Country Code 21.

PROTOTYPE ELECTRONIC STUDENT ASSESSMENT AND DATA MANAGEMENT SYSTEM

S Chester*, DA Sanders**, A Lassaniere*** and GE Tewkesbury****

*S Chester Associates, Inglis Road, Southsea, Portsmouth.PO5 1PB.
**University of Portsmouth, Portsmouth, PO1 3DJ.
*** Counterpoint MTC Ltd. Unit 2, Decoy Rd, Timberlaine Estate, Worthing, West Sussex BN14 8ND
****ICTHUS Technical Services, 121 Manners Grove, Portsmouth, PO4 0BD

simon.chester@simonchester.com

Abstract.

This paper describes a novel electronic assessment system called Electronic Student Assessment And data Management System' (ESAAMS). Audio, visual and data recordings (including teachers comments) are immediately stored. The objectives of the research are to discover and list gaps in existing electronic assessment systems, and create new electronic assessment systems.

Introduction

The term 'assessment' could be used in many aspects of education such as suitability and effectiveness or aspects of institutional performance. For example: Abate [n.d.] gave a definition of assessment that focused on the evaluation of program quality, rather than evaluation of student activity and learning. However, much of the literature considered assessment where the performance of individual students was judged by teachers [Brookhart & Durkin, 2003]. It is this form of 'student assessment' that is discussed in this paper because student assessment plays an integral role in teaching and learning [Shepard, 2000].

A basic working system called 'Electronic Student Assessment And data Management System' (ESAAMS) existed at the beginning of the research and a prototype system (ESAAMS Version 1) had been created in 2001. This paper described ESAAMS Version 2 (first mentioned in Lassauniere [2003]). A review of assessment software was conducted before the beginning of the research and no comparable systems were found. Some software systems were

available for recording the results of students' summative assessments and attendance information but there were no software systems for capturing and storing more complicated information such as audio/visual work. Other, non-software based, methods of capturing and storing students' audio/visual work were available and established in classroom practice [Ofsted, 2003].

ESAAMS Version 2 allows teachers to capture student work using various audio and video capture devices (such as video cameras and microphones) attached to Teacher PCs. This system is teaching-centred, suitable for a range of teaching fields, and allowed quick and efficient management of student work.

ESAAMS Version 2 has been tested at a collaborating company (Counterpoint MTC Ltd) and distributed to teachers for further testing. Mackrill [2004] created and distributed a questionnaire to teachers who had used the system and results were collected. Follow-up interviews were conducted with teachers to gain more detailed information about specific areas.

Student Assessment

The term 'assessment' could be used in many aspects of education such as suitability and effectiveness of teaching practices or aspects of institutional and teacher performance. Abate [n.d.] gave a definition of assessment that focused on the evaluation of program quality, rather than evaluation of student activity and learning. However, much of the literature considered assessment where the performance of individual students was judged by teachers [Brookhart & Durkin, 2003]. It is this form of 'student assessment' that is

discussed in this paper.

Assessment activities helped students learn and gauged student progress [Wiliam & Black, 1996]. The literature discussed two primary forms of student assessment: 'formative' and 'summative'.

Formative assessment provided specific and diagnostic feedback to students to improve during learning rather than aiming to determine success or failure only after the event [Topping *et al.*, 2000]. Topping *et al* [2000] suggested that formative assessment was most helpful if it yielded rich and detailed qualitative feedback information about strengths and weaknesses, not merely a quantitative mark or grade. Summative assessment was used to record a student's level of achievement at a given point in time as a grade [Wininger, 2005].

Both formative and summative functions of assessment have needed evidence of performance or attainment that has been interpreted and acted upon, in some way. These actions may then directly or indirectly generate more evidence so that this process was iterative. The key agents in this process were the assessed and the assessor, often called the teacher and the student [Wiliam & Black, 1996].

The models and systems presented in this Dissertation were designed to support formative and summative functions of student assessment.

Audio Assessment.

Feedback could powerfully influence student learning [Rotherham, 2007]. Rust [2001] suggested that teachers were under pressure to find and use assessment techniques that were both efficient and effective. Audio feedback using tape could be advantageous because information could be conveyed to students more quickly than with written comments. Rotherham [2007] suggested that teachers had not implemented audio feedback because of limitations with audio cassette technology and that many of these could be overcome by using digital technologies. MP3 recorders could be used to record verbal feedback for students. Comments could then transferred to a computer and uploaded to a Virtual Learning Environment (VLE) system where students were able to access this feedback. Rotherham's work addressed many of the original impediments and was tested by students who received audio feedback [Rotherham, 2007].

Computer hardware.

A computer was a machine that manipulated data according to a set of instructions. One of the first

computers was the 'Electronic Numerical Integrator And Computer' (ENIAC) [Goldstine & Goldstine, 1946]. Originally, the ENIAC did not use stored programs. Instead, engineers programmed the ENIAC by changing electrical wiring between internal components. Vacuum tubes were used to implement computing logic. Computers such as the ENIAC began to be replaced in 1959 by a second generation of computers, which used transistors. A subsequent generation of computers that used Integrated Circuits (IC) and multi-processor, multi-programming technologies began to emerge in 1965. [Rosen, 1969] Microprocessors were ICs that contained an entire Central Processing Unit (CPU) of a computer in a single microchip. Microprocessors allowed computers to be smaller and faster, and led to the creation of the first Personal Computers (PC).

Originally, PCs were desktop machines. However, advancements in technology led to the creation of smaller portable hardware devices such as laptops, notebooks, and tablet PCs.

A tablet PC was a type of notebook computer. Tablet PCs relied on digital ink technology, using a digitiser to capture the movement of the special-purpose pen and record the movement on the LCD screen. A tablet PC is shown in figure 1.

Figure 1 – Photograph of a Philips Tablet PC
Reproduced from http://www.doctorsgadgets.com

Tablet PCs allowed software to be used whilst walking around in non-office based environments [Jarrett & Su, 2003].

The new systems described in this Dissertation contained specific functionality that allowed them to be used on tablet PC hardware.

Operating systems.

An Operating System (OS) was an interface between computer hardware and users. Operating systems were responsible for management and coordination of

activities and the sharing of a computer's resources. [Tanenbaum, 2001].

Three of the most popular systems are described.

