
**THE DEVELOPMENT OF A MODEL TO
EFFECTIVELY UTILISE COMPUTER
MEDIATED COMMUNICATION TO SUPPORT
ASSESSMENT IN A VIRTUAL LEARNING
ENVIRONMENT**

by

Grant Pullen

Submitted in compliance with the requirements for the

**MASTER'S DEGREE IN TECHNOLOGY:
INFORMATION TECHNOLOGY**

in the Faculty of

COMPUTER STUDIES

at the

PORT ELIZABETH TECHNIKON

PROMOTER: Mrs Theda Thomas

CO-PROMOTER: Mr Rudi Harmse

March 2001

Declaration

I, Grant Pullen, hereby declare that:

- i. the work in this dissertation is my own original work;
- ii. all resources used or referred to have been documented and recognised; and
- iii. this dissertation has not been previously submitted in full or partial fulfilment of the requirements for an equivalent or higher qualification at any other recognised educational institution.

.....
SIGNATURE

March 2001
DATE

I would like to express my thanks to:

My promoter Theda Thomas, for her guidance and dedication that went beyond the call of duty. Rudi Harmse, my co-promoter who made the evaluation of the system possible.

My mother, grandmother and family, for their assistance and encouragement. Gadget and Ebony, my four-legged companions, who lifted my spirit when it was down.

Abstract

Programming lecturers are faced with logistical problems associated with evaluating students' assignments. This dissertation will discuss the problems associated with paper- and electronic-based submission and evaluation systems. Managing student assignments is thus an important issue and this dissertation investigates how the computer can be used to help in both the management of programming assignments and giving of feedback for those assignments.

Computer Mediated Communication (CMC) technologies offer a technical solution for the submission of assignments. CMC technologies are therefore studied and their use in the educational sphere discussed. Information Security is another important feature that is needed when using CMC for assignment management.

A guideline showing the different features that an assignment management and feedback system would need was developed. Current systems that offer some assignment management features were studied and evaluated according to the guideline.

A model was developed for the management of programming assignments. The model addresses the problems of paper-based and previous electronic submission systems and includes the assessment of the assignments. The viability of the model was determined by building and testing a prototype based on the model's specifications. The prototype was evaluated within a second year Visual Basic programming class. The evaluation determined that there were areas that needed improvement, however the feedback was generally positive.

Table of Contents

List of Figures	viii
List of Tables	xi
Chapter 1 - Introduction	1
1.1 Background	1
1.2 Problem statement	3
1.3 Objectives	4
1.4 Methodology	4
1.5 Organisation of the dissertation	5
Chapter 2 – Computer Mediated Communication technologies within educational environments	8
2.1 Introduction.....	8
2.2 Defining CMC.....	8
2.3 CMC systems	9
2.3.1 E-mail.....	10
2.3.2 MUDs and MOOs.....	10
2.3.3 Word Wide Web (WWW).....	11
2.3.4 Audio conferencing.....	12
2.3.5 Video conferencing.....	12
2.3.6 Newsgroups.....	12
2.3.7 Internet Relay Chat (IRC)	13
2.3.8 Whiteboard	13

2.4 Media usage of CMC.....	14
2.5 Traits of CMC systems	15
2.5.1 Synchronous and asynchronous nature.....	15
2.5.2 Time and distance constraints	16
2.5.3 Cost saving	16
2.5.4 Personal development	16
2.5.5 Textual nature of CMC	17
2.6 Groupware	17
2.7 Workflow	19
2.8 From paper to electronic assignments	21
2.9 Summary.....	22

Chapter 3 - The importance and application of information

security	23
3.1 Introduction.....	23
3.2 A distributed security architecture	24
3.3 Trust.....	26
3.3.1 Security	27
3.3.2 Availability.....	29
3.3.3 Performance.....	30
3.4 The use of cryptographic technologies to ensure a secure environment.	31
3.4.1 What is cryptography?	31
3.4.2 Factors influencing the strength of cryptographic systems.....	33
3.4.3 Encryption and decryption algorithms	34
3.4.4 Hash functions	36
3.5 Digital Signatures	37
3.6 Security protocols for the Internet.....	38
3.6.1 Secure HTTP	39
3.6.2 Secure Sockets Layer (SSL).....	39

3.6.3 Kerberos	40
3.7 E-mail Security considerations	40
3.7.1 Privacy-Enhanced Mail (PEM)	42
3.7.2 Pretty Good Privacy (PGP) and OpenPGP	44
3.7.3 Secure Multipurpose Internet Mail Extensions (S/MIME).....	46
3.8 Summary.....	46

Chapter 4 – Requirements of an assignment management system..... 48

4.1 Introduction.....	48
4.2 Aspects to consider when choosing a solution	48
4.2.1 Platforms	49
4.2.2 Standards and architecture	49
4.2.3 Integrated or specialised products	49
4.2.4 User- friendliness.....	50
4.2.5 Training.....	50
4.2.6 Cost	50
4.2.7 Deployment and Fit	51
4.2.8 Stability	51
4.2.9 Vendor support and strength	51
4.2.10 Security	52
4.3 Features required by a computer-supported assignment feedback system.....	53
4.3.1 Communication	55
4.3.2 Submission	56
4.3.3 Storage	57
4.3.4 Evaluation.....	58
4.3.5 Feedback and retrieval	59
4.3.6 Administrative features	59
4.4 Summary.....	60

Chapter 5 - Evaluation of currently available products 62

5.1 Introduction.....	62
5.2 The use of class management software to provide assignment management.....	63
5.2.1 WebCT.....	63
5.2.2 Web Course in a Box (WCB).....	64
5.2.3 TopClass.....	65
5.2.4 Lotus Learning Space	67
5.3 Comparison of the integrated solutions	69
5.4 Stand-alone evaluation software	71
5.4.1 Microsoft Word 97 and Word 2000.....	72
5.4.2 Markin.....	73
5.4.3 Mindtrail.....	75
5.5 Comparison of the assignment evaluation solutions	78
5.6 Summary.....	79

**Chapter 6 - The development of a model to support the
management of assignments 80**

6.1 Introduction.....	80
6.2 A Programming Assignment Management Model (PAMM)	81
6.2.1 Management component	83
6.2.2 Communication component	83
6.2.3 Submission component	84
6.2.4 Storage component	87
6.2.5 Evaluation component	88
6.2.6 Feedback and retrieval component	90
6.3 Summary.....	91

**Chapter 7 - The implementation of a web-based
assignment management model..... 93**

7.1 Introduction.....	93
-----------------------	----

7.2 The physical components of Assignment Manager 2000.....	93
7.2.1 Web Browser.....	95
7.2.2 Web Server.....	95
7.2.3 Database.....	95
7.3 Management features.....	98
7.3.1 Access to the system.....	98
7.3.2 Assignment creation.....	99
7.4 Student features.....	100
7.4.1 Assignment submission.....	100
7.4.2 Viewing evaluated assignments.....	104
7.5 Lecturer features.....	105
7.5.1 Lecturer retrieval of submitted assignment.....	105
7.5.2 The Evaluation of student assignments.....	109
7.5.3 Annotation.....	110
7.5.4 Error Types.....	110
7.5.5 Historical assignment information.....	111
7.5.6 Packaging evaluated assignments.....	112
7.5.7 Evaluation limitations.....	112
7.5.8 Lecturer evaluated assignment submission.....	113
7.6 Summary.....	113
Chapter 8 - Evaluation & Conclusion.....	114
8.1 Introduction.....	114
8.2 Method of the evaluation.....	115
8.3 Results of the questionnaire.....	115
8.3.1 “Did you have any difficulties logging onto the system?”.....	115
8.3.2 “How easy or hard was it to learn how to use the system?”.....	116
8.3.3 “How would you compare electronic to paper-based assignment submission”.....	117
8.3.4 “How do you feel about receiving evaluated assignments electronically compared to paper?”.....	118

8.3.5 “What did you like the most about the system?”	119
8.3.6 “What did you dislike the most about the system?”	119
8.3.7 “What changes should be made to improve the system?”	119
8.3.8 “Do you see a potential use for electronic-based assignment submission?”	119
8.4 Evaluation by the lecturer	120
8.5 Meeting the requirements	121
8.5.1 Multiple file attachments.....	122
8.5.2 Security	123
8.5.3 Assignment annotation.....	124
8.5.4 Assignment storage.....	125
8.5.5 Conclusion.....	125
8.6 Future Research and Development	125
8.6.1 Integrated mark-sheets	126
8.6.2 Online evaluation.....	126
8.6.3 Plagiarism.....	126
8.6.4 Graphics capabilities.....	127
8.6.5 Trend analysis.....	127
8.6.6 Standardisation of file formats	128
8.6.7 File Uploading.....	128
8.6.8 Non-repudiation.....	128
8.6.9 Extending the user base.....	129
8.6.10 Extra features.....	129
8.7 Summary.....	130
References.....	132
Appendix A - Electronic Assignment Submission and Retrieval Questionnaire.....	144

Appendix B - User Manual: Assignment Manager 2000	146
B.1 Introduction	147
B.2 Installation and configuration of Apache and PHP	147
B.3 Login	148
B.4 The Administrative Functions Of The Administrator.....	149
B.5 The Lecturer	151
B.6 The Student	157
B.7 The Evaluation Application	160
B.7.1 Installation and Configuration	160
B.7.2 Lecturers view of the application.....	160

List of Figures

Chapter 1 - Introduction	1
1.1 An example of how comments are made on paper	2
Chapter 3 - The importance and application of information security	23
3.1 Bruce and Dempsey's (1997) distributed security architecture	25
3.2 The process of encryption and decryption of a message	31
3.3 Secret-key (symmetric) based encryption and decryption.....	32
3.4 Public-key (asymmetric) based encryption and decryption.....	32
3.5 Displays the hierarchical tree structure that PEM uses	44
3.6 Illustrates how a PGP message is encrypted and decrypted	45
Chapter 4 – Requirements of an assignment management system.....	48
4.1 Displays the process of assignment management	54
Chapter 5 - Evaluation of currently available products	61
5.1 Assignment upload form.....	64
5.2 Displays the uploaded assignment file details	64
5.3 LearningSpace's assignment file submission form.....	68
5.4 Illustrates the use of Words edit tracking ability	72
5.5 Displays Microsoft Words commentary feature	72

5.6 Shows a sample evaluation (Holmes, 1999).....	74
5.7 Shows MindTrail’s knowledge tree	76
5.8 Displays how the evaluator fills in the appropriate criteria during assessment.....	77

Chapter 6 - The development of a model to support the management of assignments 80

6.1 Displays the process of assignment management	81
6.2 Shows the flow between PAMM’s components	82

Chapter 7 - The implementation of a web-based assignment management model..... 93

7.1 A web-based assignment management system	94
7.2 An ERD-equivalency diagram of the database.....	97
7.3 Assignment creation form.....	100
7.4 The student main menu.....	101
7.5 Student assignment submission form	103
7.6 Displays the student’s view of an evaluated assignment.....	105
7.7 A section of the lecturer’s main menu.....	106
7.8 Displays the submitted assignments for a particular class and assignment.....	106
7.9 Displays the details of a submitted assignment	107
7.10 The evaluation application.....	109
7.11 Error history tree	111

Chapter 8 - Evaluation & Conclusion 114

8.1 Displays how often students had problems logging onto the system.....	116
---------------------------------------------------------------------------	-----

8.2 System ease of use.....	117
8.3 Displays the students' submission preference	117
8.4 Evaluated assignment retrieval preference	118
8.5 Displays the students opinions, on the potential future use of electronically-based submission of assignments	120

Appendix B - User Manual: Assignment Manager 2000 146

B.1 A section of the login screen	148
B.2 Add user form.....	149
B.3 Add class form.....	150
B.4 Register user in class form	150
B.5 Lecturer main menu	151
B.6 Assignment creation form	152
B.7 A section of the lecturer's main menu	153
B.8 Displays the submitted assignments for a particular class and assignment	154
B.9 Displays the details of a submitted assignment.....	154
B.10 Automated evaluated assignment submission form	155
B.11 Manual evaluated assignment submission form.....	156
B.12 Add notice form.....	156
B.13 The student main menu	157
B.14 Student assignment submission form	158
B.15 HTML view of an evaluated assignment	160
B.16 The evaluation application	161
B.17 Error history tree.....	163

List of tables

Chapter 2 – Computer Mediated Communication technologies within educational environemnts	8
2.1 CMC systems and the media that they support.....	14
2.2 CMC systems and their (a)synchronous nature	15
2.3 Four-square model of groupware	18
Chapter 5 - Evaluation of currently available products	61
5.1 Comparison of the features of the classroom management solutions	69
5.2 Comparison of the features of the assignment evaluation solutions	78
Chapter 8 - Evaluation & Conclusion	114
8.1 Compares the features of the classroom management solutions.....	121
Appendix B - User Manual: Assignment Manager 2000	146
B.1 Required format for imported user registration.....	151

Chapter 1

Introduction

1.1 Background

Technology-based curricula are usually best taught by a combination of theoretical instruction and practical assignments. Programming, in particular, is usually taught by making use of both of these approaches, thus allowing them to strengthen one another (Shield, 1996). The theoretical component is taught in class where the lecturer explains the concepts, however, much of the student's time is spent on practical assignments in which the theoretical knowledge is tested and applied. This means that programming students generally have many practical assignments.

Programming can be compared to creative writing. Editing and commentary help develop a programmer's style, just as it does an author's style. The similarities between these fields of study make it possible to extrapolate research done in teaching creative writing to teaching programming. Studies show that one of the best methods to teach creative writing is to give writing assignments and allow for feedback on the assignment, not just grading of it (Day & Batson, 1989). Grading alone taught the student very little, but continuous feedback helped develop the writer. This is also the case within the programming paradigm where feedback is an essential component in the student's development and education.

The feedback that students gain from practical assignments is therefore very important. Unfortunately various factors have negatively influenced the quality of the feedback. Through informal discussions with lecturers and students, it was established that traditionally the code was printed out and then evaluated by the lecturer.

Feedback was given by correcting code segments and making suggestions where needed, in the code. This was easy to do on paper and worked reasonably well, as shown in Figure 1.1. The problem with this system is that while feedback was easy, it was difficult for the lecturer to find all of the syntax mistakes in the program. A popular solution to this problem was to use e-mail to send the code to the lecturer. This allowed the lecturer to run the code on his or her computer to find any problems, and permitted the students to be situated anywhere. This additionally lowered the

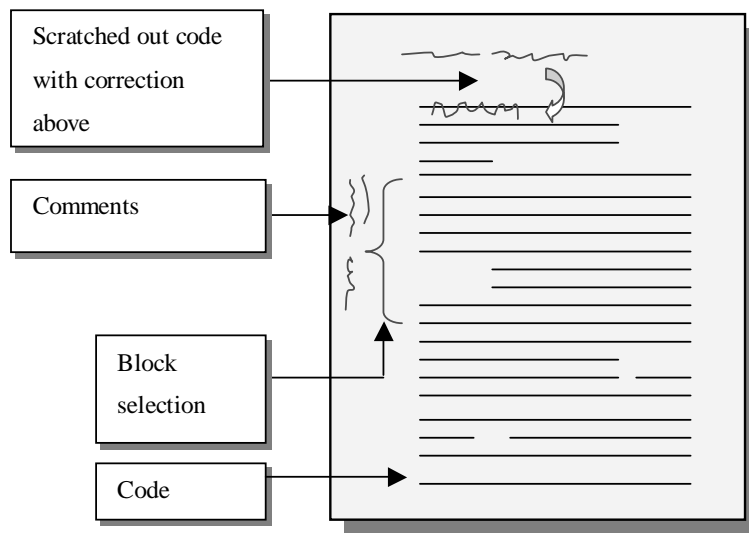


Figure 1.1 An example of how comments are made on paper.

stress placed on the computing resources such as printers, at the institution.

The e-mail-based solution is however insecure, with students claiming to have submitted assignments, and lecturers claiming never to have received them. An additional problem that surfaces when using the current e-mail-based system is that if the lecturer does not print out the assignment, it becomes difficult to provide the same level of feedback, as could be provided on paper. This is largely due to the lack of assignment annotation tools. The lack of feedback makes it difficult for the person who submitted the code to understand what he or she did wrong. Whereas, if adequate feedback were available to the students, they could have learnt from positive criticism to their coding technique.

It is important that the whole process of submission, storage, evaluation, feedback and retrieval be managed in a secure and trustworthy manner. Although systems that could be used for the electronic submission and annotation of assignments do exist, they are generally not well suited for use with programming assignments, which may consist of many files. The advantage of printed-paper over the current electronic systems is that it's easier to scribble comments all over the pages such as in Figure 1.1 (Thomas, 1997). This is not the case with most electronic systems, especially since they generally change the formatting of the source code, making it difficult to read. One may ask, "Why can the lecturer not just make printouts and work from them?" The reason why this is not practical is that, while it works with small amounts of text, problems arise when there are many students submitting large assignments. This may take hundreds of pages and take the lecturer hours to print out, consuming time that most lecturers do not have to waste. Printing costs are also a contributing factor when considering printing out assignments. Programming classes are becoming larger and this puts additional strain on the lecturer in the management, evaluation and giving individual attention to students.

1.2 Problem statement

It has become increasingly popular for students to e-mail their programming assignment to their lecturers. This has placed the onus on the lecturer to print out the assignments for evaluation because providing feedback electronically is difficult. This still does not solve the problem of managing the whole feedback cycle of assignments. Unfortunately very few electronic marking tools exist, and those that do exist are domain specific and not customisable enough for the programming environment. An additional problem is that while extremely powerful groupware products exist, they are very expensive and don't offer the feedback that is required for programming assignments.

It has also become difficult to manage the whole submission of assignments and monitoring who has handed in what and when.

There is no integrated system that allows students to submit their assignments in a secure way and then lets the lecturers mark those assignments and give feedback to the students electronically. Such a system needs to be trusted by both lecturer and students.

These problems are made worse by the inability of most educational software applications to share standardised data or work together to accomplish joint objectives (Rowley, 1995). This is costly and wasteful because many of the products features overlap, or don't integrate well with one another.

1.3 Objectives

The objectives of this dissertation are:

- To determine how the flow of programming assignments between students and lecturers can be streamlined using current technologies;
- To ascertain how rich, intuitive feedback can be attained in programming assignments that are evaluated electronically;
- To determine how the submission, storage and retrieval of assignments can be secured within an electronic environment that is trusted by lecturers and students.

1.4 Methodology

A literature study was conducted which looked at the issues and technologies surrounding the problem areas, namely Computer Mediated Communication (CMC), information security, as well as various groupware-based features. Furthermore the

literature study looked at the needs of such a system, as well as the viability of using currently available solutions for the management of programming assignments.

An assignment management model was then presented and implemented. The model provides for the secure electronic submission, storage and retrieval of assignments, while maintaining adequate levels of feedback,

The viability of the model was tested by means of developing a web-based assignment management system. This also included an evaluation application that could be used by lecturers to evaluate programming assignment.

The system was then evaluated by students and lecturers. This was done by giving the students a questionnaire, and interviewing the lecturer. The information gained from the evaluation was finally used to determine the effectiveness of the proposed system, and to discover further areas of improvement.

