courses that have to be introduced to such a system is always increasing. Its is therefore a challenge to introduce a way to display the vast amount of information contained in the training database in a simple, innovative, and easy to visualise manner, for perusal by the training manager. The human species have an array of very advanced brain functions at their disposal. These functions include the storing of information, the processing of new information (derived from internal or external resources) and, most importantly, the ability to analyse information and make informed decisions from it [1, chapter 1]. However, humans lack the ability to absorb, retain, and process, the vast amounts of ever increasing information that is available today. Such a task is more suited to a computer. The problem of a person reacting to this data is therefore one of communicating with such a computer. Virtual-reality is perhaps the solution in providing a way to manipulate and visualise vast quantities of data using intuitive man-machine interfaces. As a consequence, we are implementing Virtual Reality to facilitate visualisation of data and relationships between data elements. The use of Virtual Reality Modelling Language (VRML) [6] opens up numerous opportunities to use attributes which are far more palatable to the human brain than conventional methods of data visualisation. We are currently researching into the use of spatial awareness, time, shapes, and sounds to enhance the understanding of complex data structures. The use of Virtual Reality will allow training managers to instantly evaluate training effectiveness, employee proficiency and enable them to hunt down and contact subject matter experts when required. Using virtual reality to visualise data is not new. Virtual reality has been used for molecular modelling, astrophysics, engineering and large-scale environmental research [2 chapter 5]. There are also existing instances of where virtual reality has been implemented to visualise complex data; Virgilio, is a system that generates 3D visualisations of complex data objects [4].

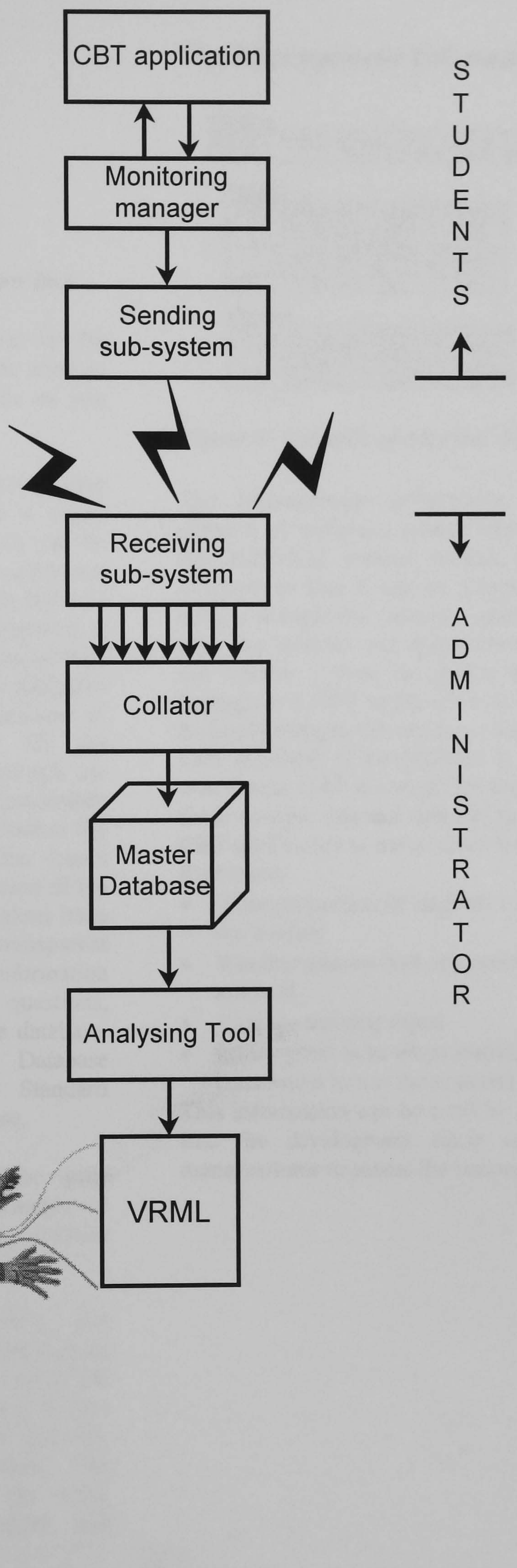
Examples Of Usage

We have developed a Windows dynamic link library (DLL) of C++ functions that the creative CBT author can use to include additional monitoring facilities into training material without an in-depth knowledge of how they actually work. The system (figure 1) consists of two geographically separated sections. The 'Student' or 'Client' section consists of the CBT itself and the integrated monitoring and transmission system. This 'Student' section transmits information about student activities across a network medium (such as the Internet) to the 'Administrator' section. There is only one 'Administrator' section, but there could be thousands of instances of the 'Student' section. The 'Administrator' section collates all of the information it receives from each of the students into a single master database, and ultimately, into Virtual Reality Mark-up Language (VRML) code. The Training Manger is able to view the vast quantities of data such a system provides, summarised in an easy to explore virtual world.

Monitoring and Reporting

Within Authorware the author constructs an application by assembling icons on a flow-line. (figure 2)
Icons contain the objects that you include in the piece. Different types of icons contain different types of objects, such as graphics, text, sound, digital video movies, or a set of instructions.