Microsoft Windows: Microsoft Windows was an operating system for IBM PC's that provided a Graphical User Interface (GUI) and a multitasking environment. The first version was released in November 1985. Version 2 included overlapping windows and icons to represent programs and files. [Bellis, n.d.]. Windows Version 3 was released in 1990. This version included an improved program manager and icon system, a new file manager, support for sixteen colours, improved speed and reliability and extended memory addressing. Version 3 also provided a viable platform that allowed programmers to create Windows-compatible software. [Bellis, n.d.]. Windows 95 was released in August 1995 and was intended to be easier to use. It included an integrated TCP/IP stack, dial-up networking and long filename support. It was also the first version of Windows that did not require MS-DOS (MicroSoft Disk Operating System) to be installed. [Bellis, n.d.]. Windows 98 was released in June 1998. That version integrated a web browser and made navigation through the file system similar to navigating the WWW. [Bellis, n.d.] Windows NT 3.1 was released in 1994 and a subsequent version, Windows NT 4, was released in 1996. Windows NT was designed to run on multiple instruction set architectures and multiple hardware platforms within each architecture. Windows NT's core components were fundamentally different from previous versions of Windows. However, Windows NT 4's GUI was designed to match that of Windows 95. [Tanenbaum, 2001]. Windows NT 5.0 was released in 1998 and marketed as Windows 2000. Different editions were available for servers and workstation PCs [Bellis, n.d.]. This version also included a new file system that supported disk quotas and file-system-level encryption [Tanenbaum, 2001]. Windows XP was released in October 2001. Windows XP used the same core as Windows 2000 but provided a significantly redesigned GUI and enhanced multimedia and networking capabilities. [Barber *et al.*, 2001]. Windows XP Tablet PC Edition was an OS derived from Windows XP Professional and included utilities and basic drivers created specifically for tablet PC hardware. This OS required hardware to have a tablet digitizer or touch screen device, hardware control buttons, scrolling buttons, and at least one user-configurable application button. Windows XP Tablet PC Edition provided a Tablet PC Input Panel (TIP) that converted handwriting into text. Speech recognition functionality was also incorporated into the TIP. This allowed users to dictate text using speech in certain supported applications and control the GUI. A Tablet API that allowed programmers to access and manipulate low-level data captured by tablet PC hardware was also provided [Jarrett & Su, 2003]. Windows XP Tablet PC Edition was superseded by the Windows Vista OS, which improved support for tablet PC hardware. Windows Vista was released in January 2007. This version included new audio, print, display, and networking subsystems. Many changes to memory manager, process scheduler and I/O scheduler components were made to reduce security exploits [Hargreaves *et al.*, 2008]. A screenshot of Windows Vista is shown in figure 2.

The new systems described in this Dissertation were created for the Windows OS because this platform was used by many schools [Smith *et al.*, 2008] and provided APIs for audio and video capture and tablet PC hardware.

Unix & Linux: UNIX was created by programmers at Bell Laboratories in the early 1960s and proved to be a reliable OS. [Bell Labs, n.d.].
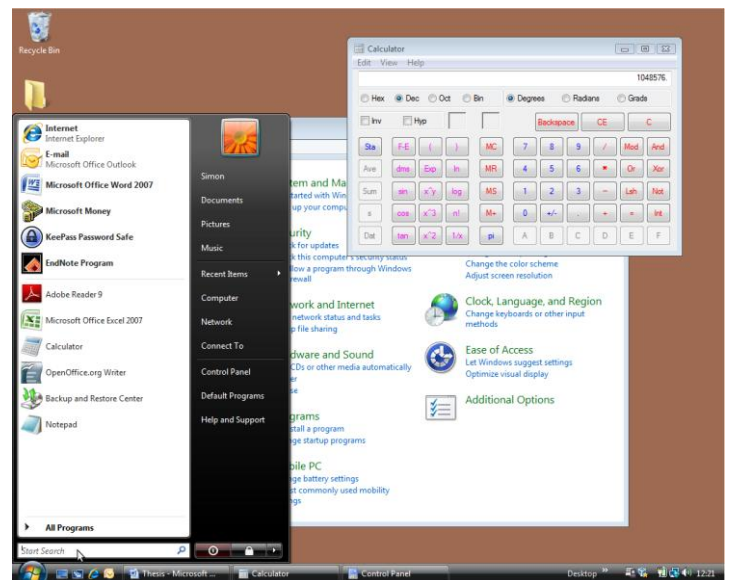

Figure 2 - Screenshot of Windows Vista

Linux was a generic term referring to a number of Unix-like operating systems based on a common kernel created in 1991 by Linus Torvalds. Linux was predominantly known for its use in servers, although it was installed on a wide variety of computer hardware, including embedded devices, mobile phones, PCs and supercomputers [Bovet & Cesati, 2000]. Many distributions included GUI's. Popular distributions included RedHat, SuSE and Ubuntu. A screenshot of Ubuntu Linux is shown in Figure 3.

Mac OS: Mac OS was a series of operating systems developed by Apple Incorporated for its Macintosh computers. 'Classic' Mac OS was released in 1984. This OS was characterized by its lack of written commands; it was a completely graphical OS. In 1999, Mac OS X was released. Unlike 'Classic' Mac OS,

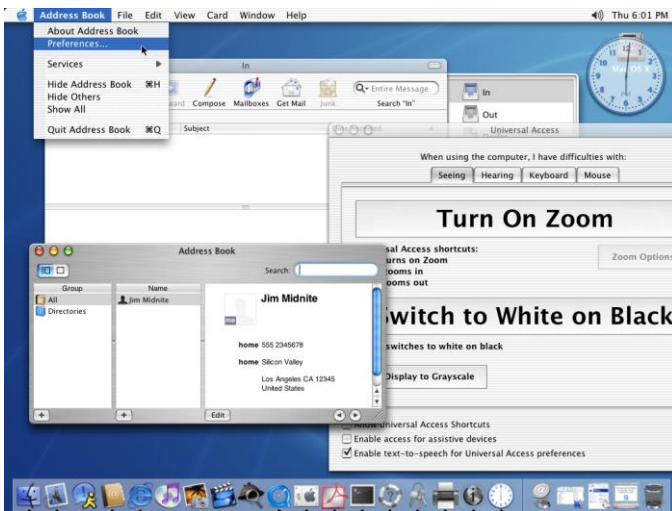this new version was a Unix-based operating system [Sanchez, 2000]. A screenshot of Mac OS X is shown in Figure 4.
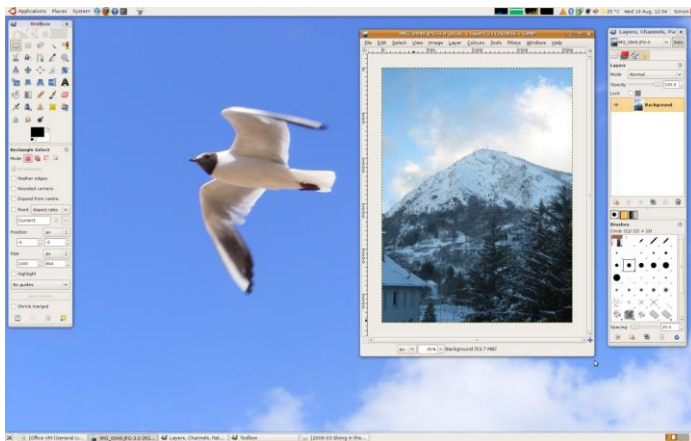

Figure 3 - Screenshot of Ubuntu Linux

Figure 4 - Screenshot of Mac OS X
reproduced from Wichary [n.d.]

Computer programming.