1.5 Organisation of the dissertation

The section below describes the content of the chapters contained within the dissertation.

Chapter 2: Computer Mediated Communication technologies within educational environments

This chapter primarily focuses on Computer Mediated Communication (CMC), with the intention to give the reader a broader understanding of how CMC-based technologies can be used to support education. This is done by looking at the general traits that most CMC systems share, and then discussing several commonly used CMC systems within educational environments.

Chapter 3: The importance and application of information security

In this chapter, the importance of security, specifically within an assignment management system, is described. The chapter additionally discuss the technologies and methodologies that can be used to create a secure system.

Chapter 4: Requirements of an assignment management system

Chapter 4 discusses some of the criteria that must be considered when choosing or designing a product, to facilitate assignment management.

Chapter 5: Evaluation of currently available products

The objective of Chapter 5 is to determine the viability of using currently available commercial solutions, to facilitate the management and evaluation of electronically submitted assignments. This is done by studying various education-orientated groupware implementations, specifically focusing on their suitability for use within programming classes.

Chapter 6: The development of a model to support the management of assignments

This chapter discusses a CMC-based assignment management and evaluation model that can be used for the secure submission, storage, evaluation, retrieval of assignment.

Chapter 7: The implementation of a web-based assignment management model

The objective of this chapter is to show the viability of the model described in Chapter 6 by implementing a web-based version of the model.

Chapter 8: Evaluation and conclusion

Chapter 8 discusses how successful the implementation described in Chapter 7 was at attaining its goals. Furthermore, this chapter discusses various opportunities for research and development within the implementation.

Appendix A: Questionnaire

This questionnaire was given to students to evaluate the assignment management system

Appendix B: User Manual

This is the user manual for the prototype described in Chapter 7.

Computer Mediated Communication technologies within educational environments

2.1 Introduction

"The big story in the field of information technology for education and training is not what marvellous new technology we educators now have at our fingertips. The big story is the slow take up of this technology and the challenge this poses for educational managers." (John Mitchell, cited in Szabo, 1998)

This chapter will primarily focus on Computer Mediated Communication (CMC), with the intention to give the reader a broader understanding how CMC-based technologies can be used to support education. This will be done by looking at the general traits that most CMC systems share, and then discussing several commonly used CMC systems within educational environments.

2.2 Defining CMC

"Computer-Mediated Communication is a process of human communication via computers, involving people, situated in particular contexts, engaging in processes to shape media for a variety of purposes." (December, 1997)

Numerous definitions of CMC exist. Some individuals see CMC as a communications component (Szabo & Fuchs, 1998; Borja, 1998; Paulsen, 1995), where computers are used to store, transmit, receive and route messages and information; while others such as December (1997) place it within a broader context involving people engaged in various activities. There is thus far no definitive “definition” of CMC, most probably because, as Murray (1997) states, “*There are many different forms of CMC; it means different things to different people, which is both its strength and the source of some of the problems arising in the research literature.*”

Computer Managed Instruction (CMI) is one form of CMC and it can be described as the use of computers’ storage and retrieval capabilities to organize instruction and track student records and progress (Gottschalk, 1995b). Wegener (2000) notes that CMI is comprised of the following three modes:

- testing,
- prescription generation, and
- record keeping.

The record-keeping component is of particular interest when dealing with an electronic assignment submission and evaluation system. This is due to the fact that detailed information about students’ evaluated assignments can be stored in such a system and later analysed.

2.3 CMC systems

“Computer-Mediated Communication systems, especially when enhanced to create what we refer to as a Virtual Classroom,[TM] can make significant improvements in both access to and the quality of education.” (Hiltz, 1995)

Various implementations of CMC exist, each one meant for a particular situation. The following section will briefly describe the main CMC features that can be used to promote education.

2.3.1 E-mail

E-mail is the electronic version of the postal system, it allows people to send and receive messages. The messages are generally text based, however modern e-mail clients can facilitate HTML and Rich Text-based messages. In addition to this, other media may be included by attaching the files to the e-mail. An advantage of email is that it can support notification of reception as well as group mailing (sending the same message to many people). E-mail requires minimal user training and hardware.

E-mail is extensively used by students and lecturers to facilitate communication and collaboration (Baker & Buller, 1995). Gibson (1997) notes that e-mail is particularly useful in education because conversations between users can take place over vast distances and time. According to Baker and Buller (1995), students and lecturers extensively use e-mail to facilitate question and answer sessions, and to assign and collect homework. Gibson (1997) additionally points out that e-mail is helpful for people with language problems, as they can take time to understand what has been said. The asynchronous nature of e-mail also means that students have time to compose well thought out replies.

2.3.2 MUDs and MOOs

Multi User Dimensions (MUDs) are primarily text-based virtual realities, in which users interact, with one another, and the computer. A user is told what the surroundings look like and the user interacts with them by typing in text commands. This is however not always the case, as some MUDs make use of 3D graphics and other media (Schneider, 2000). MUD Object Orientated systems (MOOs) are MUDs with built in programming languages that allow users to make additions and customizations to the “reality ” in which the users exist, allowing users, to create “rooms” and conference areas (Schneider, 2000). MUDs and MOOs furthermore have the capability to record real-time discussions for later playback for students who were unable to participate in the discussion (Evard, 1993).

Educators prefer to use MOOs in place of MUDs, due to their flexible nature, making it easier to create unique virtual classrooms where students can share knowledge with one another. Fanderclai (1995), in particular notes the advantages that MOOs' flexible nature bring forth, stating; *“Quite possibly they learn more from projects and activities they invent for themselves than from any I assign; certainly they learn things I could not teach them in our four-walled classroom.”*

2.3.3 Word Wide Web (WWW)

“The World Wide Web [WWW], technology holds a lot of promise as an educational tool” (Gibson, Brewer, Dholakia, Voek & Bitzar, 2000)

The WWW is a hypertext-based medium, which supports text, images, audio and video. A study undertaken in October 2000, showed that the WWW is growing at an incredible rate, and that it consists of more than 22 million web sites (Zakon, 2000). Flake (1996) points out that the WWW can be a great source of knowledge for students, and that it can become a social environment for learning. The WWW additionally permits educators to create online classrooms that could also include references to relevant information situated elsewhere on the WWW. For educators, the Web provides unique opportunities for promoting professional development, easy retrieval of information and access to colleagues in the field (Gold, 1996).

Conversely there are many challenges associated with using the WWW, firstly there is information overload, that is to say a person using the WWW to search for certain information is bombarded with information, which is often irrelevant to the person. The WWW can also be used to misinform people, since anyone can post information onto the Internet. The information on the WWW is also very “volatile”, with information and links disappearing overnight (Ciolek, 2000).

2.3.4 Audio conferencing

Audio conferencing allows real time audio communication between participants. This can take place over telephones or the Internet. The advent of advanced audio compression codecs has allowed this medium to be used over the Internet by home users e.g. Microsoft NetMeeting. It is an interactive medium, allowing direct student and instructor participation. Students have many opportunities for give and take with other students, the instructor, and outside experts. Unfortunately audio conferencing can't be used to convey nonverbal cues or body language, restricting the type of content that can be delivered (Gottschalk, 1995a).

2.3.5 Video conferencing

Video conferencing permits real time audio and video communication between two or more locations. Video conferencing requires a reasonable powerful computer at each location as well as a video camera and microphone. Dwyer, Barbieri and Doerr (1995) note that a good deal of information is conveyed through subtle body language such as hand notion and vocal inflections. Video conferencing's ability to include these subtle forms of communication therefore enriches the communications process, enabling the lecturer to more efficiently convey his or her message to students (Dwyer, Barbieri & Doerr, 1995). In addition to this the media richness of video conferencing means that it can be especially useful to students with special needs or who are at high-risk (Woodruff and Mosby, 1996).

2.3.6 Newsgroups

Newsgroups are similar to e-mail in that they are asynchronous and use text as the primary form of communication. Each newsgroup pertains to a specific subject that can normally be deduced from the name. Newsgroups also allow users to create

“threads” of discussion, with each thread dealing with a particular topic. Special “newsreader” applications are required to access newsgroups; they also have the ability to sort and filter the articles by subject, date and size. They keep track of what you have already read and to which newsgroups you have subscribed.

Newsgroups are often used for educational purposes to provide electronic discussion groups for course subjects. This allows students to participate in a discussion about a particular course subject. Students can also use newsgroups to post questions about assignments, or ask for help (Blanchard, 2000).

2.3.7 Internet Relay Chat (IRC)

“Chat in Spanish with a class in Mexico, and you'll get immediate feedback and insight into the language that you just couldn't get from a textbook.” Commonwealth of Pennsylvania (2000)

IRC is a very popular synchronous multi-user chat protocol, i.e. it allows many people to communicate with one another at the same time. It is a text-based medium but like e-mail, you can send attachments. The Commonwealth of Pennsylvania (2000) note that IRC's can be used for various education activities, such as helping people to learn new languages.

2.3.8 Whiteboard

Whiteboarding can be compared to the traditional black board. It allows images to be displayed, manipulated, annotated, and shared between people (Dwyer, Barbieri & Doerr, 1995). All of the users see the changes made on the board and all can participate. Whiteboard applications normally also allow for text communication between the participants, similar to IRC.

2.4 Media usage of CMC

CMC systems can make use of various media types (see Table 2.1). They include graphics, text, image, video and audio. Text is the simplest media and refers to characters, letters and numbers. Graphics refer to artefacts such as lines, circles and boxes that are not pixel related (e.g. Clipart). Images, on the other hand, consist of an arrangement of small picture elements called pixels, with each pixel representing a single colour. Video takes images one step further, by displaying many images at a time, motion is perceived (e.g. DVD movies work on this principle). Audio makes use of sound to communicate with us. Lennon and Maurer (2000) note that multiple media types are currently used with great success in medical environments. They go on to state that its use in classrooms can liven up the learning experience.

CMC SYSTEMS / MEDIA TYPE	Text	Graphics	Image	Video	Audio
E-mail	√	√ 1	√ 1	√ 1	√ 1
WWW	√	√	√ 1	√ 1	√ 1
Audio Conferencing	√ 2				√
Video Conferencing	√	√	√	√	√
IRC	√	√ 1	√ 1	√ 1	√ 1
MUD	√	√ 2	√ 2	√ 2	√ 2
Whiteboard	√				
Newsgroups	√		√ 1	√ 1	√ 1

√ 1 This feature is available as an attached or embedded file.

√ 2 This Feature is only available on certain software packages.

Table 2.1 CMC systems and the media that they support.

2.5 Traits of CMC systems

CMC has various traits, advantages and disadvantages. Studying these will eventually help in the selection process of choosing a CMC system.

2.5.1 Synchronous and asynchronous nature

CMC permits both synchronous and asynchronous communication. (See Table 2.2). Synchronous communication permits information to flow in both directions simultaneously (e.g. telephone call), while asynchronous communication only permits information to flow in one direction at a time (e.g. CB radio, e-mail). The asynchronous nature of many CMC systems enable users to take their time to respond to questions. Therefore, instead of rushing replies, well thought-out replies can be sent (Rowntree, 1995; Warschauer, 1997; Gibson, 1997). A problem with this however is that the users are not pressed to respond, as they would in “normal” synchronous communication. Students may also become frustrated, since it could take a long time for a lecturer to respond to pressing questions (Hiltz, 1995).

CMC SYSTEMS / COMMUNICATION TYPE	SYNCHRONOUS	ASYNCHRONOUS
E-mail		√
Newsgroups		√
WWW	√	√
Audio Conferencing	√	
Video Conferencing	√	
IRC	√	
MUD	√	
Whiteboard	√	

Table 2.2 CMC systems and their (a)synchronous nature.

2.5.2 Time and distance constraints

One of CMC's most important benefits is that it liberates instruction from time and distance constraints (Berge & Collins, 1995b; Shimabukuro, 1995). This aspect of CMC allows students to learn any subject matter from any location irrespective of their time zone or geographic location (Lewis, Whitaker & Julian, 1995). This trait is especially important with electronic submission systems as students may submit assignments at any time or from distant locations, liberating them from using set locations and times for handing in assignments.

2.5.3 Cost saving

CMC can potentially pay for itself and even save organisations money. The cost of computer equipment and communication lines can be offset by the savings in paper and printing that would have taken place in a non-CMC environment (Lewis, Whitaker & Julian, 1995). Assignment submission costs, from a distance education student's perspective, are also reduced because students don't have to post their assignments, saving on postage expenses. On the other hand, as Jones and Jamieson (1997) note, the extra hardware and support staff can mean new costs.

CMC users additionally waste less time because it allows them to communicate from virtually any geographic location, making commuting to educational institutions unnecessary.

2.5.4 Personal development

CMC gives everyone the ability to express their viewpoint and gives the user confidence in their own skills. It also promotes self-discipline and responsibility upon

its users. Unfortunately, not all students are capable of adequately managing their own learning and therefore this could act as a disadvantage (Berge & Collins, 1995a).

2.5.5 Textual nature of CMC

An aspect that is also very important is the fact that most of CMC's instructional use is text-based (Shimabukuro, 1995). This requires the users to be able to express themselves in text, a requirement that many people are not able to fulfil. (Lewis, Whitaker & Julian, 1995). Additionally users with poor typing skills may find it difficult and frustrating to type out large sections of text. This factor may prevent such users from fully participation in discussions between other users. Another trait of the textual nature of CMC systems is that social cues are lost (Berge & Collins, 1995a; Rohfeld & Hiemstra, 1995). This has the advantage in that users aren't discriminated against because of their physical appearance, race or gender. The students also don't feel like they are being measured. Soft-spoken users, who would have kept quiet in class, participate more often within the CMC-based class (Kinner & Coombs, 1995). The other side of the coin looks different. Minimisation of social cues can cause the non-reticent personalities to become aggressive and publicly inflammatory, often "flaming" other users (Berge & Collins, 1995a).

An additional problem related to the textual nature of assignment evaluation is that of typing by the lecturer. The evaluator may have poor typing skills and therefore may prefer paper-based evaluation. This is however, for the most part not a problem for programming lecturers.

2.6 Groupware

Technologies that support collaboration are in greater demand than ever before and this is where groupware enters the picture (Coleman, 1995). Groupware is an umbrella term describing the electronic technologies that supports interpersonal

collaboration. Groupware technology includes e-mail, electronic meeting systems (EMS), desktop video conferencing (DVC) as well as workflow and business process re-engineering (BPR) systems. As with CMC, a major advantage of groupware is that it liberates the users from time and distance constraints. Table 2.3 shows the four-square model of groupware that was popularised by Robert Johansson (Bock & Marca, 1995).

	Same Time (synchronous)	Different Time (asynchronous)
Same Place	<ul style="list-style-type: none"> • Shared white board • Meeting support rooms 	<ul style="list-style-type: none"> • Control room activity log
Different Place	<ul style="list-style-type: none"> • Video wall • Remote image displays 	<ul style="list-style-type: none"> • E-mail • Computer conferencing

Table 2.3 Four-square model of groupware (Bock & Marca, 1995).

Groupware belongs to field of study named Computer-Supported Co-operative Work (CSCW). CSCW in turn examines the design, adoption, and use of software that enhances collaboration, interpersonal communication and project management (Bock & Marca, 1995; Brinck, 1998).

Designing groupware systems involves the understanding and behaviour of people within group situations. It also involves the understanding of network-based technologies and how the aspects of these technologies affect a user's experience. Since groupware involves individuals interacting with technology, traditional issues in user interface design are also relevant (Brinck, 1998). These issues are also relevant to an assignment management system, since it involves many users exchanging information such as assignments and feedback, over networks.

2.7 Workflow

What is workflow? The Workflow Management Coalition defines workflow as “*The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.*” (WFMC-TC-1011, 1996). The Coalition goes further to define a workflow management system as “*A system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications.*” (WFMC-TC-1011, 1996). This means that a workflow management system manages the flow of information and control in an organisation’s business processes.

Workflow technology can be used to:

- Streamline an organisation’s processes;
- Decrease cost of business processes through automation;
- Speed up organisational processes and, as a result, deliver a faster service to customers; and
- Track and control organisational processes.

Workflow systems may also provide features such as routing, development of forms, and support for differing roles and privileges. A simple application of workflow is the processing of student assignments in an educational institution. A student completes an assignment and submits it electronically. A copy is archived then routed to the appropriate lecturer. The lecturer evaluates the document electronically, adding commentary where needed, and sends the corrected document back to the student. The student is then notified that the assignment has been evaluated.

Workflow management systems have become more popular due to their flexibility, customer orientation and adaptation to changing market conditions. Consequently new applications for workflow systems are found every day (Adelsberger, Korner &

Pawlowski, 1998). The growth of workflow has resulted in its use within educational environments (van der Veen, Jones & Collis, 1998; Rada, 1999). In particular, van der Veen, Jones and Collis, (1998) noted that prior to their adoption of workflow, their old method of electronic assignment submission, evaluation and resubmission had many delays. *“Prior to this (in academic year 1996-97 for example), student groups had handed in their assignments as attachments to email messages generated by filling in CGI-forms and submitting them via the course web-site. After comments were received and any necessary rewriting completed, the deliverables were sent on to the webmaster who linked the deliverable to the course Web site. This multi-step process caused some unnecessary delays (caused by waiting for human interventions) and was capable of further automation.”* Subsequently they applied a workflow-based solution to this processes, and established that both lecturers and students found the workflow-based solution to more beneficial than the previous solution (van der Veen, Jones & Collis, 1998). Groupware vendors have responded to workflow’s popularity by incorporating many workflow-based functions into their respective products.

Workflow applications can be generally placed into three categories, namely messaging-based, web-server-based and production-based workflow. Messaging-based workflow applications make use of messaging technologies such as e-mail servers to route tasks through e-mail. Web-server-based workflow systems use web servers, to which participants connect using their web browser. This has the advantage that the participants do not need specialized software to connect to the workflow. However; the disadvantage of web-serve-based workflow systems is that workflow cannot “push” or notify a user to perform a task, because it needs the user to first log onto the web server. Production-based workflow systems automate complex and structures business processes that people do every day, often in volume. Production-based workflow systems are also usually used for mission-critical applications.

The following section will briefly discuss some of the problems that users face when moving from paper-based to paperless assignment submission and evaluation systems. The section will specifically focus on how the reading and editing is affected by moving to a paperless environment. This will help determine some of the potential

problems that students and evaluators may face, if assignments are evaluated electronically.

2.8 From paper to electronic assignments

“Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.” (Human-Computer Interaction, 2000)

In a study by Sellen and Harper (1997), it was found that International Monetary Fund (IMF) employees preferred printed-media to electronic-media when comprehension was required. This resulted in them printing out such documents, enabling them to comprehend the content better. This also appears to be the case within the programming environment, as it is not uncommon for programmers to print out code in order to better trace through the code and find logical bugs. Moreover this could also be true for evaluators, as they have to comprehend and evaluate other people’s code.