Figure 1. Schematic of the Training Information System



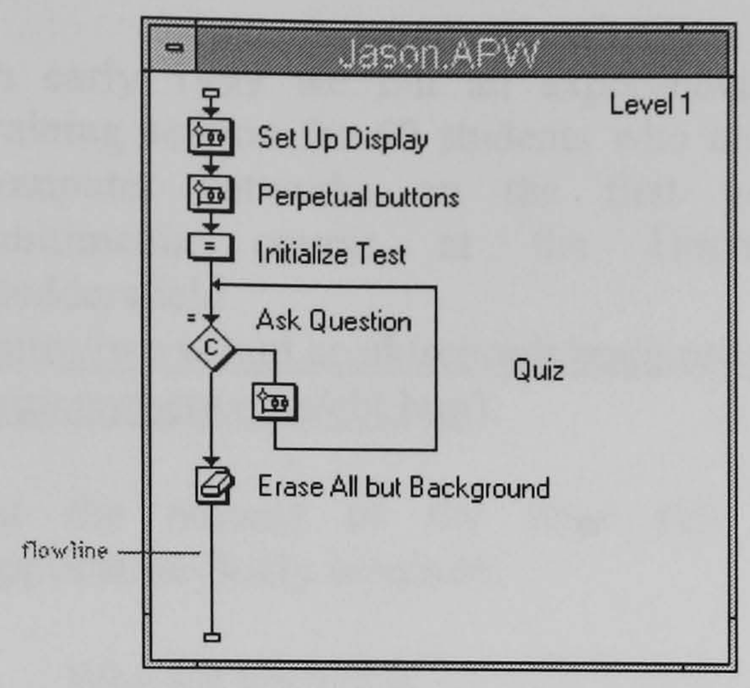


Figure 2. Example of an Authorware flow-line

The visual parts of the piece appear in the presentation window. This is where you arrange objects on screen and test-run the piece as you construct it.

The system implemented for monitoring user progress and performance is based on a single specially created Authorware icon which can be easily integrated into a CBT application and hence add monitoring functionality. By inserting the icon into the normal program flow and renaming it accordingly the icon will either automatically identify the host computer, manipulate databases or write information about a to a database of progress and performance records. All the monitoring functionality is accessed through the same generic Icon which has its functionality dictated by its title. This completely detaches the author from the inner workings of the icon's function and immediately defines the action of the icon at a glance. These monitoring functions have been implemented in this flexible and transparent manner so that an author can pass any information such as usage time, responses to questions, response times, navigation routes to the database. They utilise the Microsoft Open Database Connectivity (ODBC) facilities with Standard Query Language (SQL) calls to a database.

Although most databases created will be quite complex, figure 3 shows a simple example of some records created by the monitoring utilities described above:

The database comprises of several tables. The 'Details' table is standard to all databases created by the system and contains information about the source CBT, the session, the username of the student, the host name and IP address and also when the session started and finished. The remaining tables are specific to the CBT application, their structure is standard and consistent but their content is dependant on how the author decides to implement monitoring in that particular application.

The Management Information System

Details

CBTname	Session	Username	Host	Source	Start	Finish
Networks	0	Watson	161.12.6.23	Source	3-5-1999 1632	3-5-1999 1713

Types

Session	Time Reported	Variable name	Variable Value
0	3-5-1999 1634	Question 1	Correct
0	3-5-1999 1638	Question 2	Incorrect
0	3-5-1999 1639	Question 3	Not attempted
0	3-5-1999 1645	Question 4	Correct
0	3-5-1999 1648	Visits	3/5

Protocols

Session	Time Reported	Variable name	Variable Value
0	3-5-1999 1655	Question 1	Not attempted
0	3-5-1999 1657	Question 2	Incorrect
0	3-5-1999 1701	Visits	8/9

Figure 3. Example of a typical database records

The Management Information System (MIS) consists of collation system that brings together the individual session records into one single database so that it can be queried. The collation system accepts the 'session databases' that arrive over the Internet and determines their origin. If the collator system recognises that the database belongs to a CBT application its already aware of it simply merges the session data into that master CBT database. If the database is not recognised a new master CBT is created for that application. Once a single database describing the use of the CBT application is constructed it can be queried to determine:

- What proportion of students have completed the module
- Whether anyone had any problems with the material
- Average training times
- Information as to when students find it convenient to use the material.

This information can be used by CBL as feedback into the development cycle or by a training manager/tutor to assess the success of the training.

Data Visualisation

In early 1999 we ran an experimental on-line training session for 60 students who are studying computer networks on the first year of a multimedia course at the University of Huddersfield (http://www.hud.ac.uk/schools/engineering/undgrad/mmvr/networks/cbt.htm).

At the request of the tutor this particular application firstly monitors:

- Who the student is
- When they accessed the application
- Where they accessed the application from

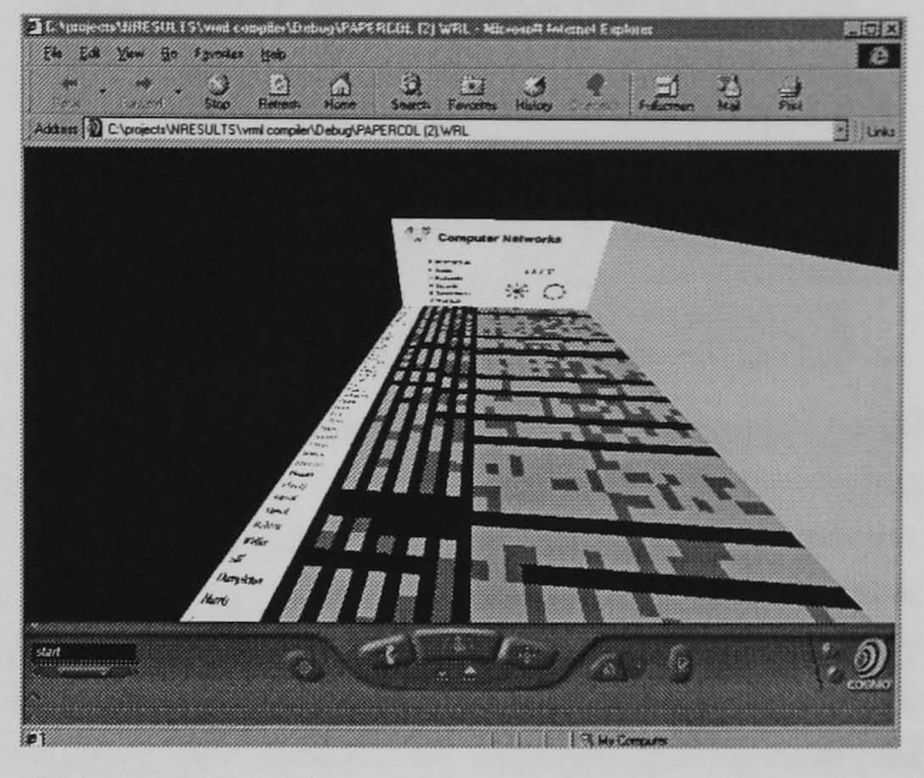
It records:

- How long the session lasted
- Which chapters/pages they visited
- How long they spent there
- Their response to tutorial questions delivered at the end of each section
- Their performance in an end of Module assessment

All of this data for all of the students therefore gets transmitted to a central collection point where it is collated into one large database. This database can of course be queried, but the results of the queries are often difficult to interpret. In the current phase of the project we are therefore looking at data visualisation in three dimensions. Since the whole project is Internet based, the obvious tool to do this with is VRML. Also a VRML description is a neutral ASCII text format file which we can easily generate automatically from the contents of a database. VRML also requires no compiler or specialist viewer and therefore requires no additional investment by the user.

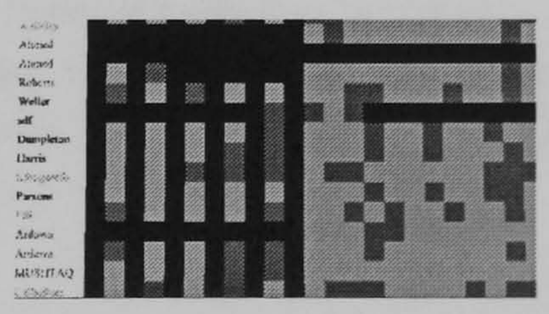
As a consequence of the decision to use VRML, we have developed an application (written in C++/MFC) that examines the contents of the training database and automatically generates the VRML to represent the data. The VRML 'worlds' that the application generates can be placed directly onto a web-site for perusal by both students, and training managers, at any location. Currently, the system is able to represent the data in three different types of virtual worlds; The 'World of Colour', the 'World of Columns', and the 'World of Spheres' world. Various studies on database visualisation systems have been conducted, but they are proprietary systems aimed at their specific objectives and could not successfully integrate with training information systems (e.g., statistical data [7], medical data, document databases [8],[9],[10],[11]). Unlike these existing systems, our solution provides quick, light and flexible approach to data visualisation.

"The World of Colour"



In this world, student responses to questions, and the extent of the content in each section they have viewed, are both represented by colour. If a student has visited a section it is demonstrated by a blue colour. The brighter the shade of blue, the more content the student actually viewed in that section. If a student has responded correctly to a question then the response is shown by the colour green. An incorrect response is indicated by the colour red. This method of data visualisation demonstrated the benefits of being able to visualise large amounts of data from different perspectives. If the user orientated themselves so that the Y axis was in front, data could be observed on individual student performance. However if the user orientated themselves so that the X axis was in front, data could be observed on overall acceptability of an individual question. This first attempt at data visualisation method did not employ many of the facilities that VR can offer to assist with data visualisation, (such as different shapes, size etc..).

View in this direction to see individual student progression

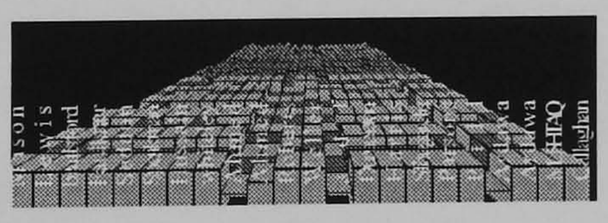


View in this direction to see question

"The World of Columns"

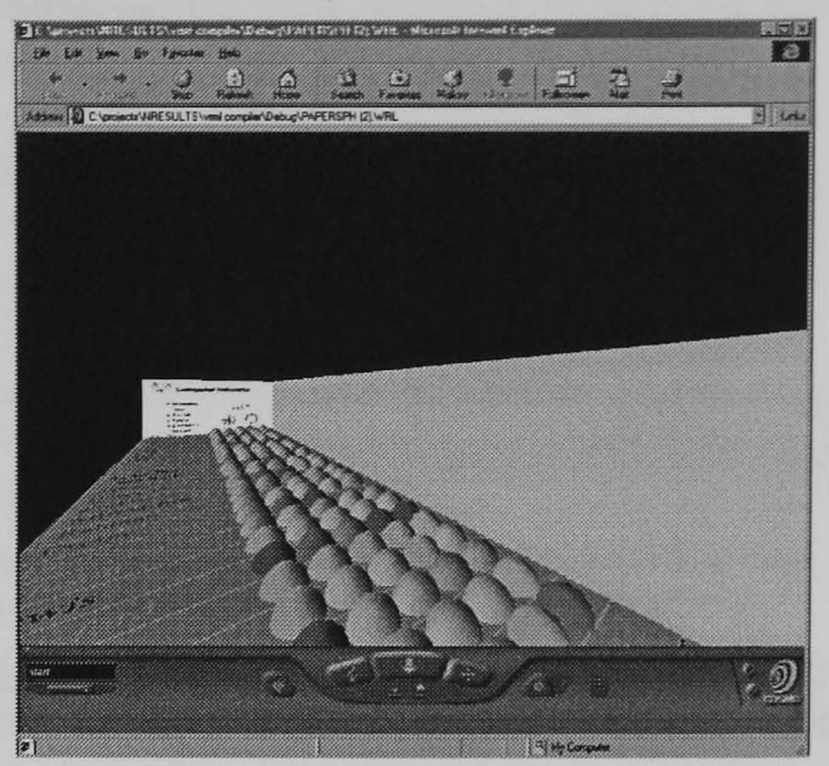
In this world, student responses to questions, and the extent of the content in each section they have viewed, are represented by the height of the columns. The tall columns represent questions with which students had difficulty with. If a column is absent, then the student failed to visit that section of material, or answer that particular question. By orientating themselves correctly, the user is able absorb large amounts of information by performing a 'fly-by' over the field of columns. If the user flies over the terrain in one direction individual student performance can be assessed. Overall course suitability for the students can be assessed by flying over the terrain in the other. However, this visualisation method did prove to be problematic. If the columns were viewed close up, the use of height made overall evaluation of the whole scene difficult (the columns are not translucent, and the users vision of the remainder of the terrain would be blocked). However, if the user was sufficiently high enough, then it was difficult to keep track of which line of results (or course acceptability by students) was being viewed. We decided that the use of height, with this quantity of results hindered successful visualisation.