Computer programs were sets of instructions and rules that manipulated data. These instructions and rules were composed of sets of symbolic expressions, called statements, from programming languages. These statements prescribed tasks to be performed. Computational processes, in correctly working computers, executed programs precisely and accurately. Computational processes were used for intellectual work and answering questions. [Abelson et al., 1996] .

Programmers anticipated the behaviour and results of programs that they created. Programmers could make errors (called bugs or glitches) in programs that could have complex and unanticipated consequences. Programs could be structured so that unanticipated problems did not lead to catastrophic failures. When

problems did arise, programmers could debug their programs to correct errors. Computer programs were designed in a modular manner, so that individual parts could be constructed, replaced, and debugged separately. [Abelson *et al.*, 1996].

Structured Programming: Structured programs were hierarchical, nested structures of statements [Wirth, 1974]. The primary goals of structured programming were to minimize the number of errors that occurred during programming and reduce the effort required to correct errors in sections of deficient code or to upgrade sections of code when more reliable or efficient techniques were discovered [Jensen, 1981].

Object-oriented Programming (OOP): Object-Oriented Programming (OOP) was created to improve the design process and reusability of program code. Morris *et al.* [1999] identified the fundamental concepts of object-oriented programming as: Encapsulation, Inheritance and
Polymorphism.

- **Encapsulation**: OOP allowed tangible and conceptual entities in the problem domain to be represented as objects [Pressman, 2000]. Encapsulation was achieved by packaging relevant data and methods together as individual, identifiable objects. Methods typically accessed or manipulated objects' data. A class was a template that could be instantiated as a number of objects and was the basic element of OOP. Encapsulation aided program design by allowing details of an object's implementation to be hidden from other objects. [Armstrong, 2006].

- **Inheritance**: Inheritance in OOP allowed the definition and implementation of one class to be based on that of other existing classes. Inheritance allowed programmers to create 'child' classes that contained more specific instances of abstract concepts than classes at the top of a class hierarchy.

- **Polymorphism:** Polymorphism allowed different implementations to be hidden behind a common interface. This allowed methods with the same name from individual classes to respond differently. [Armstrong, 2006]. OOP was selected for this work because of its ability to encapsulate data and functions in re-usable components.

Component Technologies

Program functionality could be encapsulated in generic, reusable, self-contained packages called components. These components could be developed by different people, at different times, and possibly with different uses in mind [Madiajagan, 2006].

Object Linking and Embedding (OLE) was a software component technology developed by Microsoft. OLE was initially used for the creation and management of compound documents but its architecture was broadened to enable extended code reuse and to create a multi-purpose plug-in model that supported a wide range of component software. Microsoft introduced further technologies that built upon OLE architecture including Component Object Modelling (COM), Distributed COM (DCOM), shell extensions, and ActiveX technologies. [Brockschmidt, 1996] These OLE-based technologies were replaced by the Microsoft .NET platform [Richter, 2000].

Microsoft .NET was a new software development platform that provided a common OOP framework, replaced arcane Application Programming Interface (API) constructs, supported rapid application development and many database systems [Richter, 2002] and was platform independent [Easton & King, 2004]. Programmers were able to write programs using a number of high-level languages such as Visual Basic .NET and Visual C# [Richter, 2002].

Java was a high-level programming language and software development platform created by Sun Microsystems [Gosling & McGilton, 1996].

Microsoft .NET was used for this work because of its support for database systems and rapid application development.

Human-Computer Interaction

Human-Computer Interaction (HCI) was a discipline concerned with the study, design, construction and implementation of human-centric interactive computer systems. The purpose of HCI was to systematically apply knowledge about human purposes, human capabilities and limitations, and machine capabilities and limitations so as to extend the reach of users. Another goal was to enhance the quality of interaction between humans and computers. [Preece *et al.*, 1994]. Effective interface design was a multidisciplinary process requiring a holistic view of design problems. The capabilities and disciplines required to meet those goals included graphic and industrial design, an understanding of organisational dynamics and processes, an understanding of human cognitive, perceptual and motor skills, a knowledge of display technologies, input devices, interaction techniques and design methodologies, and an aptitude for elegance in system design [Baecker & Buxton, 1987].

A Graphical User Interface (GUI) was a means for a human to interact with a computer via a visual representation of data and processes. Use of graphics in user interfaces promoted the exploration and understanding of complex domains. GUIs could also

be referred to as WIMP because they contained Windows, Icons, Menus and Pointers [Hix & Hartson, 1993].

The components of WIMP GUIs were:

i. Windows: Different areas into which screens were divided. Users could run different programs or display different files, move windows around the display screen, and change their shape and size.
ii. Icons: Small pictures that represented commands, files, or windows. By moving a pointer to an icon and pressing a mouse button, users could execute commands or convert icons into windows. Users could also move icons around the display screen.
iii. Menus: Lists of options that users could select to execute commands.
iv. Pointer: A symbol that appeared on a display screen and was moved by users with a pointing device such as a mouse or trackball to select objects and commands.

Usability concerned the extent to which users were able to interact with computers to perform tasks successfully and without difficulty [Ravden & Johnson, 1989].

The new systems described in this Dissertation used WIMP and GUIs to represent data and processes visually.

The Electronic Student Assessment And data Management System (ESAAMS)
.

This Section describes the creation of a software system called an Electronic Student Assessment And data Management System (ESAAMS). A prototype system (ESAAMS Version 1) had been created in 2001 by Lassauniere & Tewkesbury. ESAAMS Version 2 was first mentioned in Lassauniere [2003]. ESAAMS allowed teachers to capture student work using various audio and video capture devices, such as video cameras and microphones, attached to a teacher's computer, and enabled quick and efficient management of student work.

KAAN & ESAAMS: Lassauniere [2003] described the creation of the Keyboard And Audio Network (KAAN) system.

KAAN was a hardware and software system that assisted music teachers in monitoring and supporting students and recording their work during lessons involving portable keyboards.

KAAN's hardware allowed audio to be flexibly routed around the classroom. Software was written to control the KAAN hardware system and incorporated an information system for administrative purposes. Audio

recordings could be associated with student records, for instant review. The KAAN system was marketed by a collaborating company called Counterpoint MTC Ltd. KAAN was expensive (installation costs for a single classroom were typically in excess of £10,000 GBP) and specific to music departments using portable keyboards. Mackrill [2004] recognised the potential for a system based on the same embedded information system that could be used in any teaching field. A review of assessment software was conducted by Mackrill at the collaborating company. Software systems were available for recording the results of students' summative assessments and attendance information [Capita Education Services, n.d.] but there were no software systems for capturing and storing their audio/visual work. Other, non-software based, methods of capturing and storing students' audio/visual work, such as audio cassettes and video tapes, were available and established in classroom practice [Ofsted, 2003]. Audio cassettes and videos were a useful and inexpensive recording medium. However, access to specific tracks was difficult, due to the time required to wind to the desired point in the tape and the inaccuracy of tape counters. The length of each track was often thirty seconds, or less. Therefore, it was almost impractical to replay specific recordings by individual students, or groups of students, in the classroom, over a period of time; even with detailed written index records. The problem was mitigated by the use of MiniDisc technology because of the digital nature of the medium and the associated ability to quickly locate specific tracks. However, the challenge of cataloguing disks and managing access to historic work remained. Mackrill [2004] suggested that to achieve effective assessment, mobile technology recording devices would be needed to collect and store this information for later use. This review led to the creation of a new system called Electronic Student Assessment And data Management System (ESAAMS). The KAAN software had allowed teachers to capture students work through KAAN hardware. ESAAMS allowed teachers to capture student work using various audio and video capture devices, such as video cameras and microphones, attached to the teacher's computer. This made ESAAMS suitable for a range of teaching fields, other than just music. Captured audio and video were stored as files. Associated information about student work, such as a title/description, grade and comments, was stored in a relational database for reporting purposes. Collectively, this information was referred to as a 'student work piece'. The new information system was teaching-centred and allowed quick and efficient management of student work pieces.