Three commonly recurring factors appear to contribute to the users’ preference to paper verses electronic media (Sellen and Harper, 1997; O’Hara and Sellen 1997). Namely that users found annotation with text editors difficult, and that users preferred navigation and positioning with paper. In their study it appeared that navigation through paper was quick and easy, while it was slow when using electronic media. Users also found it easier to get an overall view of a text when using paper. This was mainly due to the fact that the users could unclip documents and arrange them adjacent to one another to get a better overall view. The users could also use one hand to bookmark their current position while quickly moving to another page to read some other part easily. Conversely users found it difficult to do this with electronic media, especially since the screen space did not permit an entire page to be displayed at a time, making it impractical to work with multiple windows simultaneously. Sellen

and Harper (1997) did, however, note that larger displays or the use of multiple displays could improve access to other documents, partially alleviating that problem.

Paper-based systems are generally more portable than their electronic counterparts (Yankelovich, Meyrowitz & van Dam, 1991). This is especially important fact to consider, as assignments are often evaluated by the lecturer at home, where the required hardware facilities for electronic evaluation may not exist. This problem could additionally be further compounded by the fact that users often complain about eye strain when reading large volumes of text on monitors, a task assignment evaluators most certainly have to contend with (Yankelovich et al., 1991).

2.9 Summary

This chapter gave a brief overview of CMC, looking at various implementations of it and its use in education. The chapter also discussed some traits that are common to most CMC systems, and finally looked at some of the problems that users face when moving from paper to electronic media. Groupware and workflow were also described. None of these technologies would be useful in an assignment management system if students and lecturers did not have trust in their security features, however.

The next chapter will describe the importance of information security, specifically within an electronic assignment management system. The chapter will additionally focus on some of the most commonly used technologies which could be used to create a secure electronic assignment management system.

The importance and application of information security

3.1 Introduction

An electronic submission and feedback system needs to be considered trustworthy, reliable, and thus secured (Joy & Luck, 1995). The information that such a system deals with is of a sensitive nature and it would have disastrous effects if students could gain unauthorised access to the system and change marks, modify submitted assignments etc.

Information has become a vital asset in most organizations and thus needs to be protected, as other assets are. Information security means preserving confidentiality, integrity and availability at all times. This has become a major task because of multiple platforms and the distribution of information systems. A good information security architecture must therefore be followed to ensure that all aspects of protecting information security resources are covered.

In this chapter, the importance of information security, especially within the proposed system, will be pointed out. Due to the distributed nature of the problem at hand, a distributed information security architecture will be discussed. After which, the chapter will look at some of the technologies that are prevalently used to provide information security. This will be accomplished by firstly introducing the reader to the “building blocks” upon which cryptographic technologies are based. It will then look at how these “building blocks” can be used within security protocols and systems to provide information security.

Information security plays a vital role within an electronic assignment management system. The information stored within, and traversing, such a system, needs to be secured (Joy & Luck, 1995). Assignments contribute a large percentage of students' semester marks, which makes this especially important. Students should not have the ability to modify, copy or delete one another's assignments and marks. The systems must therefore only permit authenticated users to have access to the various facilities of the system. Malicious users may also attempt to modify or capture information that is transmitted over an unsecured medium, it is therefore also important to maintain the information's confidentiality and integrity.

3.2 A distributed security architecture

In an electronic submission and evaluation system, students and lecturers may be located at different places while using the system. The components of the system itself, may similarly also be distributed over various machines and networks. It is thus more difficult to secure the system and provide a high level of trustworthiness in the system. This section will therefore discuss a distributed security architecture. Creating a secure distributed environment is a multifaceted problem requiring a multifaceted solution. The complexity can be managed through effective planning. To this end, a blueprint of the secure environment is needed. This "blueprint" is also known as the security architecture. The security architecture is the framework around which the protected environment is built.

The architecture provides the conceptual definition and structure of the environment. It allows for the design of individual secure components within the environment. Additionally the architecture specifies how the individual components are to be integrated into the overall environment. Finally, the architecture can be used to verify that the solution corresponds with the original established vision of the project.

The following section will discuss Bruce and Dempsey's (1997) distributed security architecture (see Figure 3.1). It is very complete, considers both technical and social aspects, and works in distributed environments. It is not, however, the only way to structure a security architecture.

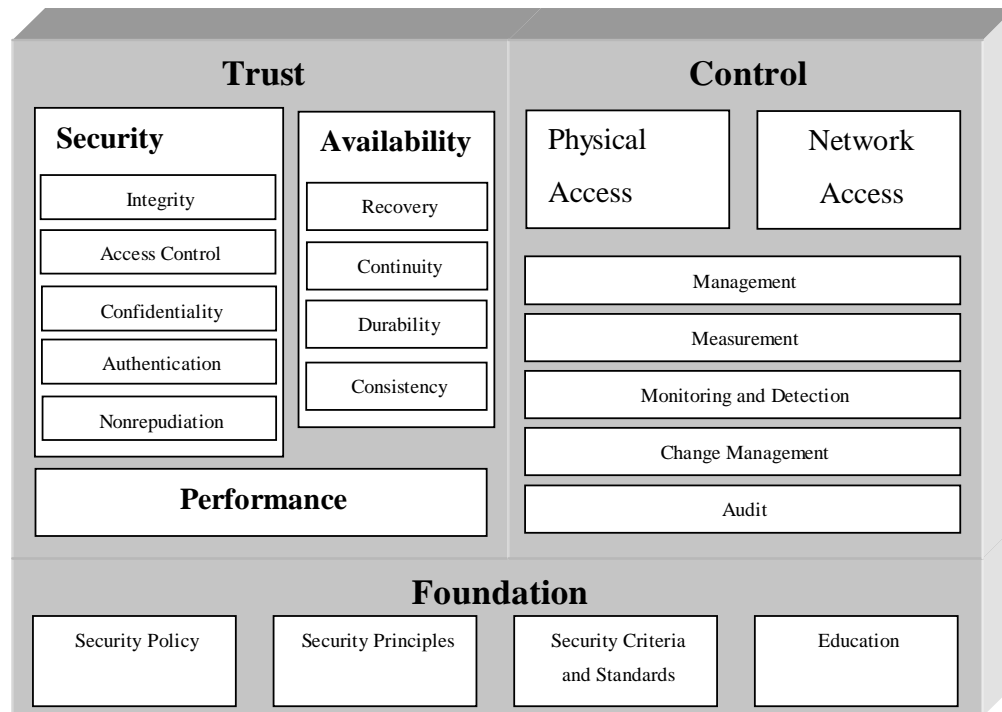


Figure 3.1 Bruce and Dempsey's (1997) distributed security architecture.

Bruce and Dempsey's (1997) distributed security architecture is composed of three major components, **foundation**, **trust** and **control**. The **foundation component** is comprised of the overall **security principles** as defined by the organisation. **Security policies** are also integrated within the foundation component and govern the implementation and use of the security mechanisms. Additionally the foundation includes chosen **security criteria and standards**. Bruce and Dempsey (1997) compare the architecture, specifically the foundation, to a house. The foundation is comparable to a house's foundation and must possess the ability to, not only support the rest of the "house", but also to additionally support further additions and renovations. In a security architecture the foundation is composed of statements and decisions that describe the type of secure environment that is needed together with the qualities that the environment must possess.

The **control component** outlines the mechanisms that are used to control and manage the required security components. It provides measurement and management features that are required to oversee the secure operation of the system. **Physical access** mechanisms are used to control access to the computing and networking hardware within the system. Various devices can be used to prevent unauthorized access to equipment, some of which may include smart cards, keys or even fingerprint and retina scanners. Similarly **network access** controls are used to limit access to network resources to only people or processes authorised to do so. **Management** is required over numerous areas within a distributed system and include the network, operating system, network operating system, database management, middleware, access control mechanisms, and key management.

Measurement facilities are used to analyse and report on the performance characteristics of the various components within a system. Measurement facilities can also be used to detect anomalies thereby indicating possible problem areas. **Monitoring and detection** agents could, for example, be used to detect if any student is attempting to gain illegal access to the system and notify the relevant people. **Audit trails** are used to keep track of various security events within the system. This information can later be used to determine who or what was responsible for a particular security incident. Finally, the control component is responsible for **change management**; this is the management of the impact that changes to the security mechanisms and policies will have on the organization. (Bruce & Dempsey, 1997)

The **trust component** of the architecture looks at how performance, security and availability determine the level of trust that can be placed in a system. The following section will describe the trust component of the model, as it plays the most crucial role in an electronic assignment management system.

3.3 Trust

Bruce and Dempsey (1997) state that trust refers to the ability of a system to perform

its actions with integrity, to keep confidential information private, and to perform its functions on a continual basis. Trust can, therefore, be defined as the composition of security, availability and performance, each of which must be present within a system and within acceptable limits. The issue of trust is especially important in development and implementation of a solution to the problem area of this dissertation, therefore the chapter hereafter will focus on the technologies that can be used to strengthen the level of trust placed in a system based on the proposed model.

3.3.1 Security

Security is an essential building block in helping users to trust a system, because users need to know that the assets are protected from various threats. This section will briefly describe the mechanisms that are needed to ensure a secure distributed system, such as within the proposed assignment submission and feedback system. Later in the chapter, various technologies that can be used to implement these mechanisms will be discussed.

- **Integrity**

Integrity mechanisms ensure that the data has not been modified in any form during its storage or transit. Mechanisms must identify and correct or report incidents where the integrity of data was breached so that appropriate actions can take place (Bruce & Dempsey, 1997; CPS4001, 1998; Information technology security team, 1998). Within an electronic submission system the integrity of the data plays an essential role in determining the level of trust that the students have in the system. One of the issues that should be addressed is the verification of the integrity of the students' submissions, as the data could have been unintentionally corrupted during transit. Students should therefore be presented with tools that enable them to verify the integrity of their submissions.

- **Access Control (Authorisation)**

Once an entity has been identified and authenticated, access control mechanisms ensure that an entity is only allowed to access authorised system resources. Access control, also known as authorisation, is usually determined by a user ID together with a validated password and can additionally be based upon time dependency, the users role, data classification and so forth (Bruce & Dempsey, 1997; CPS4001, 1998; Information technology security team, 1998).

- **Confidentiality**

Confidentiality services ensure that the data is protected from disclosure to unauthorised entities. Encryption is extensively used to provide confidentiality, especially over unsecured mediums such as networks. Unfortunately due to performance and cost related issues it is not feasible to always encrypt all traffic and therefore a balance must be found between the security needs and the performance penalties and costs incurred (Bruce & Dempsey, 1997; CPS4001, 1998; Information technology security team, 1998). Submitted student assignments need to be kept confidential, to prevent other students from plagiarising. Students are also often concerned about the public availability of their marks. Mechanisms that ensure the privacy of the students' information, including assignments, should therefore be put in place.

- **Authentication**

Authentication mechanisms are used to verify the authenticity of an entity's identity. Authentication can be based on something an entity knows, such as a secret pin number, a physical trait of a person such as a fingerprint, or on the ownership of an item, such as an identification card. (Bruce & Dempsey, 1997; CPS4001, 1998; Information technology security team, 1998).

- **Non-repudiation**

Mechanisms that support non-repudiation are used to prevent entities from denying that an action was taken or that a message was sent or received. Two cases of non-repudiation exist. Repudiation of origin happens when there is a disagreement of the origin of the data item, while repudiation of delivery is concerned with whether the data was actually received (Bruce & Dempsey, 1997; CPS4001, 1998; Information technology security team, 1998). Non-repudiation mechanisms can provide students with proof of their submissions, as students have often said that they have e-mailed their assignments to a lecturer, who in turn denied ever receiving them.

3.3.2 Availability

The availability of a distributed information system is extremely important. A system that is randomly unavailable is untrustworthy. Availability is very important within an electronic submission and feedback system because the students must be able to submit assignments at all times, especially when the system is used within a distance education environment. The lecturer must also have access to these assignments so that assignment evaluation can take place in a timely manner.

- **Continuity**

Continuity is the ability to withstand a complete interruption of service, and to subsequently invoke recovery procedures in the event of the hardware failure (Bruce & Dempsey, 1997). Although important, continuity is not critical in an electronic assignment management system, as assignment due dates could be extended. Lecturers do not like to do this however.

- **Recovery**

Recovery refers to the ability to recover from a complete service interruption without any manual intervention taking place (Bruce & Dempsey, 1997). It is important that

an assignment management system is able to correctly recover from partial transactions, such as partial submissions. A partial submission in particular may result in the evaluator only receiving and evaluating a portion of the submission, negatively influencing the mark attained.

- **Durability**

Durability is the ability of a system to withstand a partial or gradual degradation of service, due to the failure of physical components within a distributed environment (Bruce & Dempsey, 1997). As with continuity, durability is not critical in an electronic assignment management system, as assignments could be submitted or retrieved once the problem has been solved, and due dates could also be extended if need be.

- **Consistency**

Consistency refers to the ability to consistently attain the same results if the same data and processing criteria are given (Bruce & Dempsey, 1997). Consistency is very important in an assignment management system. One must be confident that the assignment management system handles the submission, processing and retrieval of assignments consistently. Furthermore, any processing that is done on assignment marks that are stored within the system, should consistently produce the same results.

3.3.3 Performance

A system must respond to commands and requests in a timely fashion for it to be regarded as trustworthy. Various factors such as hardware, communications medium, encryption and protocol choices affect the performance characteristics of a system. (Bruce & Dempsey, 1997). In an assignment management system the performance of assignment submission and retrieval-based functions are crucial. The main factors that influence this is the network's bandwidth and the actual storage and retrieval of assignments from a database or storage structure.

3.4 The use of cryptographic technologies to ensure a secure environment.

The remainder of the chapter will look at information security from a technological viewpoint, specifically focusing on the use of cryptographic technologies, to ensure a secure and trustworthy environment.

3.4.1 What is cryptography?

The word cryptography comes from two Greek words κρυπτο and γραφή meaning *hidden/secret* and *writing* respectively (Kaufman, Perlman, & Speciner, 1995). Cryptography can therefore be defined as the science of rendering information into an unintelligible form; as well as reconstructing the information back into intelligible form. The following section will briefly look at the terminology and use of cryptographic techniques, particularly within distributed environments such as within the Internet and intranets.

In cryptographic terminology, a message in its original form is known as **plaintext** or **cleartext**. **Encryption** is the process whereby the plaintext is modified into a new form known as **ciphertext**. **Decryption** is the reverse process of encryption whereby ciphertext is changed back to plaintext (see Figure 3.2).

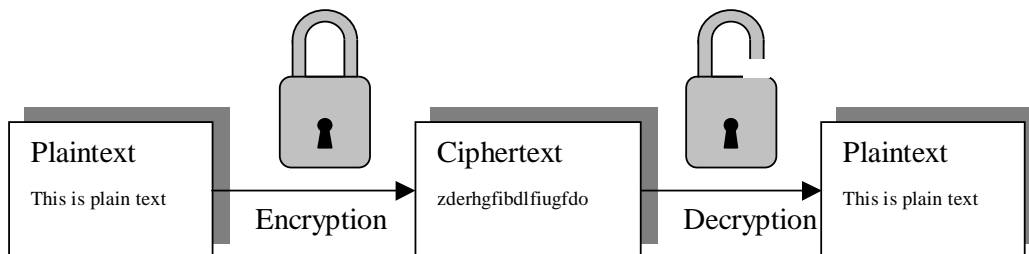


Figure 3.2 The process of encryption and decryption of a message.

Cryptographic systems usually use an algorithm in conjunction with a secret value. The algorithm or method of encryption and decryption is called a **cipher**, while the secret value is referred to as the **key**. A key is comprised of a number of bits, the exact number is known as the key's **keyspace**. Most respectable cryptographic ciphers have their algorithms published for peer review, as knowing the algorithm without the appropriate decrypting key, is of little use to people with bad intentions. Publishing the cipher also has the advantage that many people may help to uncover any possible weaknesses and solutions to the cipher, thereby strengthening it.

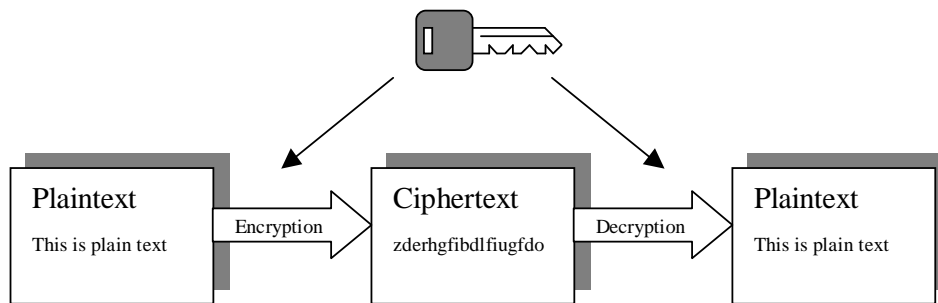


Figure 3.3 Secret-key (symmetric) based encryption and decryption.

Key-based ciphers can be placed into two categories, namely **symmetric** and **asymmetric** algorithms. Symmetric algorithms, also known as **secret-key** algorithms, use the same key for encryption and decryption (Figure 3.3). In turn, asymmetric or **public-key** algorithms use related, but different keys for encryption and decryption (Figure 3.4). An additional property of public-key algorithms is that the one key cannot be derived from the other and visa versa (SSH Communications Security Ltd., 1999b).

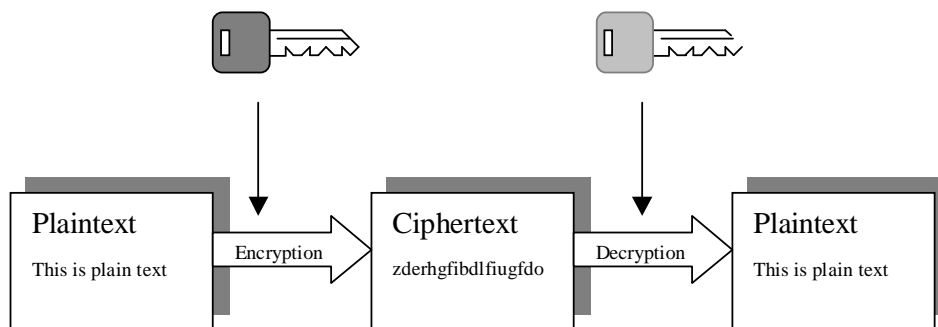


Figure 3.4 Public-key (asymmetric) based encryption and decryption.

Symmetric algorithms can be divided into stream and block ciphers. Stream-based ciphers operate on a single bit, byte or word at a time, while block-based ciphers operate on groups (usually 64 bits) of data at a time (SSH Communications Security Ltd., 1999b).

3.4.2 Factors influencing the strength of cryptographic systems

It is important to know how secure or “strong” a particular cryptographic system is as this will certainly affect the choices made during the development of a secure system. The strength of a cryptographic algorithm is dependant on various factors, however almost all encryption mechanisms strength is highly dependant on the size of the keyspace and the integrity of the algorithm used (Bruce & Dempsey, 1997). The larger the keyspace, the stronger the code. Experts therefore generally consider 90-bit keys as the minimum safe size for secret key-based algorithms and 1000-bit for public key-based systems (SSH Communications Security Ltd., 1999b; RSA Data Security, Inc., 1997). Unfortunately, the government of the United States of America classifies encryption technology as munitions and has strict exportation laws prohibiting the exportation of strong encryption-based systems (Bruce & Dempsey, 1997). The Clinton administration relaxed these restrictions in 1999 (Associated Press, 1999). Unfortunately only the latest web-browsers such as Microsoft Internet Explorer 5.5 have built in support for 128bit encryption. Older web-browsers need to be specifically updated to add this support and this is rarely done

In 1997 Ian Goldberg, a student at the University of California at Berkeley cracked a 40-bit algorithm in under four hours, by making use of a network of about 250 computers. The professor, Eric Brewer, who oversaw Goldberg's work with the Internet Security, Applications, Authentication and Cryptography research group had the following to say, "*If we can break a random 40-bit key in three-plus hours, it means Internet commerce based on 40-bit keys is unacceptable,*" (CNET News.com,

1997). This fact is important to remember as the international versions of S/MIME and SSL make use of 40-bit keys (Counterpane Internet Security, Inc., 2000).