View in this orientation to assess course acceptance



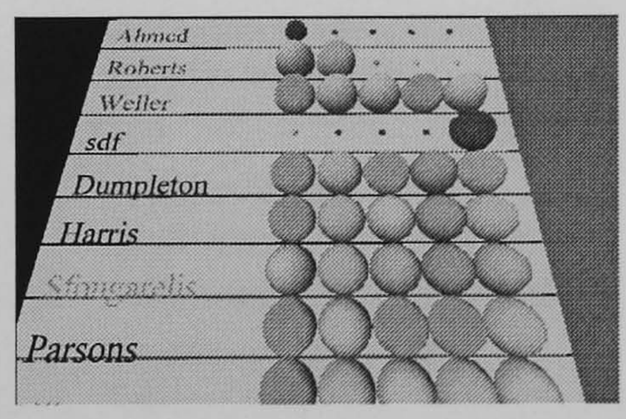
View in this orientation to assess individual student progress

"The World of Spheres"



In this world, student activities and responses to all questions have be summarised into 5 autonomous chapters. Each chapter is represented by a sphere. As a consequence each student has 5 spheres that together represent his/her total training activity within the courseware. The size of a sphere represents the amount of content/lecture-material that the student viewed in that particular chapter. The colour of the sphere varies in shades between red and green depending on how well the student performed in the assessment linked with that particular chapter. If the student got all questions correct, the sphere is fully green, if the student got all questions incorrect, the sphere is fully red. This visualisation technique allows the training manager to make comparisons between students at a glance.

View in this orientation to assess student progress in specific chapters.



View in this orientation to assess overall acceptance of each chapter.

This technique also allows different categories of students to be easily distinguished between. For example:

Small green sphere:	Indicates a student who performed extremely well in the assessment, without consulting much lecturing material.
Large green sphere:	Indicates a student who performed extremely well in the assessment, and viewed all the lecture material.
Large red sphere:	Indicates a problem student who performed badly in the assessment, even after viewing all the lecture material.
Small red sphere:	Indicates a problem student who performed badly in the assessment, but did not bother with the lecture material.

The tutor is therefore primarily looking for any large red spheres that indicate that a student has not understood it.

Conclusions

The early stages of this project have demonstrated that traditional CBT can be much improved by exploiting computer network technology to improve delivery and turnaround times. It can also produce a far more attractive CBT product with superior functionality and many scheduling and monitoring applications, and perhaps "pay per view".

The facilities that can be offered reflect different learning and working practices that are now commonplace. This project was developed to provide a flexible solution to the challenging requirements of a training manager. The developed system allows any CBT application to be run by any user from a distributed WWW site, vastly easing the dissemination of their training material. In addition, the system automatically monitors what has which student has seen which material and when, along with any other necessary information that the training manger wants to record. These features, combined with the automatic generation of Virtual Worlds to facilitate visualisation of reported data provides both authors of training, and training managers with a complete solution for delivering, monitoring and tracking of activities in computer-based training.

Feedback from both the students and the group tutor was both positive and interesting. The most striking feedback was how negative some students felt about having their actions monitored to such an extent.

The use of the networked CBT and data visualisation has allowed the tutor to:

- Monitor attendance at the session
- Identify students who had trouble understanding the material
- Correlate marks for the written submissions with student'' performance in and attendance at the CBT Session

At this stage, this project has not exploited the Virtual Environment to its full potential. The work completed so far is in early stages and does not benefit fully from using the interactivity that determines what is considered by some people as the only true Virtual Reality. Further work will explore interactivity between the Training Manager and the objects in the virtual worlds. In addition, the benefits from applying the use of characteristics such as shapes, sound, and Time (through animations) to represent data will be explored.

The proliferation of computer networks will continue to promote distance learning, virtual laboratories and people working from home bases. In these types of situations the functions which this project is creating are likely to play an essential co-ordinating role.

References

[1] Philip Barker, Harry Yeates, "*Introducing computer assisted learning*", Prentice/Hall, 1985.