Validation of Concept and Initial Requirements

The concept of ESAAMS was presented to a music advisor. The concept was received with enthusiasm, so it was then decided to present it to a group of music teachers teaching at secondary level. Initial requirements for the solution were discussed and it was determined that:

- audio and video clips needed to be captured, stored and associated with students.
- user interfaces should be intuitive and suitable for teachers with basic IT experience.
- editable marking schemes should be included.
- the ability to export and import work was important.
- users should able to import student information from school management systems.
- the system should store data securely with a back-up facility.
- there should be a facility to generate reports on students' work.

[Mackrill, 2004]

Software Language, Database & Media Capture Technologies.

The programming platform selected was Visual Basic 6. This was because Visual Basic 6 was particularly suited to creating applications with extensive user interfaces and had some object-oriented abilities.
Microsoft Access 2000 was selected as the database because it was a relational database system that could be interfaced from within Visual Basic using Structured Query Language (SQL) to query the database.
A format and method for recording, playing-back and storing audio and video data needed to be identified. It was clear that data would need to be stored in a compressed format because it was likely that a large amount of data would be stored on teacher computers. Windows Media Player 9 and Windows Media Encoder 9 components were selected because they provided data compression, were compatible with the programming platform, and provided a common set of components for dealing with audio and video data.

Unified Modelling Language Analysis

Initial system analysis was conducted by Lassauniere at the collaborating company using Unified Modelling Language (UML). The Use Case (UC) diagram shown in Figure 5 was created to describe the high-level functional requirements of ESAAMS and its key actors.

It was intended that this model would be used as a basis for building test cases during functional testing and validation.

Data Storage: Three aspects of data storage were considered: **Error! Reference source not found.**,

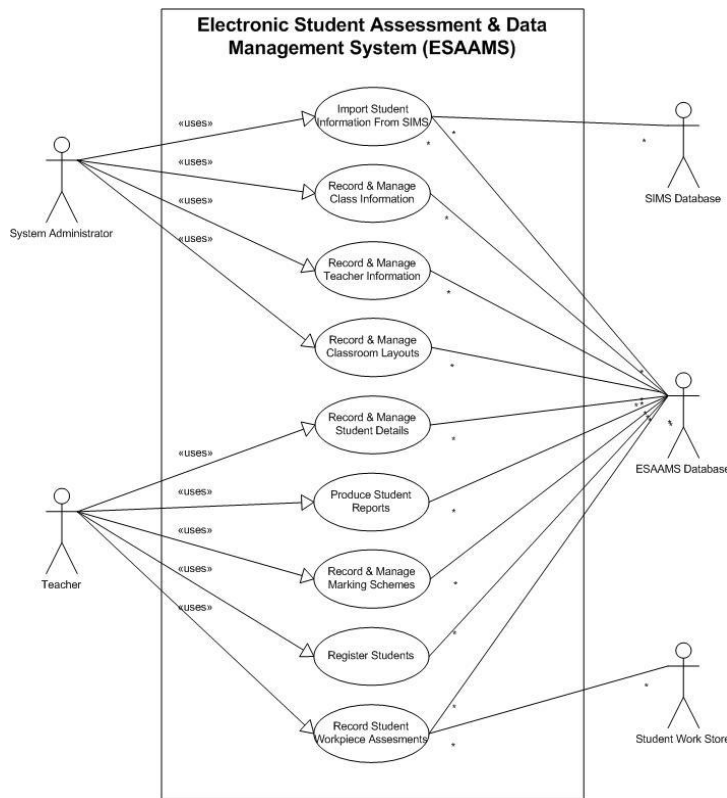Audio & Video Data Storage and Data Security and Privacy.

Figure 5 - ESAAMS Version 2 Use Case Diagram
(*produced by Lassauniere but never published*)

Audio & Video Data Storage: It was decided that audio and video data should be stored as separate files rather than as data embedded within the database because: manipulating audio and video data in the database would degrade performance, the selected media software components only supported this mode of storage and storing audio/video data as files provided a simpler method of importing and exporting student work pieces. A folder structure was considered to store the audio & video files. The first approach considered class centred management, where students' work was stored in different folders for each student class. It was found that this method was not appropriate because classes could be renamed and students could move between classes. A second approach was based on academic year folders in which work for all students was stored in a folder named with the academic year that the work was recorded. As soon as the program was used at or after the beginning of a new academic year, an 'academic year rollover' routine was triggered. This routine created a new academic year folder. The former year's database was copied and each class group promoted to the next year group. This method isolated old, unused data so that it could be archived.

Data Security and Privacy: The ESAAMS database contained personal information about students and needed to be secured to comply with the Data Protection Act [UK Government, 1998]. Therefore, each ESAAMS user had to log-in to the user interface using a username and password combination with a minimum password length of 5 characters. As the system was to be used in classrooms during lessons, and therefore often left unattended for short periods, teachers were able to quickly 'lock' the UI with a single mouse click. After locking the user interface, teachers were required to enter their password to resume using the software. A facility to change passwords regularly was provided in case the password became known. The facility to add additional users who did not have 'Administrator' rights was incorporated. This facility was intended to reduce the risk of accidental deletion or alteration of work piece and student information and other settings. The database contained sensitive information, such as students' personal details and teachers' passwords. To prevent unauthorised access to this data and to ensure the integrity of the database, an additional 'master' password was applied to the database file to prevent it being opened directly using the Microsoft Office Access application. The master password was a static shared secret between the database and the ESAAMS software. A backup facility was created by Lassauniere in case of hardware failure, data corruption, or accidental deletion. This was achieved by providing an option to perform an incremental backup of all files in the current academic year folder whenever the ESAAMS program was closed. Users could choose a local destination for backups on a disk connected to the teacher's computer. However, to achieve a higher level of protection, users were advised to choose a destination on a remote computer (through a mapped network drive) or an external backup device (such as a USB hard drive). A limitation of the incremental backup facility was that it did not perform a date comparison check on files. For each file in the source academic year folder, it merely checked if a file with the same name existed in the backup destination folder. Therefore, after a file had been backed-up once, subsequent modifications to the original file would not be backed-up as well. The exception to this rule was the database file, which was copied during each backup. To ensure a unique file name in the destination (a constraint of the operating system) the date of the backup was inserted in to the destination file's name.