This can have important implications on an electronic assignment submission system as a secure web-based or e-mail-based system would in most cases use SSL or S/MIME. It would also be relatively easy for an Information Technology (IT) student to get access to multiple computers. He or she could do this by embedding the cracking software into a screen saver and freely distributing it, or by embedding the cracking software within a Java Applet. The applet approach would require the student to embed the applet within a web page. Visitors to the web page would then unknowingly assist in the cracking process.

3.4.3 Encryption and decryption algorithms

Many encryption algorithms exist, each with their own strengths and weaknesses. The most common issues that must be considered when selecting an algorithm for use within a system is an algorithm's strength, performance, reliability as well as the legal issues of using the particular algorithm. This section will briefly discuss the traits of the most commonly used encryption and decryption algorithms. Knowing the strengths of these algorithms enables the reader to determine the level of security that products which make use of these algorithms exhibit. This can be useful when choices must be made between competing solutions such as OpenPGP and S/MIME.

- **DES (Data Encryption Standard) and TDES (Triple DES)**

DES is a block cipher and uses a fixed-length 56-bit key that encrypts data in 64 bit blocks. The extra 8 bits are reserved for parity and are ignored by DES. The key-space is therefore 2^{56} making it vulnerable to attack, and should therefore not be used in new applications (SSH Communications Security Ltd., 1999a).

Triple DES (TDES) is a more secure variation of DES, from which various other variations in turn exist. One variation in particular uses two keys while another uses three keys thereby using 112-bit and 168-bit keyspaces respectively (Hughes, 1995).

- **IDEA**

International Data Encryption Algorithm (IDEA) is a 64-bit block cipher that uses a 128-bit key. IDEA is optimised for software-based implementation and therefore has similar, if not, faster performance characteristics than DES (Hughes, 1995). IDEA is considered to be a very strong algorithm and is therefore extensively used within applications such as PGP (SSH Communications Security Ltd., 1999a).

- **RC4**

RC4 is a proprietary cipher. Its algorithm was until recently a trade secret, however there is strong evidence that anonymously posted source code on the UseNet, is RC4. RC4 is essentially a random number generator, the output of the generator is exclusively-ored with the data stream. RC4 supports variable length keys and is much faster than DES (Hughes, 1995; SSH Communications Security Ltd., 1999a).

- **Diffie-Hellman**

Diffie-Hellman is an algorithm that defines a secure mechanism by which symmetric keys can be exchanged. It is generally considered to be secure when sufficiently long keys and proper generators are used (Hughes, 1995; SSH Communications Security Ltd., 1999a).

- **RSA**

The RSA algorithm is an encryption and decryption algorithms that was named after its inventors, Rivest, Shamir and Adleman (Kaufman, Perlman, & Speciner, 1995). The system uses two related keys, one which is kept secret and is used for decryption and another publicly available key, used for encryption.

Messages encrypted with the public key can therefore only be decrypted with the private key and visa versa. Unfortunately the RSA algorithm is about 100 times slower than symmetric block algorithms such as DES and thus mainly used to encode or decode small amounts of data (Hughes, 1995).

3.4.4 Hash functions

Hash functions are typically used to determine a **message digest** of a document, which is like a “digital fingerprint” of the document. The hash function computes the digest by compressing the bits of a message to a fixed-size hash value. This is done in such a manner that it distributes the possible messages evenly among the possible hash values. The resulting hash value is relatively unique for it is extremely unlikely for any two documents to have the same hash value.

Hash functions have many possible applications, for example within an electronic assignment submission system, when an assignment is submitted, a hash function can be run to determine the message digest of the submission. This value could then be stored in a database. If, at a later stage, the assignment needs to be resubmitted, the new submissions message digest could be compared to the value stored within the database. Similarly, if the integrity of the stored assignments come into question, their integrity can be verified by comparing their message digests with those stored within the database.

- **MD2, MD4 and MD5**

MD2, MD4 and MD5 are all one-way hash functions, with each incarnation improving on the previous version. MD2 was designed for use in Privacy Enhanced Mail (PEM) and produces a 128-bit hash value. In turn, MD4 also produces a 128-bit hash value, however the algorithm is optimised for 32-bit registers and is more secure than MD2. MD2 and MD4 have however known flaws, and have been broken. MD5

was therefore specifically created to overcome some of the weaknesses in MD4 (Hughes, 1995; SSH Communications Security Ltd., 1999a).

- **SHA**

Secure Hash Algorithm (SHA) is closely modelled around MD4 however it contains various enhancements, one being a 160-bit hash value making it even more secure than MD5.

3.5 Digital Signatures

A digital signature is used to verify that a message really comes from the claimed sender (Kaufman, Perlman, & Speciner, 1995). The first step in creating a digital signature of a document is by creating a **message digest** of the document. This is then concatenated with information about the signer, a timestamp and other relevant information. The resulting string is in turn encrypted with the private key of the signer. The result of this is an encrypted block of bits, called a digital signature. For the signature to be of use, the recipient must be able to know who signed it, therefore information pertaining to the public key that was used to sign it is usually distributed together with the signature.

The recipient of the document or message would verify the signature as follows. Firstly, the recipient must determine whether the supposed sender's key is trustworthy. Once this has been determined, the signature can be decrypted using the public key of the person (Kaufman, Perlman, & Speciner, 1995). If the signature's decryption is successful and the message digest is valid, the signature is valid.

Digital signatures can play an important role in an electronic assignment submission system as they can be used as tokens that are given to students as proof of submission. A hash algorithm could be used on the submitted assignments to create digital fingerprints of the submissions. These would be signed by the system and then sent to the students as a token of their submission.

3.6 Security protocols for the Internet

The Internet is a worldwide collection of computer networks. It was devised in the sixties by researchers at the Defence Advanced Research Project Agency (DARPA) to act as a communication system that could withstand substantial damage and still function, such as in the case of nuclear war. This is achieved by using data communications standards that enable computers on the Internet to find one another via various routes.

The World Wide Web was envisaged by Tim Berners-Lee and his colleagues in 1992. They sought an easy method of circulating pre-print research papers for peer review around the Internet without requiring each recipient to have various word processing software packages installed. The first step in accomplishing this goal was to devise a platform independent language, Hypertext Markup Language (HTML). An HTML browser was then specified that would have the ability to navigate and view the documents. Berners-Lee proposed an Internet protocol called Hypertext Transport Protocol (HTTP). Berners-Lee then named the interconnection of HTTP servers and HTML browsers the World Wide Web. Unfortunately the original Internet protocols did not have security in mind, resulting in additional protocols being designed to fit into the TCP/IP suite (Dumas, 1995).

Many of the transactions which take place over the Internet are of a sensitive nature and must be secured from various forms of attack. Security protocols were consequently developed to counter various threats to security and to give the user peace of mind. Although many security protocols exist, protocols such as PCT (Private Communications Technology), STT (Secure Transaction Technology) and SEPP (Secure Electronic Payment Protocol) will not be discussed due to their limited scope and use. This section will hence discuss the three most commonly used general-purpose security protocols namely S-HTTP (Secure Hyper Text Transport Protocol), SSL (Secure Sockets Layer) and Kerberos.

3.6.1 Secure HTTP

S-HTTP acts as an extension to HTTP, its purpose is to provide security services. S-HTTP was designed to be flexible and to easily support various algorithms, certificates, key management and security policies. The flexibility includes the ability to allow clients and servers to negotiate policies such as allowing or disallowing certain services. S-HTTP's flexibility extends to supporting multiple key management systems, including public key-based systems, Kerberos, and symmetric key algorithms. The structure of S-HTTP messages is comparable to that used in PEM, and likewise can be used to distribute certificates (Bernstein, Bhimani, Schultz & Siegel, 1996).

3.6.2 Secure Sockets Layer (SSL)

SSL is used to provide confidentiality and authentication of messages. It does this by providing a generic security framework on which application protocols can run. This is done by securing the lower layer protocols such as TCP and IP, enabling protocols such as HTTP, FTP and Telnet to run as normal over the secure medium. SSL consists of two protocols, the record and handshake protocols. The record protocol is used for the actual transmission of data while the handshake protocol is used to negotiate the techniques that will be used to provide various security services such as confidentiality and authentication. (Bernstein et. al., 1996).

The operation of SSL works as follows. Once the client has connected to a secure server, the server sends a *hello request* message. The client then replies with a *client hello* message. This client's message contains a session ID, which uniquely identifies this session between the browser and the server. The message also includes a number which was randomly generated by the browser as well as information telling the server which cryptographic ciphers the browser supports together with the SSL version and compression methods supported. The server then chooses an encryption

algorithm, compression method and SSL version, together with a different random number, and an acceptable session ID number. This information is then sent back to the client in a *server hello* message.

The client and server can now exchange digital certificates, to authenticate each other if need be. The server's certificate can also include a public key appropriate to the public-key encryption algorithm selected during the handshake protocol. This key will be used only for a short time, however; the actual transaction will be encrypted using a private-key encryption algorithm.

3.6.3 Kerberos

Kerberos is a network authentication protocol that is designed to provide strong authentication for client-server applications, and was originally developed at the Massachusetts Institute of Technology (MIT) in the mid 1980s. Kerberos provides for authentication across unsecured networks, however it can additionally provide integrity and encryption if required. It works by assigning a unique key, called a *ticket*, to each user who logs on to the network. These tickets are used for authentication between clients and include a time stamp to limit the time period for which the ticket is valid. Kerberos is mostly used in application-level protocols, such as TELNET or FTP, to provide user to host security (Jaspan, 1995). Kerberos is available on Unix and Win32 platforms, with Windows 2000 having built in support for Kerberos version 5 (TechNet, 2001).

3.7 E-mail Security considerations

As stated before, e-mail has often been used by students to hand in assignments. This has caused problems such as “disappearing” assignments, and low levels of confidentiality as assignments aren’t encrypted. This section will therefore look at some secure e-mail-based solutions that could be used for assignment submission.

Most secure e-mail solutions are based upon public-key cryptography. Unfortunately due to the diversity of the implementations, various problems arise (Bernstein et. al., 1996). The following section will hence highlight most of these issues, after which three implementations; namely, PEM, PGP and S/MIME will be discussed.

- **Certificate management**

Certificates are documents that contain identification information together with a public key, and are normally digitally signed by a trusted entity to prove the authenticity of the certificate. The recipient of an encrypted message uses the trusted entities public key to decode the digital certificate that is attached to the message. This results in the recipient obtaining the sender's public key and identification information, that is held within the certificate. With this information, the recipient can send an encrypted reply (Bernstein et. al., 1996; Webopedia, 1998).

Various implementations of public-key cryptography have to contend with the difficulties involved with the management of certificates, which are used for verification of signatures. Most of the issues involved are interpretability related, with different applications managing tasks such as certificate storage, retrieval and revocation differently (Bernstein et. al., 1996). This is especially important as secure e-mail systems such as PEM and PGP make use of digital certificates.

- **Trust models**

Different applications use different trust relationships between entities. PEM for example, uses a strict hierarchical approach while PGP uses a more informal bottom up approach (Bernstein et. al., 1996).

- **Support for multiple recipients**

Various secure e-mail systems don't properly support the sending of encrypted messages to multiple recipients. This is difficult because it requires part of the

message to be encrypted with all of the recipient's public keys so that they, in turn, can use their private keys to decrypt the session key and ultimately the message (Bernstein et. al., 1996).

- **Forwarding messages**

A person may want to forward a message that was encrypted and signed to other individuals or entities. It is therefore desirable to remove the encryption while retaining the signature as proof of its authenticity. Unfortunately, different applications may implement this differently, resulting in incompatibilities between various secure e-mail implementations (Bernstein et. al., 1996).

Three common methods of implementing e-mail security will be described below.

3.7.1 Privacy-Enhanced Mail (PEM)

PEM is a standard for exchanging private e-mail, as documented within RFC 1421-1424 (RFC or Request For Comments). PEM supports both symmetric and asymmetric key management, however asymmetric is more popular and will therefore be discussed (Hughes, 1995). PEM makes use of MD5 for digital signatures, and RSA and DES for encryption. Unfortunately the small key size that PEM uses makes it vulnerable to attack.

The main features provided by PEM are:

- **Originator Authentication**

Digital signatures ensure authentication.

- **Confidentiality**

Secret-key and public-key encryption ensures confidentiality.

- Integrity

Digital signatures are used to provide message integrity.

- No repudiation of origin

The sender of a message cannot repudiate having sent the message because the message is digitally signed with the sender's unique key.

PEM's certificates adhere to the X.509 standard, which is a standard format for public key certificates and Certificate Revocation Lists (CRLs) (Bernstein, Bhimani, Schultz & Siegel, 1996). PEM uses a strict certificate hierarchy, where **Certificate Authorities** (CA) vouch for the authenticity of users and their certificates (Figure 3.5). The Certification Authorities certificates are in turn signed by **Policy Certification Authorities** (PCA), who maintain various policies that govern operations and entry requirements into the PCA's domain. Finally the PCA's certificates are signed by a root certificate of the **Internet Policy Registration Authority** (IPRA) (Bernstein et. al., 1996).

PEM allows for a single message to be sent to multiple recipients by using a single session key to encrypt the data. Multiple copies of the session key are then encrypted with the public keys of the recipients. Unfortunately various factors have impeded PEM's acceptance with users. Firstly, users do not like the strict hierarchical structure that PEM uses. PEM also does not support MIME and therefore does not support non-text messages (Hughes, 1995).

Hughes (1995) mentions that critics of PEM complain that it does not provide enough privacy, as anyone can verify the sender's signature. This is due to the fact that the signature is not encrypted. Finally, PEM uses a complex name scheme, and does not allow for forwarded messages to retain their signatures (Bernstein et. al., 1996).

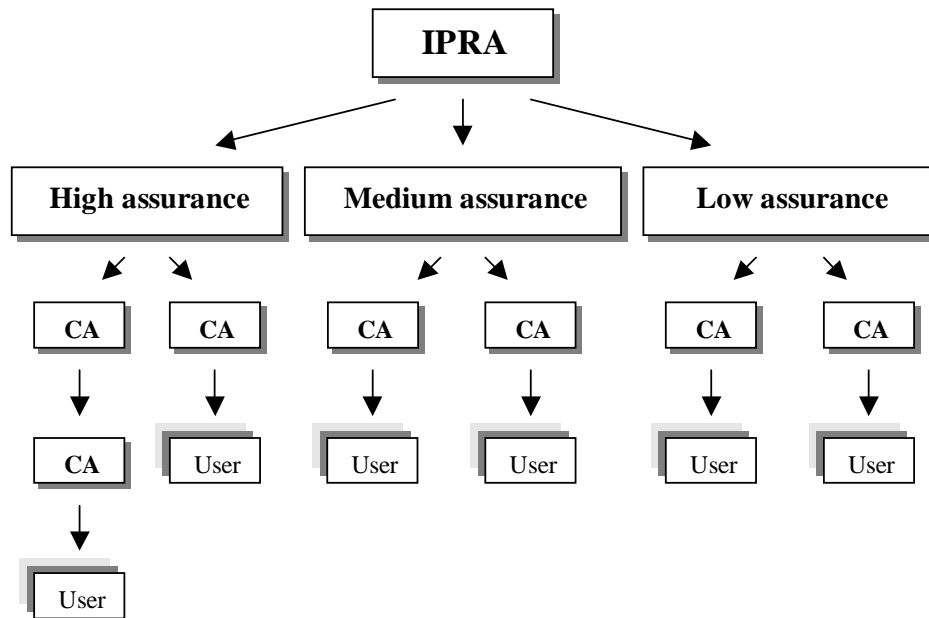


Figure 3.5 Displays the hierarchical tree structure that PEM uses
(Bernstein et. al., 1996).

3.7.2 Pretty Good Privacy (PGP) and OpenPGP

PGP was primarily created by Phil Zimmermann and is one of the most popular mail-encryption programs available. Like PEM, PGP supports both symmetric and asymmetric encryption, however the message formats are different and therefore incompatible. PGP also differs from PEM in that PEM uses a trust model, based on a hierarchy of certificate authorities, whereas PGP is not as rigid and is based primarily on personal decisions (Hughes, 1995).

Once PGP has been installed a public/private key pair are generated for the user. The public key is placed in a data structure known as a **public key ring**. The key also has a **user id** and **key id** associated with it. PGP then asks the user for a pass-phrase and uses it to generate a 128-bit IDEA key. The IDEA key is, in turn, used to encrypt the private key. The pass phrase and 128 bit key is then discarded and the encrypted private key is placed on a **private key ring** together with the owner's **user id** and a copy of the **public key**. Therefore even if the private key ring is stolen it is unusable

without the pass-phrase (Cooper, Goggans, Halvey, Hughes, Morgan, Siyan, Stallings & Stephenson, 1995).

A message between two participants would look something similar to Figure 3.6. Firstly, the sender uses his or her private key to create a digital signature, which is added to the message. The message is then compressed and encrypted with the session key. The receiver's public key is then used to encrypt the session key after which it is added to the message. Finally, the message is converted to an ASCII armour format and sent to the recipient.