[2] Joshua Eddings, "*How Virtual Reality Works*", Ziff-Davis Press, 1994

[3] Roy Rada, "*Multimedia Education*" - Handbook of Internet and Multimedia Systems and Applications, CRC Press/IEEE Press, 1999

[4] Antonio Massari , Lorenzo Saladini, Fabio Sisinni, Walter Napolitano, "*Virtual Reality Systems For Browsing Multimedia*" Handbook of Internet and Multimedia Systems and Applications, CRC Press/IEEE Press, 1999

[5] "*CBL Technology Ltd.*", Promotional Material for CBL technology Ltd, January 1996

[6] "*The Virtual Reality Modeling Language*" Version 2.0 Spec.http://vag.vrml.org/ VRML2.0/FINAL/spec/

[7] L. Meo-Evoli, M. Rafenelli, F.L. Ricci, "*An Interface for the Direct Manipulation of Statistical Data*",

Journal of Visual Languages and Computing, 1994

[8] M. Hemmje, "*A 3D Based User Interface for Information Retrieval Systems*", Proceedings IEEE Workshop on Database Issues for Data Visualisation, Vol.871, Oct. 1994

[9] U. Krohn, "*VINETA: Navigation through Virtual Information Spaces*" Proceedings Workshop on Advanced Visual Interfaces, 1996

[10] B. A. Wilson, R. H. Fowler, W.A.L. Fowler, "*Integrating Query, Thesaurus, and Documents through a Common Visual Representation*", Proceedings of the 14th Annual International ACM SIGIR conference on Research and Development in Information Retrieval, 1991

[11] M. Chalmers, P. Chitson, "*Bead: Explorations in Information Exploration*", Proceedings SIGIR'92, published as a special issue of SIGIR Forum, ACM Press, pp. 330-337

11.4 Appendix D – Monitoring System Student Feedback Form

NFE145/6 Computer Systems
Computer Networks Lecture & Lab Friday 23/04/99

you do not have to identify yourself on this form		(delete as appropriate)	
Do you think the Computer-Based lecture gave you enough information about "Computer Networks"?		Yes	No
Do you think the content and the detail contained in the application was…..	Too Much	Not enough	About right
Would you like other subjects delivered in a similar way?	Yes	No	Depends (explain below)
Did you log-in to the training using your own name?		Yes	No
Did you think the assessment was…	Hard	Medium	Easy
Were you worried about your actions being automatically monitored, and your assessment being recorded?		Yes (explain below)	No
Did you complete the assessment?		Yes	No
Have you used the training outside of the allotted time?		Yes	No
Would you have been keen to identify yourself correctly and complete the assessment if a certificate had been issued?	Yes	No	Neither
Would you have been keen to identify yourself correctly and complete the assessment if your results were displayed on the Internet for everybody to view?	Yes	No	Neither
How long did it take you to complete the training and assessment?		Ans=	

If you have any further comments about this method of lecturing/assessment please write them below:

11.5 Appendix E – Generic Monitoring System Manual

Introduction

This document describes the how to use the Generic Monitoring and Management Information system that has been designed for integration into any CBT application. There is no bespoke method that can be used to monitor all CBT applications. CBT by its nature is too diverse and customer requirements quite often different. This system combats this by providing the author with the flexibility to create their own monitoring system i.e. the definition of 'monitoring within CBT' will be up to the author rather than set by the system; however, this system provides them with the tools to do the job.

The generic monitoring system is split into 2 parts: *Reporting* and *Collating*. The author uses the reporting system to create a local database which describes a students' activities during a session. A session is defined as a single period of time a student uses CBT. A student could take several sessions to complete all the training in a CBT application and there could also be many students undergoing the same training. Consequently, the Reporting system would normally produce numerous databases each which describe the student activities within a session, and together (more importantly), describe who has used the CBT application and how they have used it. The collation system merges all the separate individual session databases together into a single entity (a master database) that is queried to provide information that a training manager would require.

The Reporting System

Files included in /reporter/:

TcsMonIcon.a4d	-	This Authorware Model contains a generic group icon. The author uses this icon to communicate with the user defined code TcsMon32.u32
TcsMon32.u32		This user-defined code library contains the reporting functionality
TcsMon32.ini	-	This ini file is read by TcsMon32.u32. The author edits this to set up paths and alter default settings
MonError.a4l	-	This library contains a bitmap that TcsMonIcon.a4d uses to display errors

Using the Reporting System

The reporting functionality is contained in the user-defined code library "TcsMon32.u32". The author uses this functionality to open, report to and close databases. The author communicates with the user-defined code library by typing instructions in the title of the generic group icon contained in "TcsMonIcon.a4d".

Reporting data consists of the following steps:
1. opening a database
2. specifying a group
3. reporting variables
4. closing a database

All the above steps are performed by placing instructions into the title of the generic group icon. The title (instruction) of the icon consists of 2 parts:

command<space>operand

The command tells the system what to do with the operand.

The commands available for use by authors are:
- Open
- Close
- Group
- Report
- Point

The commands are not case sensitive and the 'group' and 'report' which are used the most frequently can be shortened to 'grp' and 'rep'.

The example in Figure 1. shows how a database can be opened and closed. The 'Open' command creates a database on the local hardrive. The format and location of the database is dictated by parameters in the 'TcsMon32.ini' configuration file. The physical name of the database is set to the first 8 characters of the operand, or, should the operand also be a variable, the 1st 8 characters of that variable.

When a database is opened the following information is automatically recorded:
- The name of the CBT application (as set in TcsMon.ini)
- The host-name of the local computer
- The IP address of the local computer
- The time and date

When a database is closed the time and date is again recorded. All databases must be closed before the user exits the CBT. If a database is left open, a dialogue box will appear each time the CBT is exited (or programming environment closed) explaining this to the author and closing them automatically.