Audio & Video Capture Hardware: Although the work described in this Chapter was primarily concerned with the creation of a new software system, it was necessary to ensure that compatible audio and

video capture devices existed and could be used with ESAAMS. It was important that the new system was able to capture data from a range of devices that were inexpensive and portable. The creation of the ESAAMS software coincided with rapid, widespread availability of microphones and USB web cams, which were supported by the selected media capture technology (Windows Media Encoder 9). There was large variation in the quality of the media that could be captured from these devices. The collaborating company (Counterpoint MTC Ltd) researched a number of microphones and web cams to advise users on the most suitable options.

Graphical User Interface

The principal areas and components of the Graphical User Interface (GUI) are outlined in this Section.

During the
Validation of Concept and Initial **Requirements**, it was determined that the GUI should be intuitive and suitable for teachers with basic Information Technology (IT) experience. In creating the GUI, the intention was that if a user was familiar with basic Windows applications, such as Microsoft Word, and knew how to use Microsoft Windows Explorer, then they should be able to operate the program. This was achieved by using standard Windows GUI elements where possible and splitting the main screen into an arrangement that was similar to other Windows applications. The main screen in the GUI was composed of a:

- Main menu, providing access to all functionality.

- Toolbar, providing quick access to frequently used functionality.

- Classroom Layout, representing the physical position of desks and placement of students in the classroom.

- Pop-out window, on the right of the screen.

- Status Bar.

These main areas of the GUI are outlined by a red border in The main menu was divided into seven sections. Some of these were only visible to Administrators or when a class was being displayed. The functions available from each main menu were:

File: Classes and an external database could be loaded or closed from this menu.

Edit: All students could be selected and individual student details displayed from this menu.

Class: How student names were displayed could be chosen. 'Lock Placing' and other placing options could be selected. Student work pieces could be added. Registration and reporting functionality could be accessed. Details of classes could also be found in this menu.

Figure 6.

The main menu was divided into seven sections. Some of these were only visible to Administrators or when a class was being displayed. The functions available from each main menu were:

File: Classes and an external database could be loaded or closed from this menu.

Edit: All students could be selected and individual student details displayed from this menu.

Class: How student names were displayed could be chosen. 'Lock Placing' and other placing options could be selected. Student work pieces could be added. Registration and reporting functionality could be accessed. Details of classes could also be found in this menu.
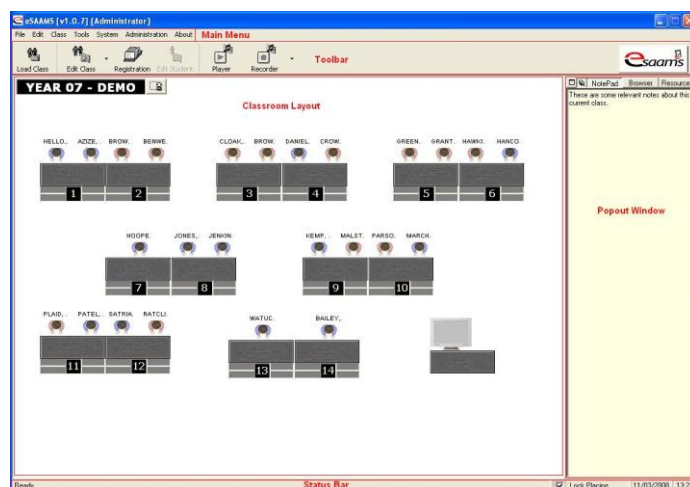

Figure 6 - Screenshot of ESAAMS Version 2 Main Window (created by Lassauniere)

Main Menu

Figure 7 shows the Main Menu from which most of the program's functions could be accessed.

Figure 7 - Screenshot of ESAAMS V2 Main Menu
(*created by Lassauniere*)

Tools: The player and recorder could be accessed from this menu.

System: This menu provided options for changing the password, arranging new Classroom Layouts and selecting various System Options.

Administration: New Classes and Marking Schemes could be added, deleted or edited. Users and Teachers could be edited and the Academic Year Folder location could be changed from this menu. This menu was only available when the user was logged-in to ESAAMS as the Administrator user.

About: Information about the software and contact details of the collaborating company.

Toolbar: The toolbar, shown in Figure 8, contained buttons and drop down menus and was divided into seven sections. However, not all of these were visible at the same time as they were dependent upon having classes loaded and the system environment.

Figure 8 - Screenshot of ESAAMS Version 2 Toolbar

The functions available from the toolbar were:

Load Class: Classes could be loaded and students belonging to the class were listed, together with any notes that have been recorded.

Edit Class: Students could be imported or removed. Registration and reporting functions could be accessed and Class Work pieces displayed. A drop-down menu also allowed access to Class Properties, Reporting and Work pieces.

Registration: Information about student attendance during lessons.

Edit Student: Students' personal and work piece information could be viewed and changed.

Player: Audio or video material could be selected and played.

Recorder: Audio, video or still images could be captured and stored.

Keyboard: Displayed the on-screen keyboard (Tablet PC only).

Classroom Layout: The representation of each desk displayed the number of the desk, the students placed at the desk (none, one or two per desk) and their names. Pale blue coloured arms indicated a boy, pale pink a girl and grey if no gender had been assigned to a

student. Figure 9 shows a screenshot of two female students placed at desk number 9.

Figure 9 - Screenshot of two female students placed at a desk

A number of main menu options were also accessible by right-clicking on objects in the Classroom Layout. The menu displayed was context sensitive, changing with the type of object selected at the time. Additionally, student record cards could be opened by selecting a student icon in the Classroom Layout.

Pop-out Window: The pop-out window was displayed on the right-hand side of the main window. It contained three tabs: Notepad, Browser and Resources. Figure shows these tabs at the top of the pop-out window.
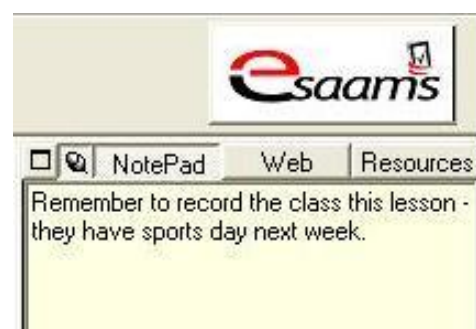

Figure 10 - Screenshot of Top Section of ESAAMS Version 2 Pop-out Window

The Notepad was a quickly accessible area for teachers to record brief comments, observations or reminders. The Notepad was cleared every time a class group was loaded. However, before clearing the notepad, the user was provided with the option to save the notes to the database. If the user chose to do so, the notes were associated with the loaded class and could be viewed through a Class Manager interface. The Browser tab contained a web browser. When this tab was selected, the pop-out window expanded to fill the whole of the main window. The web browser allowed teachers to collect resources from the web that could support their teaching.