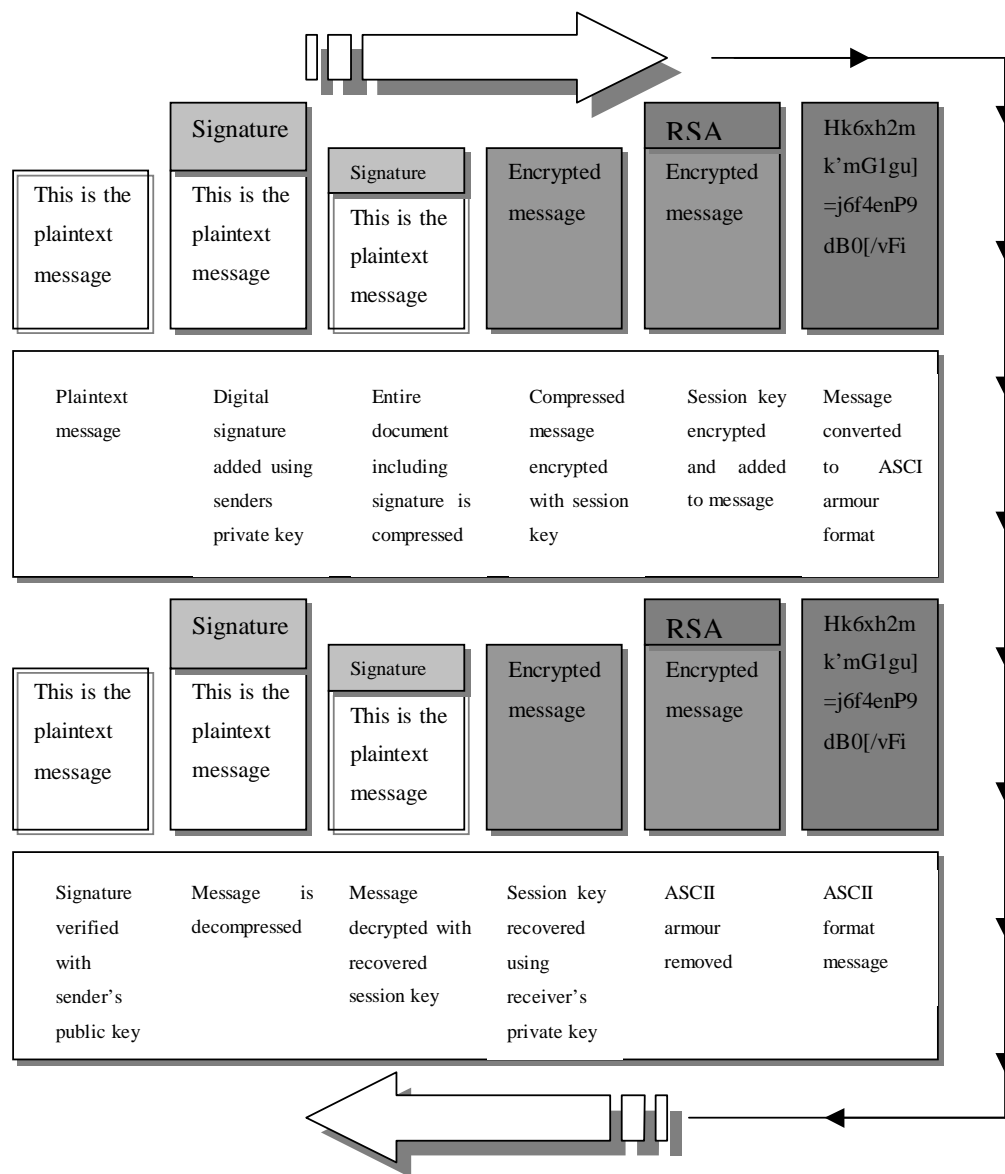


Figure 3.6 Illustrates how a PGP message is encrypted and decrypted.

The recipient firstly has to remove the ASCII armour and then use his or her private key to retrieve the session key. Once this has been performed the session key is used to decrypt the message after which the message is uncompressed. Finally, the recipient uses the sender's public key to verify the signatures authenticity.

PGP is superior to PEM due to the fact that it supports strong encryption algorithms such as IDEA, and because it does not require a trust centre. On the other hand, PGP's web of trust-based system can prove to be problematic, and although PGP itself is not a standard, OpenPGP is (SSH Communications Security Ltd., 1999a).

3.7.3 Secure Multipurpose Internet Mail Extensions (S/MIME)

In 1995, RSA Data Security released a specification for secure e-mail and named it Secure Multipurpose Internet Mail Extensions (S/MIME). S/MIME was designed to add security to e-mail messages in MIME format. S/MIME is superior to both PGP and PEM in that it uses a very flexible hierarchical certification approach (Bernstein et. al., 1996). The certificates use the X.509 format because of its acceptance as the standard for digital certificates. S/MIME uses a symmetric cipher to do the bulk of the message encryption, and a public-key algorithm for key exchange and digital signatures. The symmetric algorithms that are recommended by the S/MIME draft are DES, T/DES and RC2 (RSA Data Security, 1999). Unfortunately Counterplane Internet Security Inc. (2000) found that some implementations of S/MIME would not interoperate with one another when ciphers stronger than 40-bit RC2 were used.

3.8 Summary

It is essential to realise the importance and role that information security plays to ensure the continuation of services within an organization. Bruce and Dempsey's distributed security architecture can be used as a guideline to indicate how

information security can be attained within a distributed environment such as within an electronic submission and evaluation systems.

Security is especially important in an electronic assignment submission system, because the operation of such a system plays an important role in supporting the goals of an educational institution. This is especially true when electronic submission systems are used within distance educational environments, as the institutes are almost totally dependent upon the continued operation of the assignment submission systems. On the other hand even if the continued operation of an assignment submission system is not critical to the organisation, it still has to be considered secure as the assignment management system would contain confidential information such as student marks and details. It is therefore essential that the system be considered trustworthy. The technologies described in this chapter could be used to create such a secure environment. Chapter 4 will discuss some of the main criteria, other than security, that must be looked at when choosing or designing an electronic assignment submission and evaluation-based system.

Requirements of an assignment management system

4.1 Introduction

Chapter 3 discussed information security, its importance and the technologies that can be used to ensure a secure environment. This chapter will discuss the criteria that need to be considered when choosing or designing a product, which will be used to facilitate assignment management. This first section looks at the criteria for choosing a product in general. The section thereafter will focus on the criteria that are specifically relevant to an assignment management system.

4.2 Aspects to consider when choosing a solution

Numerous comparative studies have been undertaken to assist people to determine which products are best suited for their needs (Britain & Liber, 1999; Hazari, 1998; Kristapiazzi, 1998; FutureU, 2000). They did this by primarily focusing on the features that Course Management Software (CMS) products possess.

The following section consists of a subset of the features discussed in these studies. They were specifically chosen because this dissertation is only concerned with those features that contribute to the process of assignment submission, evaluation and feedback. Moreover this section will discuss how these features can specifically be used to help facilitated the management of student assignments. Quizzes and other

test-related assignment features will, however, not be covered as the dissertation is primarily concerned with programming assignments. Readers who want more information about such systems could however have a look at WWWAssign (Martin, 1997).

4.2.1 Platforms

Early on in the decision making process the platforms on which the proposed system is to run on must be identified (Hills 1997). Platform independence is especially important if the product will be expected to operate over a heterogeneous environment where different operating systems and hardware are used (Jones & Buchanan, 1996).

4.2.2 Standards and architecture

There are no “silver bullet” products that can cater for all of the users’ needs and therefore standards are needed to allow for additional products to work in conjunction with one another as seamlessly as possible (Bock & Marca, 1995). A proprietary-based solution may result in interoperability problems appearing in the future, as additional products are needed. Jones and Buchanan (1996) also note the use of standards promote platform independence.

4.2.3 Integrated or specialised products

Hills (1997) notes that two general types of groupware solutions exist, namely integrated and specialised solutions. Kaplan (1998) applies this specifically to software for developing web courses, differentiating between integrated solutions such as WebCT and specialised off-the-shelf component-based solutions. Integrated solutions also often possess many different features, making them appealing to the mass market (Hills, 1997). Another advantage of integrated solutions is that they present users with a common structure, look and feel, resulting in users being less

inclined to need help (Kaplan, 1998; Parkland College, 2000). Kaplan (1998) argues that the integrated solutions' common look and feel make it difficult for different institutions or classes to differentiate themselves. Integrated products are also generally very expensive, and consequently only large organizations can afford such products (Hills, 1997; Kaplan, 1998).

On the other hand, specialised products are much cheaper and may have certain features which the integrated products don't poses (Hills, 1997; Kaplan, 1998). Specialised products do, however, face the problem of integration, as they may not easily integrate and interoperate with one another. Therefore, the use of specialised products may be problematic if additional functionality is required at a later stage (Hills, 1997).

4.2.4 User- friendliness

The product must be user-friendly and intuitive to use for it to be easily accepted by the users. Complicated products might require users to go on training courses, wasting users' time and money. (Hills, 1997)

4.2.5 Training

The administrative staff might require extensive training in the particular product. The vendor should therefore be available for training and support when and where it is needed. Placing applications within web browsers can reduce training and support, as many of the users are already familiar with that interface (Hills, 1997).

4.2.6 Cost

Various factors contribute to the cost of a system. The most noticeable factor is the initial cost of the software. The product might also require various resources such as

servers, storage and bandwidth to be upgraded. Some products additionally require yearly licenses that incur greater costs over the long run (Kaplan, 1998). Users and administrative staff may also need training and complex products might be difficult to install and require help from consultants. According to Hills (1997) some statistics show that these additional costs may be some three to five times the amount of the actual software package.

4.2.7 Deployment and Fit

It is important to determine how the product will fit into the existing infrastructure, and whether it is required on all computers (Hills, 1997). This includes hardware and software such as the network infrastructure and operating systems used. The effort exerted in deploying a groupware product may additionally be substantial, as it may include the amount of effort needed to deploy software onto users' computers. The task is further complicated if different computers require different software configurations and components. One must also consider how much effort will be required if the software or hardware is to be upgraded. Many of these potential problems can however be averted in web solutions where the software runs on the server and web browsers are used to access it.

4.2.8 Stability

If the system is critical to the organisation's running, stability is an important factor to look into. Newly released software products may have many bugs and it might be wiser to use a wait-and-see approach (Hills, 1997).

4.2.9 Vendor support and strength

The software industry is a rough world and many companies rise and fall in a year. One should be cautious of new players to the scene, as young companies might not

exist in a few years. This may necessitate the move to another vendor's product. Such a move could be costly and time-consuming to an organisation, as the new product would have to be purchased and the staff would need to be retrained. The customer support that a vendor gives is also important, and choosing a product from a vendor with a well-established customer support record can only be advantageous to the organisation.

4.2.10 Security

As stated in Chapter 3, security plays an important role in an assignment management system. The following aspects were found to be of crucial importance, consequently these aspects should be considered when choosing or designing an assignment management system.

- **Integrity** - Integrity mechanisms ensure that the data has not been modified in any form during its storage or transit. Mechanisms must identify and correct or report incidents where the integrity of data was breached so that appropriate actions can take place (Bruce & Dempsey, 1997; CPS4001, 1998). In an assignment management system students should be presented with facilities to verify the integrity of their submissions.
- **Access Control** - Once an entity has been identified and authenticated, access control mechanisms ensure that an entity is only allowed to access authorised system resources. Access control, is usually determined by user ID together with a validated password and can additionally be based upon time dependency, the users role, data classification and so forth (Bruce & Dempsey, 1997; CPS4001, 1998).
- **Confidentiality** - Confidentiality services ensure that the data is protected from disclosure to unauthorised entities. Encryption is extensively used to provide confidentiality, especially over unsecured mediums such as networks (Bruce & Dempsey, 1997; CPS4001, 1998). Students are often concerned

about the public availability of their marks. Mechanisms that ensure the privacy of the students' information, including assignments should therefore be put in place.

- **Authentication** - Authentication mechanisms are used to verify the users are who they claim to be. Authentication is based on something an entity knows or owns such as a password combined with a fingerprint (Bruce & Dempsey, 1997; CPS4001, 1998).
- **Non-repudiation** - Mechanisms that support non-repudiation are used to prevent entities from denying that an action was taken or that a message was sent or received (Bruce & Dempsey, 1997; CPS4001, 1998). Non-repudiation mechanisms must be in place to prevent the often-occurring problem of students who claim they have submitted their assignments, but evaluators who claim they never received them.

4.3 Features required by a computer-supported assignment feedback system

Hills (1997) states that a list of requirements for the new system needs to be set up. The requirements should be placed in order of priority within a table together with the products that are to be considered. Once this is done, the table must be filled in, indicating whether a product meets the listed features. Missing features must then be identified and it should be determined whether the product can be modified to accommodate them or whether the product can be successfully used without the missing features. Furthermore it should be determined if the product's additional features may be of any use in the future or if they may cause hindrance. Finally the product's overall strengths and weaknesses should be identified, and where products have similar features, it may be necessary to go into further detail, comparing the individual features in detail (Hills, 1997). This is especially important, because as

Britain and Liber (1999) note, such tabular comparisons have the disadvantage that they do not indicate human factors such as the ease of use of the features.

Jones and McCormack (1997) note that assignment management (electronic and paper-based) can be considered to consist of four processes, namely **submission**, **storage**, **evaluation** and **feedback**. A fifth process, **communication** was however added by the author of this dissertation, because the existing model lacked a direct means of communication between students and lecturers. Since this dissertation is primarily concerned with programming assignment management, the following section will mainly discuss the issues relevant to programming assignments.

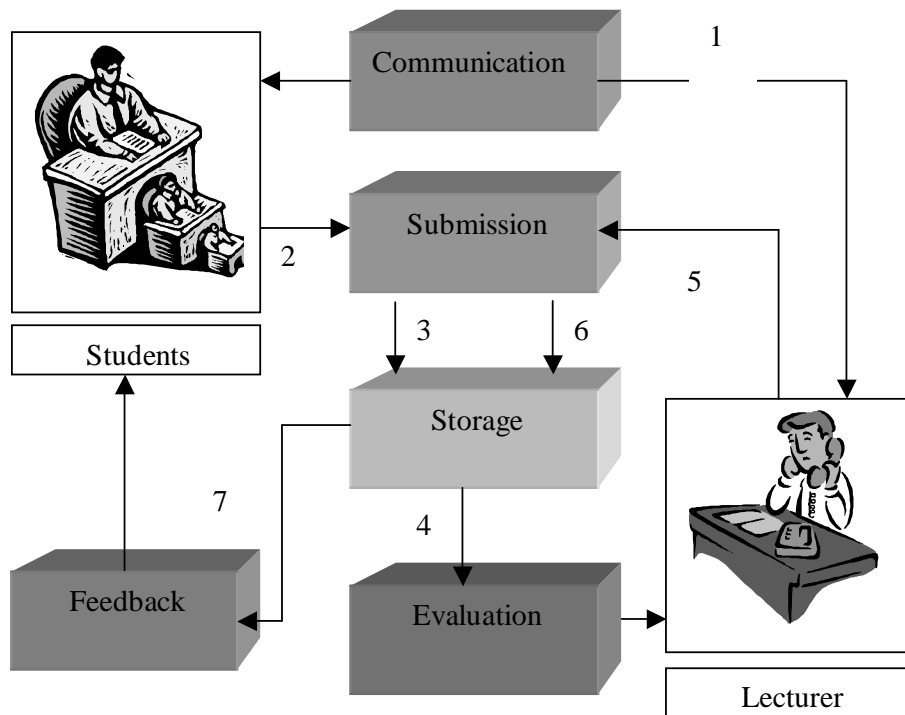


Figure 4.1 Displays the process of assignment management.

Figure 4.1 illustrates how these processes are related. The numbers in the diagram and within this section indicate the order in which actions take place. Firstly (1) the lecturer informs the students about an assignment that needs to be submitted. The students then submit their assignments (2). The submitted assignments are then stored (3). The lecturer then evaluates the assignments (4), and then submits them back into

the system (5). The evaluated assignments together with their marks are then stored (6), and finally the students retrieve their evaluated assignments to gain feedback on their assignments (7), and, if necessary, communicate with the lecturer (1). Most of the features that assignment management systems possess can be placed into these processes; the following section will therefore discuss the features within the aforementioned processes.

4.3.1 Communication

This refers to the process of communication between students and lecturers. A specific example would be the lecturer informing the students about assignments that need to be submitted, or a lecturer informed about the presence of submitted assignments. There are however problem areas surrounding automated notification of submission. Firstly the system could notify the lecturer via e-mail when an assignment is submitted. A problem with approach is that the lecturer may not want to receive large quantities of e-mail for a single submission. A workaround would be to notify the lecturer once all of the assignments have been submitted, however this would fail if some of the assignments were not submitted. Finally, the system could notify the lecturer on the due date, specifying how many students submitted their assignments. However in many cases the lecturer would log into the assignment management system on the same day, making the feature less worthwhile. The following sub-sections will discuss various features that could be used for these, and other purposes.

- **Course calendar/scheduler** – A course calendar displays when the components of the course are to be covered. It can additionally be used to mark major evaluation events that occur during the course such as tests and exams. The following questions can be asked when evaluating the usefulness of this feature.
 - Does it provide both personal and group calendaring facilities?
 - Does it have to-do lists?
 - Can it be accessed over the Internet?
 - Can tasks be assigned to other people?

- **Announcements** – Announcement tools enable instructors to notify entire groups of students of upcoming events such as tests and presentations.
- **Assignment reminders** – As the name implies this refers to tools that notify students of upcoming assignment due dates. Notifying the student via e-mail could do this.
- **E-mail**

Students and lecturers often use e-mail to converse with one another. It is also frequently used for the submission of assignments (Gibson, 1997; Nouwens & Towers, 1997; Arzt, 2000). It can be used to notify students of assignments, and provide a medium for feedback (Nouwens & Towers, 1997). The following questions should therefore be asked when e-mail is used as an active component during the submission and feedback of assignments (Hills, 1997).

- Does it provide integration facilities to calendaring, scheduling, conferencing and workflow applications?
- Does it provide for offline-reading and message creation?
- Can web browsers be used to access the mail?
- Is it easy to use?
- Does it support attachments?
- Does it provide filtering and searching capabilities?
- Does it provide management tools?

- **Newsgroups and Conferencing**

Although newsgroups and conferencing software could be used for the submission of assignments, they are generally used by students and lecturers to converse with one another. Students generally use newsgroups to post questions about assignments, or ask for help (Blanchard, 2000).

- Does it provide threaded discussions?

- Does it support moderators, polling, and anonymous postings?
- Does it provide public and private folders?
- Can HTML messages be created?
- Does it support attachments, and the inclusion of URLs in messages?
- Does it have text searching capabilities?
- Does it provide for offline-reading and message creation?
- Can new users enrol themselves?
- Does it require or include a Web server?

4.3.2 Submission

This refers to the process of routing assignments to and from students and lecturers. This was traditionally done by handing in printed assignments. Alternately, in an electronic-based system, this could be accomplished via the World Wide Web.

- **Assignment creation** – This refers to purpose-built facilities for the creation of assignments.
- **Multi-file assignment upload** – Some assignment solutions may consist of many files. This feature therefore enables students to upload multiple file assignments.

4.3.3 Storage

Traditionally evaluated assignments were given back to students. This meant that the students had the responsibility of storing their evaluated assignments. Electronic assignment management systems potentially allow for assignments to be archived for an indefinite period of time. This enables students and lecturers to have access to the assignments at all times.

- **Student submission storage** - The assignments are firstly stored once the students have submitted their assignments.
-

- **Evaluation assignment storage** – The evaluated assignments are also stored for later retrieval by the students.

4.3.4 Evaluation

This refers to the process of evaluating assignments.

- **Assignment annotation** - As the name suggests, this feature implies that a product has some facility that allows instructors to annotate assignments. Some of the more prudent questions that should be asked are:
 - Does it provide in-context annotation?
 - Does it have graphics capabilities, allowing the evaluator to “scribble” on the assignment?
 - Do the students require special software to view the annotations?
- **Online evaluation** – Online evaluation tools allow evaluators to evaluate student assignments via a web browser. This however requires the evaluator to be connected to the Internet during evaluation.
- **Offline evaluation** – Offline evaluation refers to the use of evaluation products that don’t require the use of the Internet or network-based facilities for evaluation to take place.
- **Multi-file assignments** – Programming assignments often consist of multiple files, therefore the software that is used for evaluation should be able to open multiple files at a time.
- **Source code analysis** – This refers to tools that analyse source code and report on the quality of the code. Zin and Foxley (2000) note that these tools can provide educators with feedback about the overall strengths and weaknesses of a class.