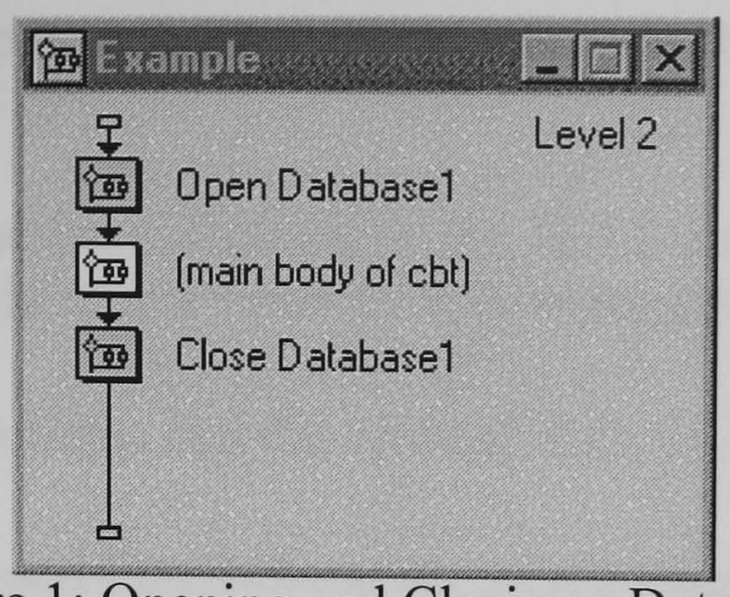


figure 1: Opening and Closing a Database

Once a database is open variables can be reported into it. The example in Figure 2. shows how this is done. It is best to group similar variables together. The 'Group' command does this by creating a table in the database. The name of the table is the instruction's operand. An author may have as many Groups (tables) as they wish and are only limited by the maximum number for the database format. The group command is initially used to create a group that does not exist but it is also used to switch between existing groups.

The 'Report' command is used to report a variable. The name of the variable to be reported is contained in the instruction's operand. When 'Report' is used the following information is recorded in the currently active group:

- Date and Time
- Variable Name
- Variable Value

The maximum size of any variables reported is set by a parameter in TcsMon.ini.

If variables are reported when no group has been defined, an automatic default group called 'Group1' is created. Authors should normally use the 'Group' command at least once before reporting information.

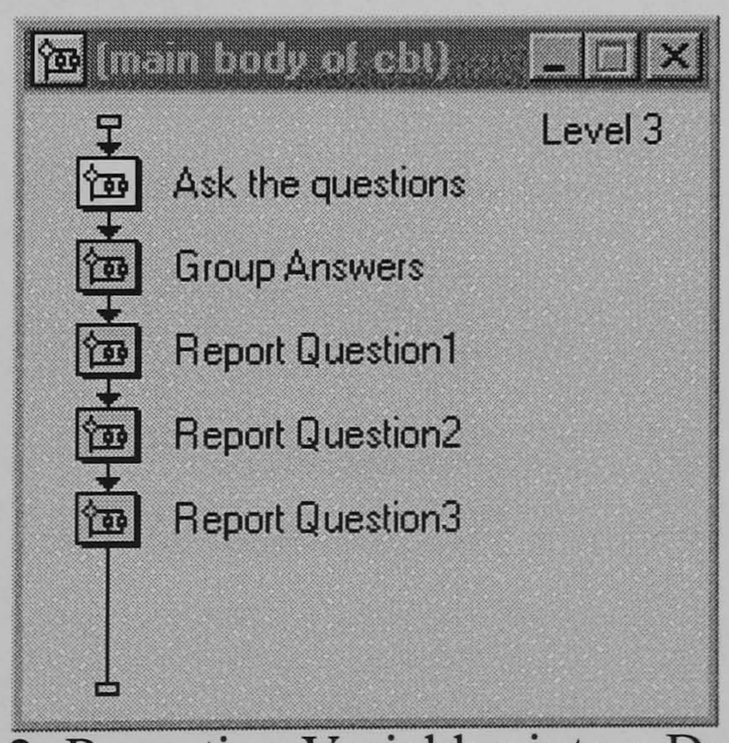


figure 2: Reporting Variables into a Database

The reporting system allows more than one database to be used concurrently. Figure 3. shows how an author can open several databases. All databases must be closed before the

CBT is exited. Each database must have an unique name in the operand of its 'Open'/'Close' instructions.

The physical name of the database is dictated by the first 8 characters of the operand, or the value of the operand should it be a variable. It is important that:

- The first 8 characters are unique and the operand isn't a variable

or

- The operand is a variable and first 8 characters of the value of the operand are unique.