The Resources tab allowed teachers to store and manage relevant, supporting, audio and video media

clips. This functionality is described in Appendix B. By default, the pop-out window was hidden and was only displayed when the user moved the mouse cursor to the extreme right-hand side of the main window. When the user moved the mouse cursor outside of the pop-out window area, the window collapsed again, out of sight.

If the user wished the pop-out window to remain open, they were able to 'pin' the window by toggling the pin icon button in the top left of the window. When pinned, the pop-out window would remain open, even when the mouse cursor was moved outside of the pop-out window area.

Status Bar: The status bar was used to display information about the current state of the ESAAMS program in an unobtrusive way. For example: when a class was been loaded, information about the progress of the operation was displayed in the status bar. There was a small check box in the left corner of the Status Bar (Figure 11) that locked the placement of students to prevent accidental misplacement of students from their assigned desks.
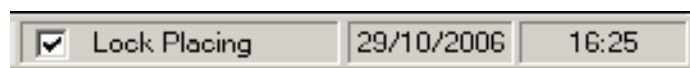

Figure 11 - Screenshot of ESAAMS Version 2 Status Bar

Tablet PC Features

Tablet PCs were computers that allowed teachers to use software whilst walking around classrooms and other teaching environments. If ESAAMS detected that it was running on a tablet PC then some aspects of its GUI were adapted to take advantage of the input methods.

On tablet PCs, the toolbar contained an additional 'On-Screen Keyboard' icon. Teachers could click this button to open an on screen keyboard that allowed text to be input letter-by-letter by tapping the relevant buttons on the screen with the stylus.

A number of windows allowed text to be entered using handwriting recognition. A small button was added to the left of each text entry field that supported handwriting recognition. Upon clicking one of these buttons, users were presented with a resizable popup window (Figure 12). Users could write in to this area. After writing and pausing for a few seconds, the handwriting was automatically converted to printed text. The font for printed text could be changed by clicking on the 'Settings' icon (depicted as a spanner). When handwriting entry was completed, users could click the 'Done' button to copy the entered text into the field and close the recognition popup window.

User Support

In order to support users, a manual for the system was produced and made available to users in portable document format (PDF). The manual and a troubleshooting guide were included on the program CD distributed to schools.

Additionally, in January 2004, a series of animated tutorials were produced to demonstrate all the basic functions of the program. These were distributed on the program CD and on a website. [Mackrill, 2004] Telephone and email support was available. A bug-tracking system was set-up to record and prioritise new bugs and enhancement requests from users.

Figure 12 - Screenshot of handwriting text entry in ESAAMS Version 2

Discussion

Research into electronic assessment systems has been undertaken and a new electronic assessment system has been created. The research work brought the following successes: existing electronic assessment systems were investigated and gaps identified and a new electronic assessment system (called ESAAMs) was created.

This paper reviewed the background technologies, systems and models used in this research and described the context of this research and the existing framework for educational instruction and assessment in the UK. Technologies used during the research were discussed. Finally a software system called an ESAAMS Version 2 was presented. ESAAMS Version 2 allowed teachers to capture student work using various audio

and video capture devices, such as video cameras and microphones, attached to a teacher's computer, and enabled quick and efficient management of student work.

References

Abate, M. (n.d.). Assessment In Health Sciences Education. Retrieved 15 June 2009, from http://www.hsc.wvu.edu/admin/facultydev

Abelson, H., Sussman, G. J., & Sussman, J. (1996). Structure and Interpretation of Computer Programs (Second edition ed.): MIT Press.

Armstrong, D. J. (2006). The quarks of object-oriented development. Communications of the ACM, 49(2), 123-128.

ASAP. (n.d.). from http://asap.convoq.com/

Baecker, R., & Buxton, W. (1987). Readings in human-computer interaction: A multidisciplinary approach.

Barber, B., Todd, C., Johnson Jr, N. L., & Shimonski, R. (2001). Configuring & Troubleshooting Windows XP Professional. In M. Grasdal (Eds.), Available from http://site.ebrary.com/lib/portsmouth

Barrett, H. C. (1998). Strategic Questions: What to Consider When Planning For Electronic Portfolios. Learning & Leading with Technology, 26(2), 6-13.

Becta. (2007). The impact of e-portfolios on learning. Coventry, UK: Becta.

Becta. (2008). Personalising learning in a connected world: A guide for school leaders.

Beetham, H. (2008). e-portfolios in post-16 learning in the UK: developments, issues and opportunities.

Bell Labs. (n.d.). The Creation of the UNIX* Operating System. Retrieved 17 June 2009, from http://www.bell-labs.com/history/unix/

Bellis, M. (n.d.). About.com. Journal. Retrieved from http://inventors.about.com/od/mstartinventions/a

Berners-Lee, T., Cailliau, R., Luotonen, A., Nielsen, H. F., & Secret, A. (1994). The World-Wide Web. Communications of the ACM, 37(8), 76-82.

BESA. (2006). Information and Communication Technology in UK State Schools - Summary Report. London, UK.

Booch, G., Jacobson, I., & Rumbaugh, J. (2000). OMG Unified Modelling Language (UML) Specification (Version 1.3): Object Management Group.

Bormann Young, C., & Henquinet, J. A. (2000). A Conceptual Framework for Designing Group Projects. Journal of Education for Business, 76(1), 56-60.

Boud, D. (1989). The Role Of Self-Assessment In Student Grading. Assessment & Evaluation in Higher Education, 14(1), 20 - 30.

Bourque, L. B., & Fielder, E. P. (1995). How to conduct self-administered and mail surveys. London: Thousand Oaks.

Bovet, D. P., & Cesati, M. (2000). Understanding the Linux Kernel: O'Reilly.

Brockschmidt, K. (1996). How OLE and COM Solve the Problems of Component Software Design. Microsoft Systems Journal, 11(5).

Brookhart, S. M., & Durkin, D. T. (2003). Classroom Assessment, Student Motivation, and Achievement in High School Social Studies Classes. Applied Measurement in Education, 16(1), 27 - 54.

Cambridge, D. (2008). Audience, integrity, and the living document: eFolio Minnesota and lifelong and lifewide learning with ePortfolios. Computers & Education, 51(3), 1227-1246.

Capita Education Services. (n.d.). SIMS - Schools Information Management System. Retrieved 29th April, 2008, from http://www.capitaes.co.uk/SIMS/

Carlbring, P., Brunt, S., Bohman, S., Austin, D., Richards, J., Öst, L.-G., et al. (2007). Internet vs. paper and pencil administration of questionnaires commonly used in panic/agoraphobia research. Computers in Human Behavior, 23(3), 1421-1434.

Chang, T.-Y., & Chen, Y.-T. (2009). Cooperative learning in E-learning: A peer assessment of student-centered using consistent fuzzy preference. Expert Systems with Applications, 36(4), 8342-8349.