They then go on to state that these tools can be of great assistance when marking student work.

- **Automated program testing** – Automated program testing tools can test submitted programs against predefined data inputs and outputs.

4.3.5 Feedback and retrieval

Jones and McCormack (1997) call this component the “result” component, stating that it is responsible for returning the students’ results. The author of this paper however refers to it as the Feedback and Retrieval component, as it is responsible for providing students with feedback on their assignments. This may be in various forms, such as, by supplying students with marks and detailed commentary. This could also refer to the process of lecturers getting feedback concerning students’ progress etc.

- **Access to own grades** - Students who have direct access to their own grades can better track the progress of their learning (FutureU, 2000).
- **Automatic assignment progress tracking** – This includes tools that notify the evaluators as to which assignments have, and have not been evaluated. It may also include facilities to notify evaluators of late submissions.
- **Summative calculations** – This tool could automatically calculate the class averages, etc. based on the marks stored within the system. Students can also use this feature to determine their standings within the class.

4.3.6 Administrative features

This section will briefly describe the administrative features that are needed for an assignment management system to function, but that don’t fit into any of the five core

processes of assignment management. This section will not however discuss these features in great detail since they are beyond the scope of the dissertation.

- **Automated registration** – Systems that support automated registration do so by being integrated into the existing registration system. As FutureU (2000) notes this is an extremely sought after feature and normally requires the system to be Open Database Connectivity (ODBC) compatible. ODBC provides a set of drivers that enable applications from different vendors to access each others data sources no matter what format that data sources are in.
- **Batch registration** – This feature allows administrators to upload and import user data into the system that has been exported from an existing registration system (FutureU, 2000).
- **Batch user removal** – This enables administrators to remove entire classes at a time.
- **Assignment file management** - File management is an important feature when dealing with assignment management since a submitted assignment could consist of many files, and a class of many students. Evaluators may make use of **batch upload** facilities to upload multiple evaluated assignments, or evaluated assignments that consists of many files. Similarly **batch delete** facilities are also very useful.

4.4 Summary

This chapter discussed some of the major factors that should be considered when selecting a system that could be used to support assignment management. This list of criteria is by no means “complete” and should therefore only be seen as a guideline to help the user in the selection or design process. Chapter 5 will evaluate various

software packages that could be used to manage assignments. This will be done by making use of the list of features presented in this chapter.

Evaluation of currently available products

5.1 Introduction

The previous chapter highlighted some of the criteria that must be considered when choosing or designing a product that would facilitate assignment management. This chapter will make use of the criteria identified in Chapter 4 to determine the viability of making use of existing commercially available software to create an electronic-based assignment submission and evaluation system.

Various factors such as time constraints, hardware requirements and the availability of evaluation products made it impractical to individually evaluate the integrated solutions. Therefore the information pertaining to the functionality of the various products was obtained by making use of the following product evaluations: Britain & Liber, (1999); Hazari, (1998); Kristapiazzi, (1998); FutureU, (2000). The software solutions that could be used to evaluate assignments were, however, personally installed and evaluated by the author of this dissertation.

This chapter will firstly evaluate software products that handle the actual management of assignments, i.e. the submission, storage and retrieval aspects of assignment management. This will then be followed by focusing on software that can be used for the evaluation of the assignments. The products' main functions will firstly be briefly discussed, specifically noting how they handle students submitting file-based assignments. The products will finally be evaluated in tabular form, from which their suitability for use in assignment management will be determined.

5.2 The use of class management software to provide assignment management

This section will only discuss commercially available solutions that could be used to manage assignments. This is because even though dedicated assignment management systems exist, they are generally custom built systems, developed by distance education universities, and not available to the public.

5.2.1 WebCT

WebCT (2000) is a highly integrated solution that can be used to facilitate the creation of World Wide Web-based educational environments. WebCT can create entire online courses, or can be used to publish supplementary course materials for use in existing courses (Kristapiazzi, 1998). Kristapiazzi (1998) notes that there are three aspects to WebCT:

- **Presentation tools** - The presentation tool is used by course designers to determine the look and feel of the course pages. Course designers can use it to change the layout of pages, colours, counters etc.
- **Student tools** - WebCT comes with a host of student tools that can be incorporated into any course. They include communications and collaboration tools, self-evaluation tools, a course calendar tool and more.
- **Administrative tools** - These tools are used to manage the delivery of course material. They allow lecturers to track the progress of students, manage student grades and create reports. Lecturers can even make use of the on-line quiz creation tools that can be automatically evaluate the quizzes.

Assignment file submission

Students can submit their assignments by using WebCT's assignment upload facility.

This provides the students with a form that requires them to select the appropriate file and destination folder. To upload multiple files this process has to be repeated for each file.

EDPY 202: Technology Tools for Teaching and Learning
Home > Upload Assignments > Manage Files > Upload File

Upload File

Note: file size and network connection may cause this process to take several minutes.

Filename Browse...

Destination folder

Figure 5.1 Assignment upload form.

Once the files have been uploaded the students can verify that the upload was successful by looking at the file size as indicated in Figure 5.2

Folders and Files

Click on a folder below to view its files.

<input type="checkbox"/>		jedavies		
<input type="checkbox"/>		jedavies-m12.sit	8412	Sat Aug 26 2000 09:51:00

Figure 5.2 Displays the uploaded assignment file details.

5.2.2 Web Course in a Box (WCB)

WCB (2000) provides facilities for the creation of secure web sites for courses, faculty home pages, and student management and tracking. When a user logs into WCB he or she is presented with a customisable course homepage which contains links to the various features of WCB. Some of the most relevant features that WCB includes are:

Gradebook and Assignment Management

- Supports assignment submission and tracking;
- Option to view on the course calendar;
- Individual student view and online assignment submission;
- Ability to download grades for further analysis; and
- Instructors can grade, comment, and return assignments to students.

Whiteboard and Chat

- Integrated Whiteboard and Chat tools; as well as
- Supports archiving of Whiteboards, allowing students to search previous tutoring sessions.

Calendar

- Calendar items such as assignment due dates can be linked to the aforementioned Gradebook facility; as well as
- Learning material such as lessons, quizzes and forums can be linked from the Calendar.

Assignment file submission

Web Course in a Box allows students to submit multiple files, however, it seems to be similar to the method that WebCT uses, namely the users must submit one file at a time. There is no facility for a lecturer to annotate an assignment online, however.

5.2.3 TopClass

TopClass (WBT systems, 2000) is a relational database-driven system, which can be used for web-based course delivery, or merely to assist live classes. TopClass permits any web-based content to be included in courses and be delivered over the Internet or a Local Area Network (LAN) (WBT systems, 2000). Asynchronous communication and information sharing is supported through e-mail, threaded discussion groups and

Bulletin Board System (BBS) file exchange. Some of the main features of TopClass include:

- **Open Content** - TopClass supports any web-compatible applications to deliver media such as animation, streaming video, audio and videoconferencing. In addition to this TopClass Publisher can be used convert Microsoft Office documents or web pages into learning content for use in TopClass (WBT systems, 2000).
- **Personalised Courses** - TopClass can automatically add or remove content to courses to meet the user's needs. It also allows for the content to be directly modified for individual users.
- **Training Module Library** - The module library is used to organise courses. TopClass also enables the sharing of learning material and media content between many classes.
- **Testing** - TopClass can be used to give individual or random tests that give each student a different set of questions. The tests can also be essay style and students can even upload files which the lecturer can review. In addition to this, TopClass sports a self-testing facility and enables both students and lecturers to track their progress (Britain & Liber, 1999).

Assignment file submission

The submission of assignment files can be accomplished by making use of TopClass's Testing component. This would require the lecturer to set up a test that includes file attachments. TopClass also permits files to be attached to messages and discussion group postings. However, it appears that most TopClass users generally use e-mail for the submission of assignments. The lecturer cannot mark and give feedback to students, as would be required for programming assignments, however.

5.2.4 Lotus Learning Space

Lotus LearningSpace (Lotus Corporation, 1999), is one of the leading solutions for creating environments of collaborative learning and knowledge management. It facilitates the development, management and delivery of on-line training and education. LearningSpace enables the creation and delivery of asynchronous instructor-facilitated and self-paced learning. Most of the information discussed in this sub-section was obtained from on Lotus LearningSpace's product blurb.

LearningSpace runs on Lotus Notes via the Lotus Domino web server. It is comprised of five interconnected modules, each of which is a Lotus Notes database (Lotus Corporation, 1999). Students have access to the **Schedule**, **MediaCenter**, **CourseRoom** and **Profiles** modules, while course designers and lecturers have additionally access to the **Assessment Manager** and **LearningSpace Central** modules.

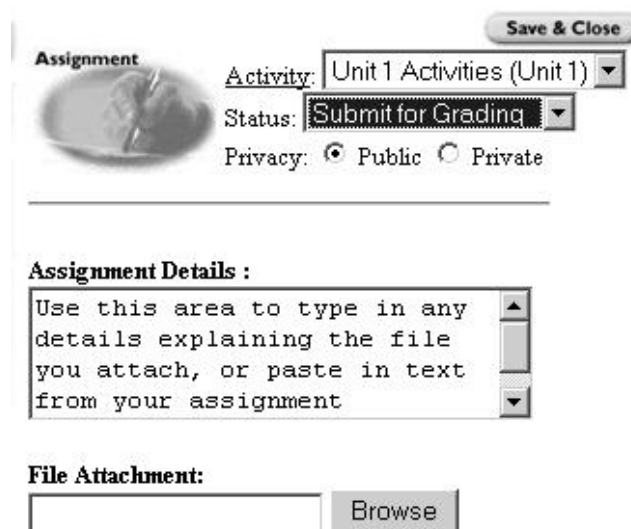
- **Schedule** - provides students with a structured approach to the assignments, materials, and assessments. Through the schedule, students can link to everything required to complete their course.
- **MediaCenter** - allows immediate and searchable access to all materials for the course as they are made available by the instructor.
- **CourseRoom** - hosts collaborative interchange between groups of students and, students and instructors.
- **Profiles** - helps students and instructors get to know their classmates, to form productive teams and to network outside of the course.
- **LearningSpace Central** – consists of management tools that lecturers use to manage the course. It additionally allows students and lecturers to create a local

copy of the course on their workstations, however, this requires them to have Lotus Notes Client installed on their computers.

- **Assessment Manager** - is used to assess and track a student's knowledge and progress.

Assignment file submission

LearningSpace's assignment file submission feature (Figure 5.3) is accessed via the CourseRoom. A student must first select the appropriate **activity** (assignment) from a dropdown box. The student must then select the type of submission from the status dropdown box. The student has three options. He, or she, can choose to submit the assignment for grading, for review or for commentary. The student can additionally decide if the submission will be publicly viewable or only meant for his or her evaluator. The Assignment details text box can be used to add commentary to the submission, while the file attachment box allows the student to attach a single file to the submission. LearningSpace does not provide adequate annotation and feedback features that are necessary with programming assignments.



The screenshot shows a web form for submitting an assignment. At the top right is a "Save & Close" button. Below it, the "Assignment" section includes a "Activity:" dropdown menu with "Unit 1 Activities (Unit 1)" selected, a "Status:" dropdown menu with "Submit for Grading" selected, and "Privacy:" options with "Public" selected and "Private" unselected. A horizontal line separates this from the "Assignment Details:" section, which contains a text area with the instruction: "Use this area to type in any details explaining the file you attach, or paste in text from your assignment". Below that is the "File Attachment:" section, featuring a text input field and a "Browse" button.

Figure 5.3 LearningSpace's assignment file submission form.

5.3 Comparison of the integrated solutions

Table 5.1 compares the features of popularly used classroom management solutions.

	Learning Space	WebCT	WCB	TopClass
Platforms				
Unix	√	√	√	√
Windows NT	√	√	√	√
Mac	X	X	√	√
Security				
Access Control	√ (user/pass)	√ (user/pass)	√ (user/pass)	√ (user/pass)
Authentication	√ (user/pass)	√ (user/pass)	√ (user/pass)	√ (user/pass)
Integrity	√	√	√	√
Confidentiality	√ (SSL) 3 rd party	√ (SSL) 3 rd party	√ (SSL) 3 rd party	√ (SSL) 3 rd party
Non-repudiation	X	X	X	X
General				
Training	√	√	√	√
Support	√	√(free)	√(free)	√(\$3000/year)
Annual license fee (unlimited number of students)	?	\$3000 year 1 \$3000 year 2+	\$4000 year 1 \$2000 year 2+	\$6000
Communication				
Calendar	√	√	√	√
Noticeboard	√	√	√	√
Assignment reminders	?	X	X	?
E-mail	√	√	√	√
Newsgroups	?	?	?	?
Submission				
Assignment creation tools	√	√	√	√
Multiple file submission	√	√	√	√
Submission method				
• E-mail	√	√	√	√
• Web-interface	√	√	√	√
• FTP	?	?	?	?
• Mapped drive	?	?	√	?

	Learning Space	WebCT	WCB	TopClass
Storage				
Submitted assignment storage	√	√	√	√
Evaluated assignment storage	?	√	?	?
Evaluation				
Assignment annotation	X	X	X	X
Online annotation	X	X	X	X
Offline annotation	X	X	X	X
Multiple file assignments	X	X	X	X
Programming specific features				
<ul style="list-style-type: none"> • Source code analysis 	X	X	X	X
<ul style="list-style-type: none"> • Automated program testing 	X	X	X	X
Feedback and retrieval				
Access to own grades	√	√	√	√
Assignment progress tracking	?	√	√	?
Summative calculations	√	√	√	√

Table 5.1 Comparison of the features of the classroom management solutions.

Chapter 4 discussed the various issues and features that need to be considered when choosing or designing an assignment management system, one can determine the core requirements that an assignment management system must have by comparing the features discussed in Chapter 4 with the objectives set out in Chapter 1. These requirements can be summarised as follow:

The first objective set out in Chapter 1 was to determine how the flow of programming assignments between students and lecturers could be streamlined. A solution should therefore be capable of facilitating the **submission, storage and retrieval** of assignments.

Another objective set out in Chapter 1 was to ascertain how rich, intuitive feedback could be attained in programming assignments that are evaluated electronically. However, as stated before, programming assignments often consist of many files. Therefore any software solution used to evaluate and generate feedback from programming assignments should be capable of handling **multiple file assignments**. Furthermore because source code is complex, **in-context annotation** is essential.

The third objective noted in Chapter 1 was to determine how the submission, storage and retrieval of assignments could be secured. Chapters 3 and 4 noted that for this to take place, **access control, integrity, confidentiality, nonrepudiation** and **authentication** need to be ensured.

All of the solutions contain tools such as e-mail, which could be used for the submission of assignments that consist of many files. It is however unclear whether the products support submissions based on File Transfer Protocol (FTP). The products tend to concentrate on quiz and short question-based assignment facilities that make use of forms, and appear to have put little effort in implement file-based assignment submissions. In addition to this, the systems such as WebCT that provide uploads via a web-based form, require files to be uploaded one at a time, wasting students time. The solutions additionally don't provide the needed annotation and security-based features. Specifically all of the products appear to lack non-repudiation-based functions, which provide students of proof of their assignment submission. Finally the systems require Secure Sockets Layer (SSL) to provide confidentiality, however they don't include SSL on their default installation, making them vulnerable.

5.4 Stand-alone evaluation software

As stated before, most of the integrated software packages do not offer adequate tools for the evaluation of student assignments. This section will therefore take a look at some of the more popular software packages that do offer such features. The products' security features will not be discussed because all of the discussed products

are standalone-offline evaluation solutions. Security must therefore be enforced on the computers on which the solutions are installed and run from.

5.4.1 Microsoft Word 97 and Word 2000

Microsoft Word 97 and Word 2000 are general-purpose word processing tools. They do, however, contain annotation and reviewing features. To utilise the reviewing feature, the users must first enable a feature called change tracking. This enables other people to directly edit the document. Text that is “removed” is shown as crossed out, while additions are underlined (see Figure 5.4). Microsoft Word also allows for multiple parties to partake in the editing of a document, by colour coding each participant’s editorial contribution.

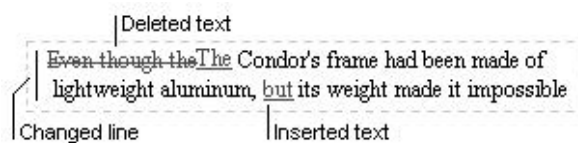


Figure 5.4 Illustrates the use of Words edit tracking ability.

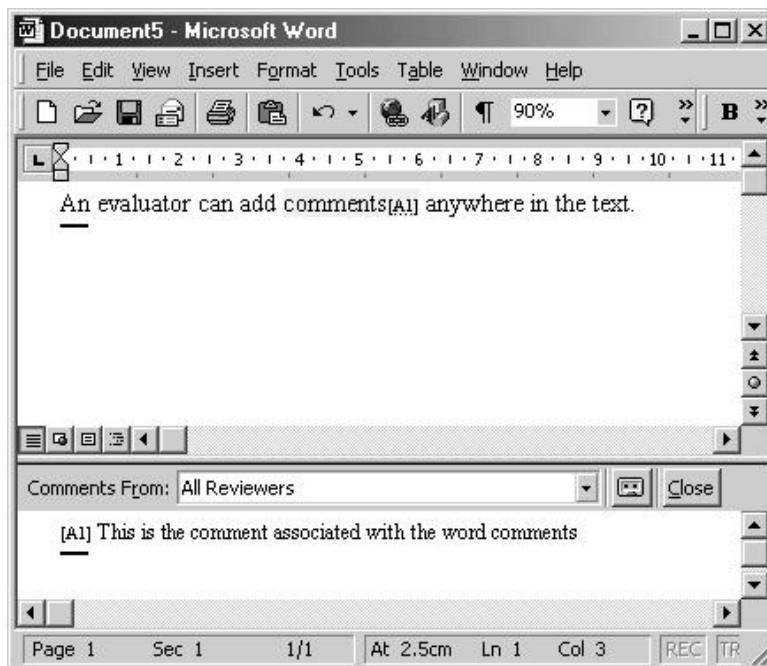


Figure 5.5 Displays Microsoft Words commentary feature.

Reviewers can also make annotations by making use of Word's *"Insert Comment"* feature. This is done by selecting the relevant text and then by clicking on the insert comment icon. Once a comment is added, it is numbered and stored in a separate comment panel. Word then embeds a comment reference mark in the document and colours the selected text to indicate the presence of a comment (see Figure 5.5).

Output formats

Microsoft Word supports most of the commonly used word processing file formats. It additionally supports the standard HTML and Rich Text file formats.

General comments and suitability of the product

Microsoft Word is a very powerful general-purpose word processor. It provides all of the basically required annotation features, and file formats. Word can be easily used to evaluate essay style assignments, however it will be difficult to use it with programming assignments. This is because it is difficult, although not impossible, to manage assignments that consist of multiple source code files.

5.4.2 Markin

Markin (Holmes, 1999) is one of the few specialised text editors available that is designed for the marking of electronic documents (Figure 5.6). Once the text has been loaded into the program the evaluator must identify the errors and add any relevant commentary.

The panel on the right hand side contains a set of buttons, each one signifying a specific type of error. These buttons can be customised, and entire button sets can be defined. A button set is a group of buttons that is used for a particular purpose, for example, Markin comes with a built in English essay marking button set.