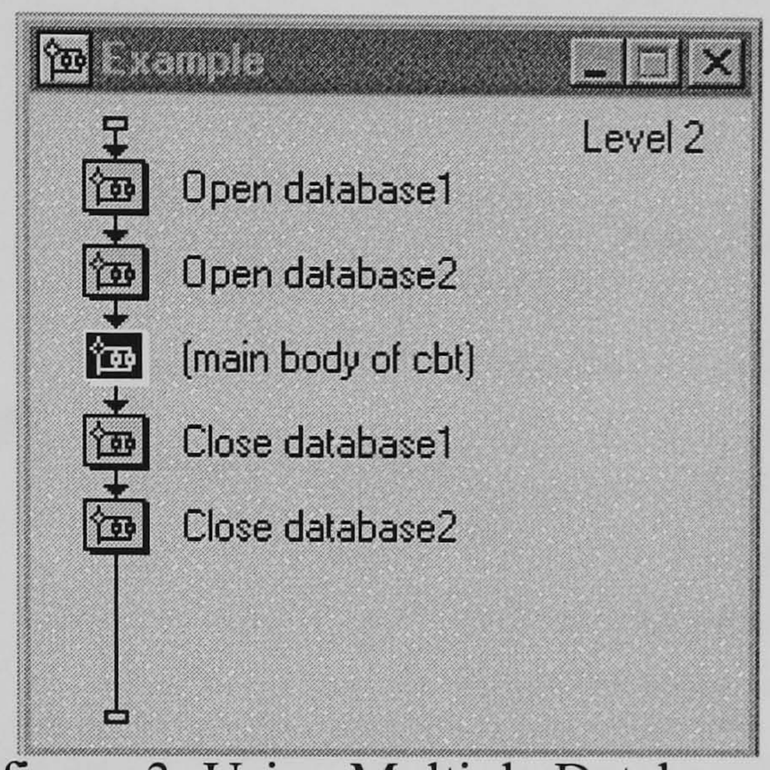


figure 3: Using Multiple Databases

The Reporting system allows the author to switch between available databases by using the 'Point' command. All grouping and reporting is done on the current database. A database achieves current status immediately after it has been opened or immediately after a 'Point' command has been used on it.

Figure 4. shows an example of how an author can switch between different databases and tables at will.

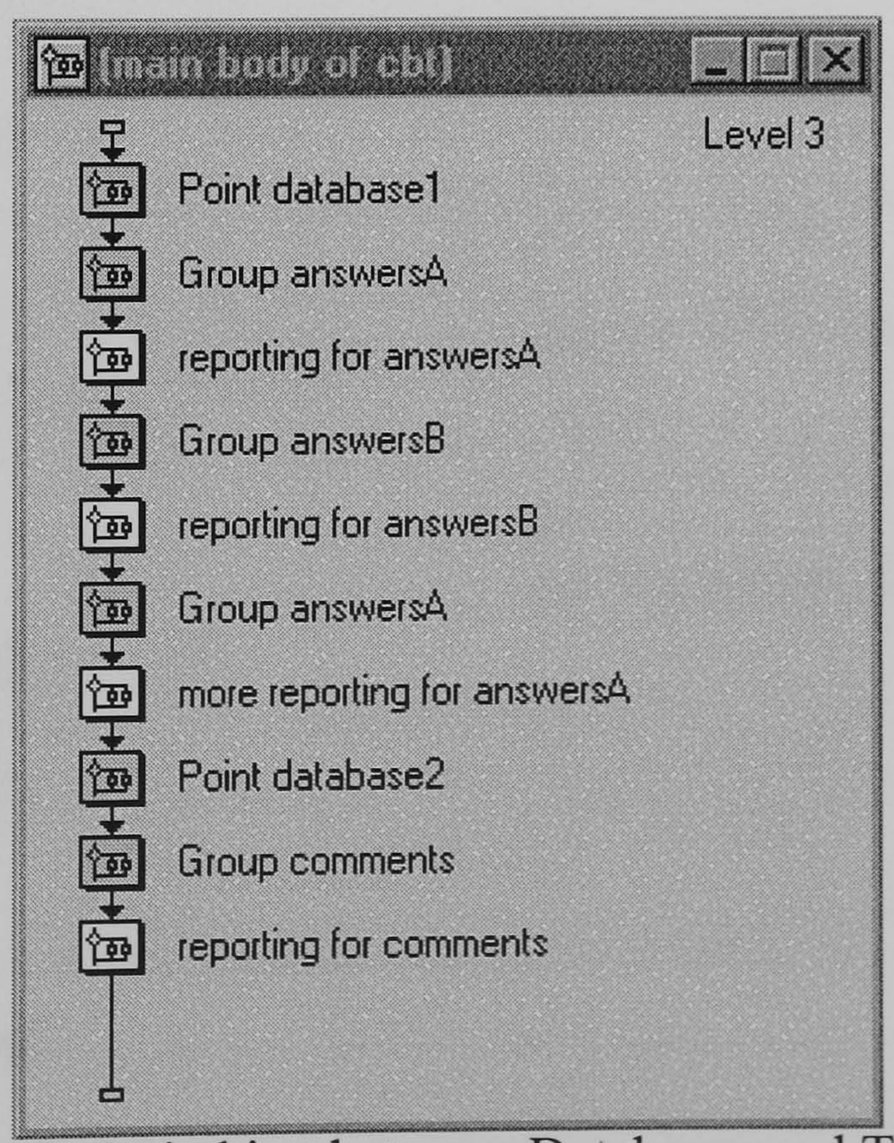


figure 4: Switching between Databases and Tables

Altering the Reporting System configuration file (TcsMon32.ini)

[CBT Information]
CBTUniqueID=CbtID - The ID of the CBT that any of the databases produced by the CBT will be stamped with

[File Locations]
Output=c:\temp\coll_in - The directory in which databases will be stored

[ODBC]
UseODBC=yes - if yes, then the system will try to use ODBC. If no, or if ODBC not present system defaults to MSTEXT format
ConfigureDLL=odbccp32.dll - name of odbc configuration dll to use
FunctionsDLL=odbc32.dll - name of odbc dll to use
FormatLabel=Access - Tells the system which set of ODBC driver parameters to use. Must equate to label below - or defaults to ACCESS

[Other Resources]
WinsockVersion=257 - The lowest version of winsock to use 1.1 = 257

[Field Sizes]
Variables=50 - max size of variable name
ValuesDefault=50 - max size of variable that can be reported

[Dbase] - Dbase ODBC driver parameters
Type=Microsoft Dbase Driver (*.dbf)
FileType=DBASE4

[Access] - Access ODBC driver parameters
Type=Microsoft Access Driver (*.mdb)
FileType=RedISAM

[MsText] - MsText ODBC driver parameters
Type=Microsoft Text Driver (*.txt; *.csv)
FileType=TEXT

The Collating System

Files included in /Collator/:

coll32_1.exe - This application merges all the separate individual session

	databases that have been created by the reporting system into a single entity (a master database)
coll32_1.ini	This ini file is read by coll32_1.exe and may be altered to set up paths or change default settings

Using the Collator System

The Collator system consists of a single application "coll32_1.exe". The system is currently compatible with Access7, Dbase4, MsText database formats. When the application is running the training manager is presented with the dialogue box shown in figure 5. To perform a collation the training manager would click on the large button marked 'Collate'. Progress is indicated by the progress bar and the action the application is performing is indicated by the status bar on the bottom of the dialogue box. The status bar will indicate when the collation is completed and if there were any errors. Any errors can be viewed by clicking on the 'View Error Log' button.