Chapple, M. (n.d.). What is a database? Journal. Retrieved from http://databases.about.com/od

Chen, G., Chen-Chung, L., Kuo-Liang, O., & Ming-Song, L. (2001). Web Learning Portfolios: A Tool for Supporting Performance Awareness. Innovations in Education and Teaching International, 38(1), 19-30.

Chen, H., Yu, C., & Chang, C. (2007). E-Homebook System: A web-based interactive education interface. Computers & Education, 49(2), 160-175.

Chen, P. P.-S. (1976). The entity-relationship model - toward a unified view of data. ACM Transactions on Database Systems (TODS), 1(1), 9-36.

Chester, S. (2000). Introduction To Relational Databases. Retrieved 16 September 2009, from http://www.simonchester.com/computing/notes

Cloete, E. (2001). Electronic education system model. Computers & Education, 36(2), 171-182.

Counterpoint MTC Ltd. (2006a). ESAAMS Version 4 Network Edition User Manual.

Counterpoint MTC Ltd. (2006b). ESAAMS Version 4 Standalone Edition User Manual.

Crystal Decisions. (n.d.).

Department for Children, Schools and Families (DCSF), UK Government (2005, 22nd August 2005). Secondary school. Retrieved 30th September, 2008, from http://www.everychildmatters.gov.uk/ete/agencies

Department for Children, Schools and Families (DCSF),, UK Government, (2008). Personalised Learning – A Practical Guide. Journal. Retrieved from http://publications.teachernet.gov.uk/eOrderingDownload/00844-2008DOM-EN.pdf

Department for Children, Schools and Families (DCSF), UK Government. (n.d.). Online Help: Glossary. Retrieved 8th October, 2008, from

Country Code 21.

http://www.standards.dfes.gov.uk/onlinehelp
DfES. (2005). Harnessing Technology: Transforming Learning and Children's Services (No. 1296-2005DOC-EN). Nottingham.

Dochy, F., Segers, M., & Sluijsmans, D. (1999). The use of self-, peer and co-assessment in higher education: A review. Studies in Higher Education, 24(3), 331 - 350.

Dominick, P. G., Reily, R. R., McGourty, J. W. (1997). The Effects of Peer Feedback on Team Member Behaviour. Group Organization Management, 22(4), 508-520.

Duffy, M. L., Jones, J., Thomas, S. W. (1999). Using portfolios to foster independent thinking. Intervention in School and Clinic, 35(1), 34-37.

Easton, M. J., & King, J. (2004). Cross-Platform .NET Development. Berkeley, California: Apress.

Edwards, P., Roberts, I., Sandercock, P., & Frost, C. (2004). Follow-up by mail in clinical trials: does questionnaire length matter? Controlled Clinical Trials, 25(1), 31-52.

Eisenbach, S., Sadler, C., & Jurisic, V. (2002). Feeling the way through DLL Hell. Paper presented at the The First Workshop on Unanticipated Software Evolution (USE 2002), Málaga, Spain.

Falchikov, N. (1995). Peer Feedback Marking: Developing Peer Assessment. Innovations in Education and Teaching International, 32(2), 175 - 187.

Frederiksen, J. R. C., A. (1989). A systems approach to educational testing. Ed Researcher, 18(9), 27-32.

Gane, C. P., & Sarson, T. (1979). Structured Systems Analysis: Tools and Techniques: Prentice-Hall.

Goldstine, H. H., & Goldstine, A. (1946). The Electronic Numerical Integrator and Computer (ENIAC). Mathematical Tables and Other Aids to Computation, 2(15), 97-110.

Gosling, J., & McGilton, H. (1996). The Java™ Language Environment. Mountain View, California: Sun Microsystems, Inc.

Graefe, G. (1996). The Microsoft relational engine. Paper presented at the Data Engineering, 1996. Proceedings of the Twelfth International Conference on, New Orleans, LA, USA.

Halsall, F. (1995). Data Communications, Computer Networks and Open Systems (Fourth Edition ed.): Addison-Wesley.

Hargreaves, C., Chivers, H., & Titheridge, D. (2008). Windows Vista and digital investigations. Digital Investigation, 5(1-2), 34-48.

Haught, D. (n.d.). Strategic Initiatives: Evolving Microsoft Access Applications to Microsoft SQL Server. Journal. Retrieved from http://www.fmsinc.com/FMSUpsize/docs/EvolvingMicrosoftAccessApplications.pdf

Hildreth, S. (2004). Software QA 101: The Basics of Testing. Journal. Retrieved from http://www.informit.com/articles

Hix, D., & Hartson, H. R. (1993). Developing User Interfaces: Ensuring Usability Through Product & Process: Johm Wiley & Sons.

Howard, Y. (2007). FREMA: An e-Learning Framework Reference Model for Assessment.

IBM. (1970). Flowcharting Techniques (No. C20-8152-1): IBM.

IEEE Standard Glossary of Software Engineering Terminology, (1990).

Jarrett, R., & Su, P. (2003). Building Tablet PC Applications. Redmond, Washington: Microsoft Press.

Jensen, R. W. (1981). Structured Programming. Computer, 14(3), 31-48.

Jordan, S., & Jackson, A. (2006). LogMeIn and IPSec VPNs: a Comparison: LogMeIn Inc.

Kaner, C. (1993). Testing Computer Software (Second ed.): Van Nostrand Reinhold.

Kench, P. L., Field, N., Agudera, M., & Gill, M. (2009). Peer assessment of individual contributions to a group project: Student perceptions. Radiography, 15(2), 158-165.

Lamers, S., Beason, S., Dunlap, L., Compton, R., & Salemi, M. (2004). HIVbase: a PC/Windows-based software offering storage and querying power for locally held HIV-1 genetic, experimental and clinical data. Bioinformatics, 20(3), 436-438.

Larman, C. (2002). Applying UML and Patterns. Upper Saddle River: Prentice-Hall.

Lassauniere, A. (2003). New music technology systems and methods to assist teachers. University of Portsmouth, Portsmouth.

Lejk, M., & Wyvill, M. (2001). The Effect of the Inclusion of Self assessment with Peer Assessment of Contributions to a Group Project: a quantitative study of secret and agreed assessments. Assessment & Evaluation in Higher Education, 26(6), 551 - 561.

Loney, K. (2004). Oracle Database 10g: The Complete Reference. Available from http://citeseerx.ist.psu.edu

Mac Iver D J & Reuman D A (1994). Giving Their Best: Grading & Recognition Practices That Motivate Students To Work Hard. American Ed 17(4), 24-31.

Mackrill, D. (2004). Using technology to improve teaching and learning: an intervention in music assessment. Unpublished Dissertation, Uni of Sussex.

Madiajagan, M. V., B. (2006). Interoperability in Component Based Software Development. Proceedings of World Academy of Science, Engineering & Technology, 16, 207-215.

Marshall, G. (2005). The purpose, design and administration of a questionnaire for data collection. Radiography, 11(2), 131-136.

McGourty, J. (2000). Using Multisource Feedback in the Classroom: A Computer-Based Approach. IEEE Transactions on Education, 43(2), 120-124.