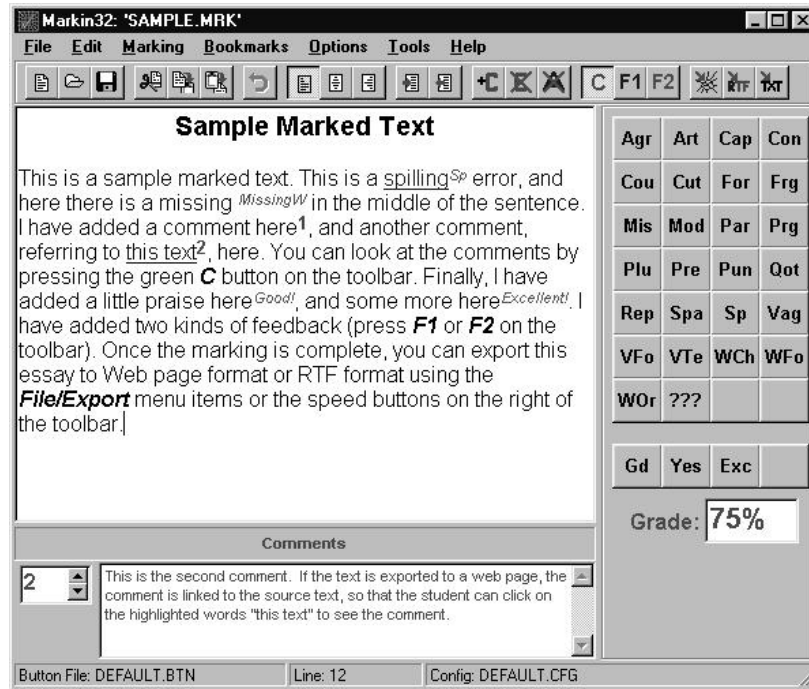


Figure 5.6 Shows a sample evaluation (Holmes, 1999).

Annotation is performed by selecting the relevant text and clicking on the appropriate button within the panel to the right. This will result in the selected text being automatically coloured and underlined. A predefined superscript short description of the annotation will also be added next to the selected text (Figure 5.6). Markin can additionally be used to add customised commentary to an assignment. This is done by selecting the relevant text and by clicking on the comment button. The evaluator can then add customised commentary in the comment frame. This commentary is “linked” to the actual assignment text by means of a number in superscript (Figure 5.6).

Output formats

A document that was evaluated within Markin can be exported to plain text, Rich Text Format (RTF) or HTML. Exporting to HTML produces a formatted document that can be viewed within a web-browser. Each error is a hyperlink to a description of the error as well as any other comments. RTF files in turn can be viewed within most modern word processors and includes the colours and commentary.

General comments and suitability of the product

Markin is probably one of the most powerful commercially available assignment annotation tools. Its high degree of customisability enables its use within simple programming assignments. It does however have a few limitations that limit its suitability for some programming assignments. Firstly, it can only be used to evaluate one document at a time. This is problematic because modern day program source code consists of many interrelated files. Another potential problem with Markin is that it does not provide a means of identifying the owner of the assignment. This could however be overcome by forcing students to embed their student numbers in all of their files.

5.4.3 Mindtrail

Mindtrail (2000) describe their products as a timesaving assessment tool for teachers, academics and training professionals. Mindtrail can be used to provide consistent marking and feedback across multiple evaluators by capturing assessment criteria. It does not support direct editing of source documents (i.e. assignments), but can be rather thought of as an advanced marksheet or reporting tool. The use of Mindtrail involves three steps:

- Step 1: Build a custom “marksheet” with the Knowledge Tree;
- Step 2: Use the “marksheet” to create assessments; and
- Step 3: Produce reports.

These steps are described briefly below.

Step 1: Build a custom “marksheet” with the Knowledge Tree

The first step is to create a knowledge tree (Figure 5.7). A knowledge tree is a

predefined set of marking criteria that are used to evaluate an assignment. The criterion can also have marks and feedback comments associated with them. The knowledge tree can be modified at any time. If during marking the evaluator discovers that some criteria should have been added or modified, he or she can make the needed changes. Mindtrail will then, if necessary, automatically update all previous assessments to reflect these changes.

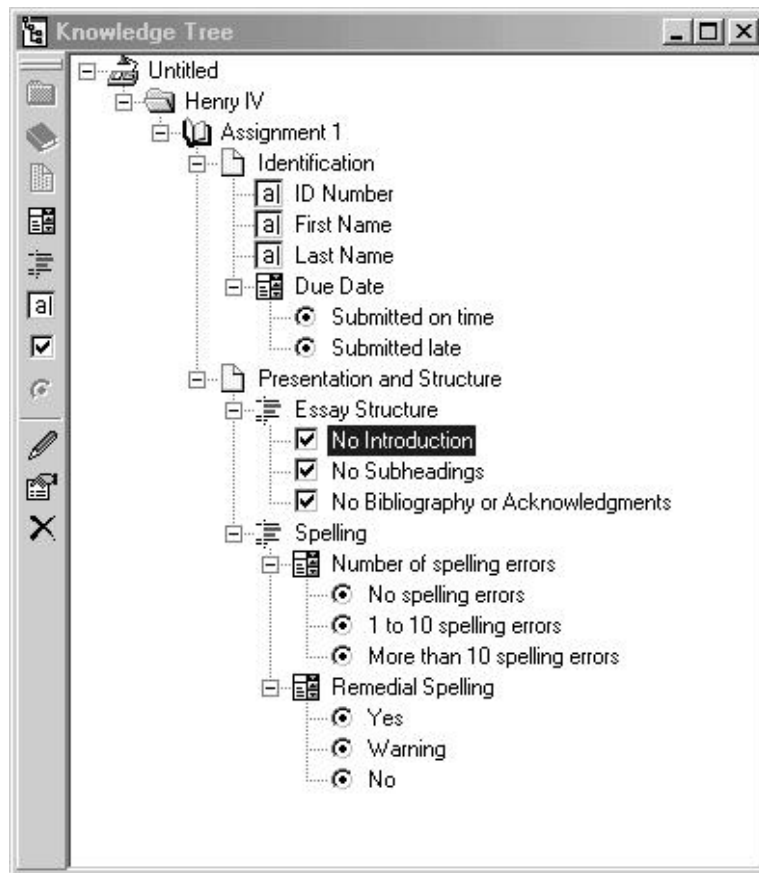


Figure 5.7 Shows MindTrail's knowledge tree.

Step 2: Use the “marksheet” to create assessments

Once a knowledge tree has been defined, the evaluator can generate an assessment for each student in the class with each assessment drawn from a single common set of marking criteria in the Knowledge Tree. The evaluator merely has to select the appropriate criteria and marks will be automatically deducted as need be (Figure 5.8). Mindtrail also allows evaluators to add customised commentary and mark allocation

for any of the predefined criteria. This allows the evaluator to reward and punish students for exceptionally good or bad assignments.

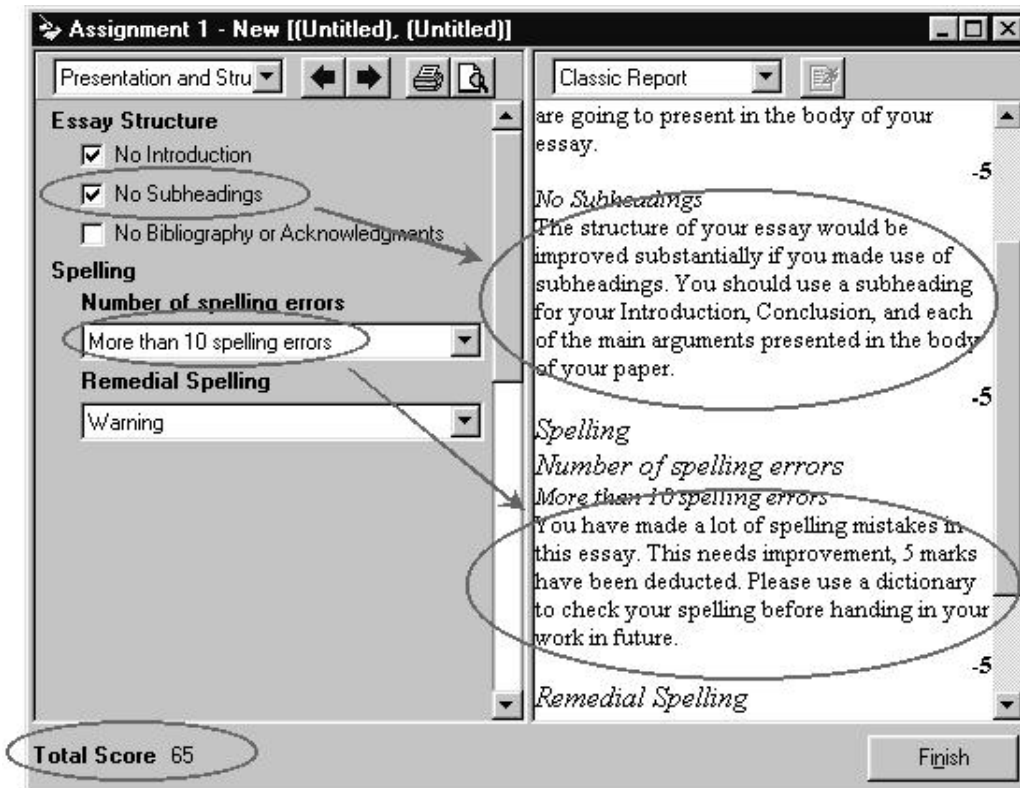


Figure 5.8 Displays how the evaluator fills in the appropriate criteria during assessment (Mindtrail, 2000).

Step 3: Produce reports

Once the assignments have been evaluated, additional reports can be generated. For example, reports that contain feedback or comments can be printed, student marks can be tallied, and the overall performance of the class can be determined and analysed.

Output Formats

Mindtrail makes use of its own project file format. It does not provide any tools to

export its reports to a commonly used file format such as HTML, therefore it will be difficult to integrate Mindtrail with any third party application.

General comments and suitability of the product

Mindtrial is arguably one of the most powerful assignment marksheet or reporting tool currently available. While it might prove to be very successful in essay style assignments, it has only limited usefulness within programming assignments. This is mainly due to its lack of any in-context annotation features, which are essential for programming assignments. Another reason why it can't be used in an electronic submission system is that it does not support any file types that could be viewed by students, relying entirely on printing. Mindtrail could however have been used in conjunction with an annotation tool such as Markin, if it had an HTML or text-based reports generation tool.

5.5 Comparison of the assignment evaluation solutions

	Microsoft Word 97	Markin	Mindtrail
Assignment annotation	√	√	X (marksheet)
Online annotation	X	X	X
Offline annotation	√	√	X
Multiple file assignments	X	X	√
Programming specific features			
<ul style="list-style-type: none"> • Source code analysis 	X	X	X
<ul style="list-style-type: none"> • Automated program testing 	X	X	X

Table 5.2 Comparison of the features of the assignment evaluation solutions.

Table 5.2 displays an overview of the features that the standalone evaluation tools support. None of the evaluated products were online, or web-based solutions, meaning that the lecturer does not have to be continually connected to an assignment storage server to evaluate the assignments. The products' security features were therefore not

taken into consideration. This would however not have been the case if online evaluation solutions were evaluated, as they would have to conform to the security requirements shown in Table 5.1.

Microsoft Word and Markin have annotation features, thus offering adequate levels of feedback. Mindtrail uses marksheets for feedback and is therefore incapable of producing in-context commentary to assignments. Unfortunately Word and Markin can only evaluate a single file at a time. This is inadequate for programming assignments which often consist of many files. The products therefore, do not provide the functionality that is required for evaluating programming assignments.

5.6 Summary

The objective of this chapter was to look at the viability of using commercially available software to provide an electronic-based assignment submission and evaluation system. This was done by firstly looking at highly integrated products and then by focusing on software that can be used for the evaluation of assignments. The reader should be able to conclude that while the class management systems discussed in this chapter are quality products, they lack the ability to provide adequate annotation and security (non-repudiation) features. The evaluation software that were discussed are popular, however, they lack certain features that are needed within a programming environment. In particular the evaluation software lack multi-file support. The integrated solutions and the standalone evaluation tools also generally don't integrate with one another. It can therefore be stated that the current solutions do not provide all of the needed features that are needed within programming classes. Chapter 6 will focus on the development of a model that supports electronic assignment submission and evaluation.

Chapter 6

The development of a model to support the management of assignments

6.1 Introduction

Chapter 4 discussed a number of the main selection criteria, when choosing, or designing, a solution that will be used for the electronic management of programming assignments. Chapter 5's objective was to determine whether existing solutions could be used for assignment management; this was done by taking the criteria mentioned in Chapter 4 into consideration. Chapter 5 found the currently available commercial solutions to be inadequate for use with programming assignments. They specifically lacked the required security and feedback features needed. The objective of this chapter is thus to devise a secure CMC-based submission and evaluation model that can be used to support the management of programming assignments.

Chapter 4 noted that assessment management (electronic and paper-based) consists of four processes, namely **submission**, **storage**, **evaluation** and **feedback** (Jones & McCormack, 1997). A fifth process, **communication** was added and illustrated in Figure 4.1, by the author of this dissertation.

Figure 6.1 is based on Figure 4.1, and refers to electronic assignment management. A sixth process, **management**, which is responsible for the creation of assignments, and the administration and management of users and classes has been added. The numbers in the diagram indicate the order in which actions take place. The initial step in using an assignment management system is adding and setting up classes and users. Once this is done the process of assignments submission and feedback can take place.

Firstly (1) the lecturer creates a new assignment, and informs the students about it (2). The students then submit their assignments (3). The submitted assignments are stored (4). The lecturer evaluates the assignments (5), and submits them back into the system (6). The evaluated assignments are stored (7), and finally the students retrieve their evaluated assignments to gain feedback on their assignments (8). Furthermore communication between the lecturer and student (2) should be made possible by the system at all times.

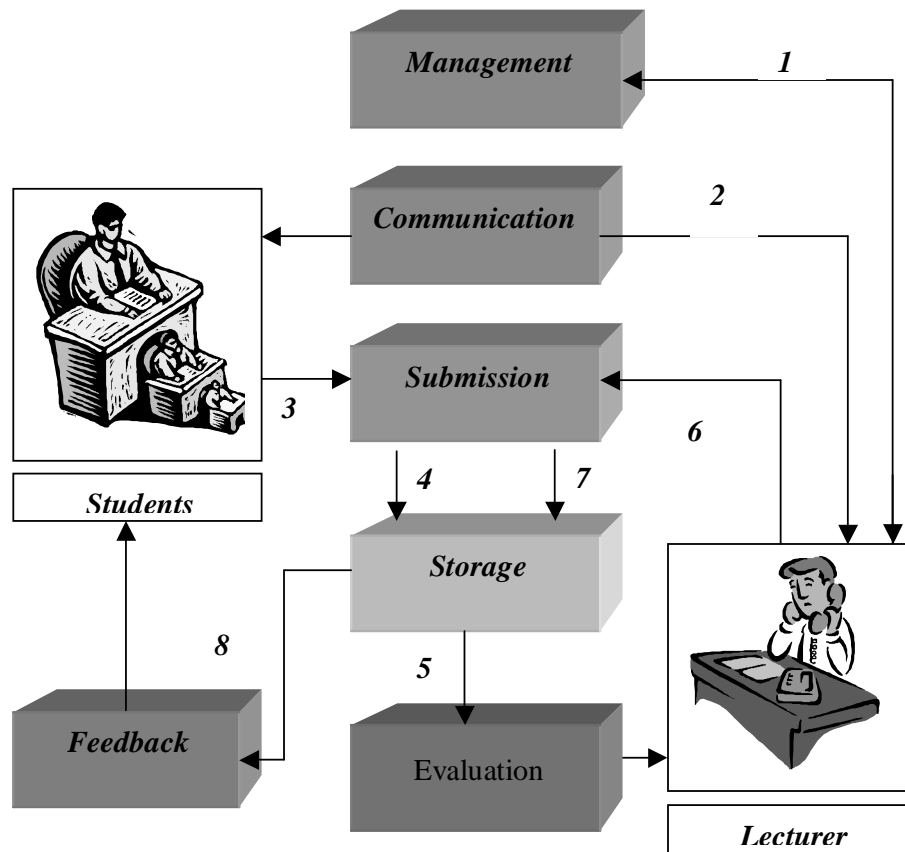


Figure 6.1 Displays the process of assignment management.

6.2 A Programming Assignment Management Model (PAMM)

The following section will discuss a conceptual model that can be used for the management of programming assignments. This will be done by describing the basic

requirements and responsibilities of each component, and the possible implementations for each of the components. It should however be stated that many of these components can be combined to form larger processes. This would, in many cases, automatically occur, when such a model is implemented, as many of the processes are closely related. (For example an e-mail server could be used as the submission and storage component.) These new processes would, however, still have to maintain the basic requirements as set out by their sub-components.

The Programming Assignment Management Model (PAMM) displayed in Figure 6.2 mimics the processes of assignment management, and therefore consists of the components illustrated in Figure 6.1 namely, **submission**, **communication**, **storage**, **evaluation**, **feedback** and **management**. The following section will describe these components in detail.

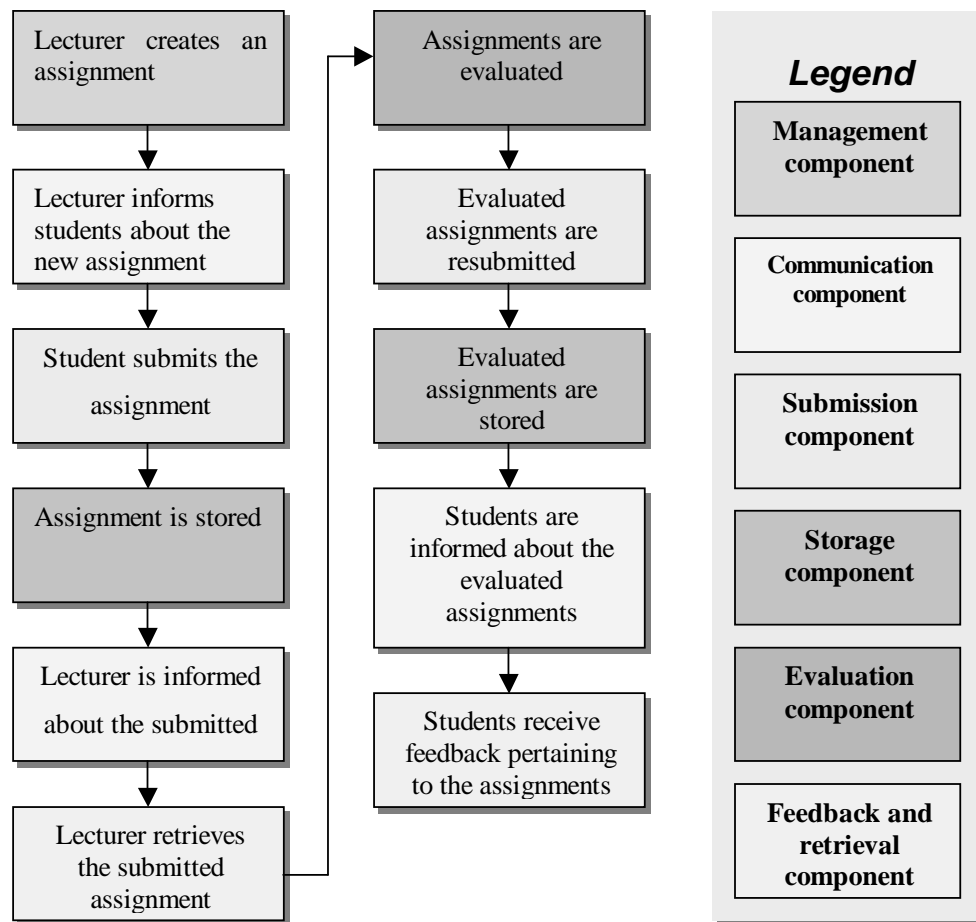


Figure 6.2 Shows the flow between PAMM's components.