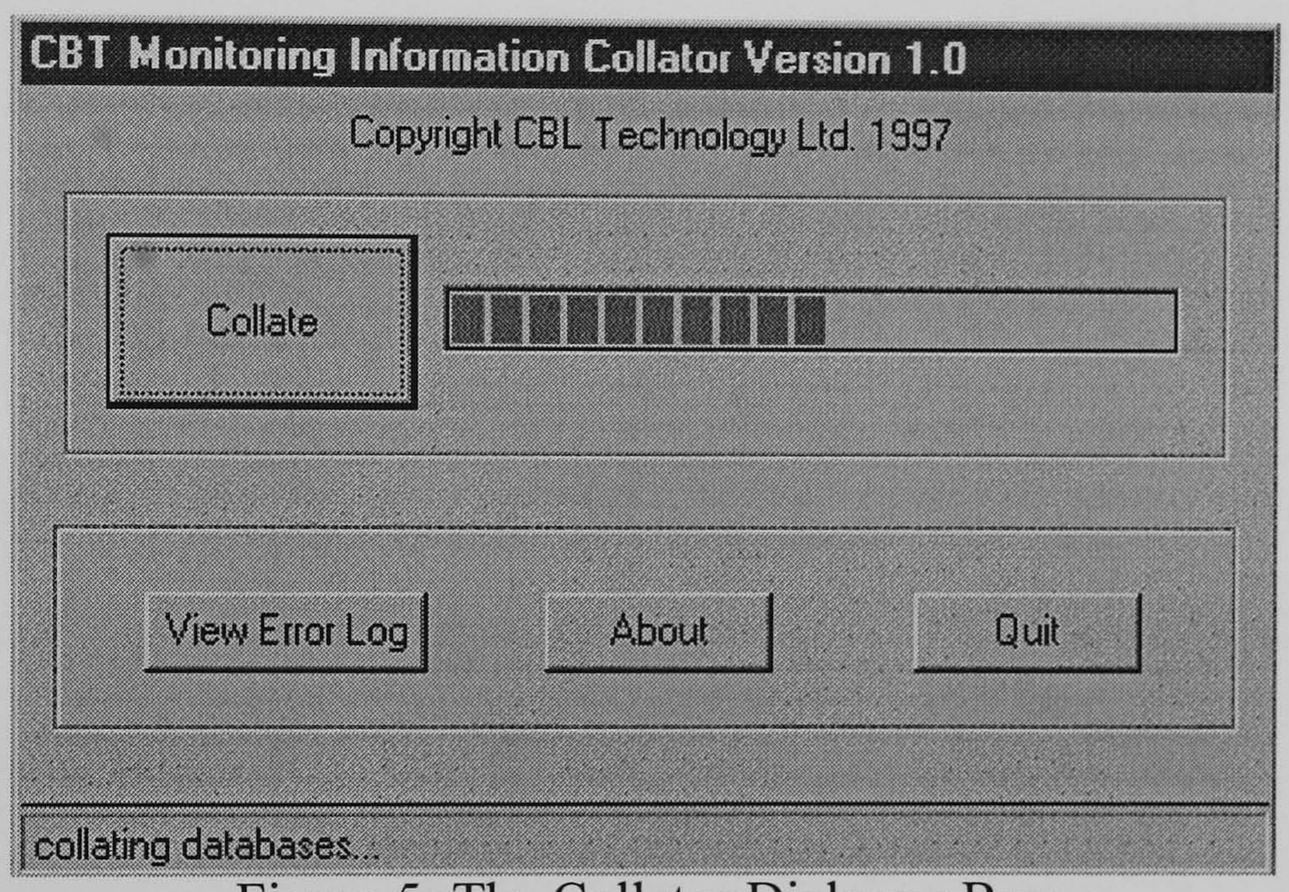


Figure 5: The Collator Dialogue Box

Altering the Collator System configuration file (coll32_1.ini)

[FileLocations]

InputFiles=c:\temp\coll_in	- The directory in which the incoming session databases reside
OutputFiles=c:\temp\coll_out	- The directory containing the master databases - or to put it if it doesn't exist
ErrorLog=c:\temp\coll_out	- Location of error log

[DatabasePreferences]

OutputDatabaseFormat=Dbase	- Tells the system which set of ODBC driver parameters to use for output. Must equate to label below - or defaults to ACCESS
DeleteAfterCollation=yes	- Delete incoming databases option -

should always be set to yes

```
[Dbase]                                    -  Dbase ODBC driver parameters
 Type=Microsoft Dbase Driver (*.dbf)
 FileType=DBASE4

[Access]                                   -  Access ODBC driver parameters
 Type=Microsoft Access Driver (*.mdb)
 FileType=RedISAM

[MsText]                                   -  MsText ODBC driver parameters
Type=Microsoft Text Driver (*.txt;
*.csv)
FileType=TEXT
```