Microsoft Developer Network. (2008). Merge Modules (Windows). Retrieved 27th August, 2008, from http://msdn.microsoft.com

Morris, M. G., Speier, C., & Hoffer, J. A. (1999). An Examination of Procedural and Object-oriented

Systems Analysis Methods: Does Prior Experience Help or Hinder Performance?*. Decision Sciences, 30(1), 107-136.

Moulder, R. (1973). An implementation of a data management system on an associative processor. Paper presented at the Proceedings of the June 4-8, 1973, national computer conference and exposition. Retrieved 12 June 2009, from http://portal.acm.org

Mowl, G., & Pain, R. (1995). Using Self and Peer Assessment to Improve Students' Essay Writing: a Case Study from Geography. Innovations in Education and Teaching International, 32(4), 324 - 335.

National Foundation for Educational Research. (n.d.). National Curriculum Levels.   Retrieved 8th October, 2008, from http://www.nfer.ac.uk/research-areas

National Stone Centre. (2004). The Educational Use of Aggregate Sites. Derbyshire: National Stone Centre.

Novak, J. D., & Cañas, A. J. (2006). The Theory Underlying Concept Maps & How to Construct &Use Them: Florida Inst' for Human & Machine Cognition.

Oakley, B., Felder, R. M., Brent, R., & Elhajj, I. (2004). Turning Student Groups into Effective Teams. Journal of Student Centered Learning, 2 (1).

Ofsted. (2002). Progress File: an evaluation of demonstration projects in schools. London: Office for Standards in Education (Ofsted).

Ofsted. (2003). Good assessment practice in music. Retrieved   29th   April   2008.   from http://www.ofsted.gov.uk/publications

Patton, R. (2001). Software Testing. Indianapolis: Sams Publishing.

Petri, G. (2005). A Comparison of Oracle and MySQL. SELECT Journal, 12(1), 41-48.

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., & Carey, T. (1994). Human-computer interaction. Wokingham: Addison-Wesley.

Pressman, R. S. (2000). Software Engineering: A Practitioner's Approach (Fifth Edition ed.). Maidenhead, UK: McGraw-Hill Publishing Company.

Ravden, S. J., & Johnson, G. I. (1989). Evaluating usability of human-computer interfaces: a practical method. Chichester: Ellis Horwood.

Richter, J. (2000). Microsoft .NET Framework Delivers the Platform for an Integrated, Service-Oriented Web. Microsoft Developer Network (MSDN) Magazine   Retrieved 30 June 2009, from http://msdn.microsoft.com/en-us/magazine

Richter, J. (2002). Applied Microsoft .NET Framework Programming. Redmond, Washington: Microsoft Press.

Robertson, J., & Robertson, S. (1998). Complete Systems Analysis. New York: Dorset House.

Rosen, S. (1969). Electronic Computers: A Historical Survey. ACM Computing Surveys (CSUR), 1(1), 7-36.

Rotherham, B. (2007). Using an MP3 recorder to give feedback on student assignments. Educational Developments, 8 (2).

Rust, C. (2001). A Briefing on Assessment of Large Groups. York, UK: Learning and Teaching Support Network (LTSN).

Sanchez, W. (2000). The Challenges of Integrating the Unix and Mac OS Environments. Paper presented at the USENIX 2000. Retrieved 30 June 2009, from http://www.usenix.org/event/usenix2000/invitedtalks

Sargent, L. (2000). Middle School Assessment Model. Retrieved 5th June 09 from http://www.smplanet.com

Shelley, J. (1996). The Internet and World Wide Web explained. London: Bernard Babani.

Shepard, L. A. (2000). The Role of Classroom Assessment   in   Teaching   and   Learning: CRESST/University of Colorado at Boulder.

Silvert, W. (2001). Modelling As A Discipline. Int' Jrnl of General Systems, 30(3), 261 - 282.

Smilowitz, E. D., Darnell, M. J., & Benson, A. E. (1994). Are we overlooking some usability testing methods? A comparison of lab, beta, and forum tests. Behaviour & Info' Technology, 13(1 & 2), 183-190.

Smith, P., Rudd, P., & Coghlan, M. (2008). Harnessing Technology: Schools   Survey   2008:   National Foundation for Educational Research.

Stiggins, R. J. (1994). Student-cantered classroom assessment. New York: Merrill.

Tanenbaum, A. S. (2001). Modern Operating Systems (2nd ed.). Upper Saddle River, New Jersey: Prentice-Hall.

Tomlinson, S. (2005). Education in a Post-Welfare Society. Maidenhead, England: Open University Press.

Topping, K. J. (1998). Peer assessment between students in college and university. Review of Educational Research, 68(3), 249-267.

Topping, K. J., Smith, E. F., Swanson, I., & Elliot, A. (2000). Formative Peer Assessment of Academic Writing Between Postgraduate Students. Assessment & Evaluation in Higher Education, 25(2), 149 - 169.

Trowler, P. (2003). Education Policy. New York: Routledge.

Tufte, E. R. (1992). The Visual Display of Quantitative Information: Graphics Press.

Uebersax, J. S. (2006). Likert scales: dispelling the confusion.   Retrieved   1st August,   2008,   from http://ourworld.compuserve.com/homepages

UK Government. (1988). Education Reform Act 1988. Retrieved 4th June 09 from http://www.opsi.gov.uk

UK Government. (1998). Data Protection Act 1998 Chapter 29. Retrieved 29th April 2008. from http://www.opsi.gov.uk/acts/acts1998

University of Portsmouth. (2007). Ethics Policy: University of Portsmouth.

University of Portsmouth Faculty of Technology Research Ethics Committee. (2006). Research Ethics Guidance for All Staff & Students: University of Portsmouth.

Van Duzer, E., & McMartin, F. (2000). Methods to Improve the Validity and Sensitivity of a Self/Peer Assessment Instrument. IEEE Transactions on Education, 43(2), 153-158.

Vera, E., Arias, L. D., & Buitrago, Z. (2001). Building Up Learners' Autonomy and Cooperative Assessment. Folios, 14.

Vieira, R. (2003). Professional SQL Server 2000 Programming. Indianapolis: Wiley Publishing.

Whalley, B., & Trew, S. (1998). Internet provision for staff and students: an information-rich approach. Computers & Geosciences, 24(7), 659-663.

Wichary, M. (n.d.). Screenshots from Mac OS X 10.2 Jaguar. Retrieved 19th August 2009, from http://www.guidebookgallery.org/screenshots

Wikipedia. (2008). Windows Installer. Retrieved 27th August, 2008, from http://en.wikipedia.org/wiki

Wiliam, D., & Black, P. (1996). Meanings and Consequences: a basis for distinguishing formative and summative functions of assessment? British Educational Research Journal, 22(5), 537 - 548.

Wininger, S. R. (2005). Using Your Tests to Teach: Formative Summative Assessment. Teaching of Psychology, 32(3), 164-166.

Wirth, N. (1974). On the Composition of Well-Structured Programs. ACM Computing Surveys (CSUR), 6(4), 247-259.