6.2.1 Management component

The management component has two major responsibilities, firstly it is responsible for the management of assignments, and secondly it is used for the management of users and classes. The component's assignment management responsibilities include the creation, modification and deletion of assignments. The lecturer would therefore use this component to add new assignments, and, for example, if need be, change the assignment due dates etc. The component is additionally responsible for the administration of users and classes. Here the lecturer could use it to add users to and remove users from his or her class etc.

Some of the issues that need to be considered when implementing this component, is who is responsible for the administration of students. Will this be the responsibility of the lecturers or a special administrative user? Numerous management and administrative features, such as batch user addition and deletion, should also be considered as discussed in Section 4.3.6.

6.2.2 Communication component

The main purpose of this component is to inform the students about assignments that need to be submitted and lecturers about the presence of submitted assignments. Lecturers would therefore use the component to inform the students, that they have to submit a particular assignment. In the traditional classroom environment, this was done by informing the students verbally about upcoming assignments. In contrast, in an electronic environment this could be done via electronic notice boards, e-mail etc.

The Communication component could additionally be used to facilitate communication between students and lecturers. That is to say, students could use the component to ask lecturers for help with certain assignments, while lecturers could

potentially use it to reach classes. Chapter 4 discusses various technologies that can be used to implement this component, such as e-mail and newsgroups.

6.2.3 Submission component

The submission component for a programming subject should be capable of managing submissions that consist of many files. The submission component must also be able to detect and prevent late submissions if necessary. Thirdly, the submission component must be secure. Chapter 3 and 4 highlighted the importance of security within a submission system. Integrity, confidentiality, non-repudiation, access control and authentication were found to be especially important. When an assignment is submitted the system must ensure that the person who submits the assignment is who he or she claims to be. This will therefore ensure that students cannot submit assignments for one another. Measures must be in place to ensure no unauthorised entity may gain access to the assignment that is being submitted. This can be done by making use of encryption. The integrity of the submitted assignment must be ensured. This may require tools that students can use to verify that their submission's integrity is intact after submission. Finally a mechanism must exist whereby students can prove that they have submitted their assignments.

Jones and Jamieson (1997) note that students often submitted assignments that were infected with macro viruses. Viruses therefore pose a substantial risk to the users of an assignment management system (Jones & Jamieson, 1997). The probability of students submitting infected assignments is especially high with programming assignments, since students often attach the compiled executable versions of their programming assignments to their assignment submissions. This places the lecturer's machine in danger from being infected with a Trojan or a virus. It is therefore advantageous to provide a means by which submitted assignments can be scanned for the presence of viruses. Such a facility could, however, also be implemented within the storage component of the model.

- **E-mail-based submission**

E-mail has been a popular method of assignment submission, especially because it is easy to attach multiple files to an e-mail. A potential method of implementing an e-mail-based submission system follows:

An assignment would be e-mailed to the submission component's e-mail address, with the subject of the e-mail being the assignment details. This may include the assignment code, class code etc. This information would then be used by the server to ascertain the details of the submission. The content of the e-mail could then be encrypted and signed with PGP or S/MIME (See Chapter 3). This would ensure confidentiality, integrity and authentication. Non-repudiation can additionally be implemented by signing and returning the submitted assignment to the student as proof of submission.

This e-mail-based approach has two main disadvantages. Jones and Jamieson (1997) in particular note that making use of information embedded within the subject line is prone to user error. This can however be overcome, by the system e-mailing the correct subject line to the appropriate students. The students would then only need to reply to that message. Another problem with using this method is that it requires PGP or S/MIME to maintain security. This means that students need the appropriate software and keys installed on the computers from which they want to submit their assignments.

An evaluator could use a similar approach when submitting an evaluated assignment. The main difference would be that he or she would also have to include the details of the student to whom the assignment belongs, in the subject line. Unfortunately this particular approach means that the evaluator would have to submit each student's assignment separately, wasting time and effort.

- **Web-based submission**

A secure web-based submission approach, would require the user to log onto the system by using a User ID and Password combination, thus authenticating the user. Once this is done the user should be able to submit the appropriate assignment. The submission can be done using one of two methods. The most commonly used method entails using a file field in a form. The submission would have to make use of SSL or SHTML (see Chapter 3 sections 3.6.1 and 3.6.2) to ensure the submissions confidentiality and integrity during transit. Due to the nature of the Internet, it is however possible for assignments to be partially submitted. A mechanism must therefore exist to ensure the integrity of the submitted assignment. This can be done by displaying the file size of the submitted assignment to the user, or by allowing him or her to download their submission for verification.

Another approach used for submission is by making use of an Active-X control. The control would allow the user to select the assignment files and handle the transmission of these files on its own. The control could additionally be used to ensure that the submission was accomplished without any problems. The Active-X control can do this by calculating the hash values of the submitted files on the user's machine. (See Chapter 3.) The storage component would then have to calculate the hash values of the submitted assignment and return these values to the Active-X control. The control would compare the values, thereby verifying the integrity of the submission. The Active-X approach is therefore superior to the form-based submission approach. It does however have a major disadvantage. Active-X controls are often considered as posing a security risk, resulting in users disabling them. This would in turn make submission impossible.

Finally, as stated before, the submission component must also ensure non-repudiation. Either of the approaches could do so by generating a custom token for the submission. A token is a string of characters that act as a receipt for a submission. Therefore, if required, the student can prove that he or she has submitted an assignment, by presenting the token that was generated for that submission. (See Chapter 3.)

6.2.4 Storage component

The main and obvious requirement of the storage component, is that it must be able to indefinitely store submitted assignments, and the information surrounding them, such as marks. It is also responsible for keeping track of who the submitted assignments belong to, when they were submitted, and for whom they are intended. The storage component must also maintain the integrity and confidentiality of the assignments stored within it.

- **E-mail-based storage component**

It is unlikely that e-mail on its own would suffice as a storage component. This is because, if important assignment details are stored within the subject line, some mechanism must process that information and act accordingly. On the other hand, the integrity and confidentiality of the e-mails in storage could be easily attained if S/MIME or PGP was used during the submission. Additionally, the e-mail server should be located on a secure machine, which means that even if the e-mails were not encrypted, they would still be relatively secure.

- **Database-based storage component**

An object-orientated or relational-database assignment storage-based system can offer many features. Submitted and evaluated assignments could be stored within the database. This has numerous advantages, such as preventing assignments from getting “lost”, and enabling the association of information with submissions, such as the date of a submission, and the mark that was attained. Databases are also the logical place to store student marks. This concept could however be taken further, by storing the rate of recurrence of commonly occurring mistakes that are made within a submission. This historical information could then, at a later stage, be statistically analysed to aid lecturers in determining potential problem areas. In addition to this, student information, such as contact details and student photos, could be stored within the database.

Until recently, databases generally relied on communications-based security, such as firewalls and network-based security protocols. These solutions do not, however, prevent malicious users, who have access to the computers on which the database resides, from gaining access to the databases contents. This is troublesome as most databases are not encrypted. Oracle 8i is one of the first database's to offer some form of encryption on the data residing within the database (Hammond, 2000; Ihrer, 2000). However, the majority of the databases in use do not offer this level of security. It is therefore important for the database to be installed on an operating system that can provide the needed security.

- **Directory-based storage component**

A directory-based storage approach is very simple, but needs to be implemented on an operating system such as Microsoft Windows 2000, that can provide secure access control to files and directories. Assignments are stored within directories specifically created for them. For example, all of the submissions for a particular subject would go into a directory with the subject code as its name. The directory would, in turn, have subdirectories for each assignment, which in turn would have further subdirectories assigned for each student. For example, a student with the student number s9935672 would be forced to place a particular assignment within the following directory, H:\submissions\Databases101\Assignment2\s9935672. This directory would in turn be only accessible to that particular student and his or her evaluator.

6.2.5 Evaluation component

The evaluation component is used by lecturers to assist with the process of evaluating student assignments. Chapters 4 and 5 stated that programming assignments often consist of multiple files, therefore the evaluation component must be able to handle multiple file assignments. Finally, the evaluation component must be able to annotate these assignments. The annotation of the assignments is essential, as it provides students with feedback to their assignments. This enables students to see where they

made mistakes, compared to only receiving a mark. The end result of the evaluation component should therefore provide students with feedback. The evaluation component should be capable of producing rich feedback. The following sub-sections will describe various methods of annotation, as well as general tools that can be used to assist the evaluator during the evaluation process.

- **Online annotation systems**

As the name implies such systems require the marker to mark the assignment over a network. This approach uses a client-server-based system, requiring the marker to be constantly connected to a server. Such systems are usually implemented with Common Gateway Interface (CGI) scripts that reside on the server, and with Active-X controls or Java applets that reside within the web browser of the evaluator. The evaluator would therefore evaluate the assignments using the web browser residing on his or her machine. It is possible to classify two types of online marking systems by focusing on their implementation.

Some systems use a multi-transaction-based approach. This means that as soon as an evaluator marks or annotates a specific section, this information is sent to the server to reflect the changes made. This implementation has various advantages and drawbacks. Firstly, this approach implies that for every annotation made, the web browser must communicate with the server. This may cause a noticeable delay that could frustrate evaluators (Mason & Woit, 1999). On the other hand, this approach is more resistant to communications failure over the Internet, because the server will always be up to date. The alternative approach is to update the server only once the evaluation has been completed. This approach does not have the transaction delay problem that the multi-transaction approach has, it is however more vulnerable to communications failure over the Internet.

An advantage of using an online annotation approach is that it is streamlined, automating most of the submission process. This approach does however have a drawback, since assignments are evaluated online, the confidentiality and integrity of

the assignments are at risk. For this reason a protocol such as SSL should be used when using the online-based approach.

- **Predefined errors**

Students in a particular subject often make commonly recurring mistakes. It is tiresome for the lecturer to repeat his or her description of such commonly recurring mistakes. It would therefore be advantageous to have a feature whereby lecturers could pre-define commentary for commonly occurring mistakes.

- **Automated program testing and analysis tools**

Zin and Foxley (1997) state that these tools can be divided into two categories: static analysis and dynamic testing. Static analysis-based tools inspect the code's structure and syntax to determine static errors and to produce statistical information about the code. On the other hand, dynamic testing involves executing the code and testing it against test data. Zin and Foxley (1997) go on to state that these tools are especially useful in programming classes, with Joy and Luck (1998) describing how such tools can be integrated within electronic-marksheets to assist with the evaluation process.

Unfortunately these tools are far from perfect, with Zin and Foxley (1997) acknowledging that certain types of programs cannot be suitably analyzed. In particular, they note that it is difficult to generate test data for some types of programs. Due to these, and other problems related to such software, Joy and Luck (1998) note that these tools should merely be used as aids in the evaluation process.

6.2.6 Feedback and retrieval component

The Feedback component is responsible for giving students feedback to their assignments. The component must also ensure that that the feedback is given in such a manner as to ensure the confidentiality and integrity of the feedback.

- **E-mail-based feedback**

E-mail provides an effective means of providing students with feedback to their evaluated assignments. This is because it is easy to embed general commentary and the mark that was attained in the body of the message. The feedback would, however, only be on the assignment as a whole, with the commentary not linked to the specific place where the problem occurs in the programming assignment. Additionally, the evaluated assignment itself can be attached to the e-mail. The e-mail would then be secured with PGP or S/MIME ensuring integrity authentication and confidentiality.

- **Web-based feedback**

An advantage of using a web-based feedback approach is that students only require a web browser to interact with the assignment management system. This approach allows for feedback to be in the form of HTML pages or even allowing the users to download the evaluated assignment files themselves. Security would also be maintained by making use of protocols such as SSL. A web-based approach would however, additionally require the user to log onto the system with a User ID and Password to ensure authentication and access control.

6.3 Summary

The objective of this chapter was to propose a model for the secure management of assignments. This was done by proposing a model that consists of five components, each of which have their own responsibilities. The **management component** is firstly responsible for the creation and administration of classes and users. The management component is additionally used to manage assignments (creating and editing assignments). The **communications component** is used by lecturers to inform students about assignments that need to be submitted, while lecturers also use it to determine whether any assignments have already been submitted. The **submission component** is responsible for the secure routing of assignments to and from students

and lecturers, while the **storage component** is responsible for the storage of the submitted and evaluated assignments and marks. The **evaluation component** is used by the lecturer for the evaluation of the assignments, and finally, the **feedback component** is responsible for providing students with feedback to their assignments.

These components can be implemented in several ways, while still providing the needed features. Chapter 7 will discuss a specific implementation of the model proposed in this chapter, so as to prove the viability of it. The implementation will be in the form of a web-based assignment management prototype, specifically customised for use within the programming environment.

The implementation of a web-based assignment management model

7.1 Introduction

Chapter 6 proposed a Programming Assignment Management Model (PAMM), that could be used to support electronic-based assignment submission and evaluation. The model was tested by building a prototype named Assignment Manager 2000. From a user's point of view, Assignment Manager 2000 is comprised of three components, each for a particular user type, namely an administration, lecturer and student component. This chapter will primarily focus on the lecturer and student components, since the administrative component is mainly responsible for the administration of users and classes. More detail is available in the user manual in Appendix B.

This chapter describes how Assignment Manager 2000's components were implemented. The implementation that is presented within this chapter was specifically designed to meet the needs of Visual Basic programming classes within the Port Elizabeth Technikon.

7.2 The physical components of Assignment Manager 2000

The following section will describe the physical components of the web-based assignment management system, as it is currently implemented within Assignment Manager 2000.

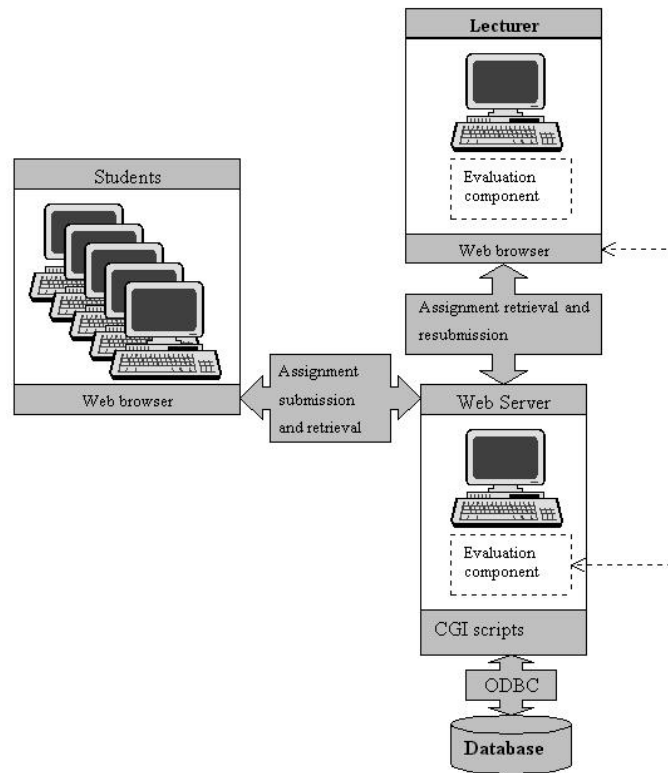


Figure 7.1 A web-based assignment management system.

Figure 7.1 illustrates a possible web-based implementation of the model (PAMM) described in the previous chapter. Students and lecturers use a web-browser to submit and retrieve assignments. The lecturers use the evaluation component to evaluate the assignment. Figure 7.1 displays the evaluation component in a dashed-line block that can reside on either the web-server or on the lecturer's machine as a standalone application. If the evaluation component were to be implemented on the web-server, the lecturers would also use the web browser to access it. The evaluation component within Assignment Manager 2000 is, however implemented as a separate application residing on the evaluator's computer. This is primarily due to the requirement of lecturers being able to evaluate assignments offline. The following sections will describe the physical components of Assignment Manager 2000 in greater detail.

7.2.1 Web Browser

Students and lecturers use a web-browser to interact with the system. This communications channel is secured by means of SSL (Secure Sockets Layer), providing data confidentiality and integrity. The web browser needs to be capable of correctly displaying DHTML (Dynamic HTML), as this is required for the submission of, and online viewing of, evaluated assignments to work.

7.2.2 Web Server

The web server handles all of the web browser's transactions. This is done with the help of CGI scripts, which were written in a server side scripting language PHP, that resides on the web server. PHP was chosen because of its ease of use and portability. It can run under Windows and Unix environments making it extremely flexible. The web server that was used was Apache-SSL, a derivative of Apache and OpenSSL, which implements Secure Sockets Layer (Apache, 2000). This web server was chosen because it is portable and secure and because Apache has been the most popular web server on the Internet since April of 1996, according to the Netcraft Web Server Survey which claimed that Apache and its derivatives are used in over 62% of the web sites found on the Internet (Netcraft, 2000).

7.2.3 Database

A relational database is used to store detailed information about the students, practical assignments as well as any information pertaining to them. The web server uses ODBC to communicate with the database, which enables the database to be implemented on any ODBC-enabled database. (Figure 7.2 displays the tables and field found within Assignment Manager 2000's database.) The database was implemented using Microsoft Access 97 for the prototype.

An advantage of storing all of the assignment details in a relational database is that it allows the lecturer to have instant access to all of the assignments of any student. The database also stores information pertaining to the types of mistakes that students made in their assignments. The lecturer can therefore use this information during the evaluation process. The students also benefit because the database acts as a repository for them as well, making it impossible for them to lose old assignments.

A disadvantage of using this approach, is that it firstly requires a machine for the database to run on and secondly that the database will grow fairly large. This potential problem was overcome by storing pointers to the files in the database. The assignment files are therefore stored within a secure directory that cannot be directly accessed by students or lecturers. (The “File” field in the “File” table in Figure 7.2 is used to store a pointer to each of the assignment files.)

PHP scripts communicate with the database via ODBC. A problem with this is that ODBC offers no protection from unauthorised entities accessing the data that is transmitted over the communications medium. This possible security breach can consequently be avoided by moving the database onto the same machine as the web server. This will result in the information that is relayed from the database to the PHP scripts not needing to travel over the entire network or Internet. Moving the database onto a secure web server also has another advantage. Databases in general don’t have any embedded cryptographic security (See 6.2.4 in Chapter 6). That is to say, their contents are not encrypted. Any person with physical access to the database files can therefore get to the data stored within them. Consequently, since a great deal of effort is generally undertaken to secure web-servers, the database should be relatively secure.

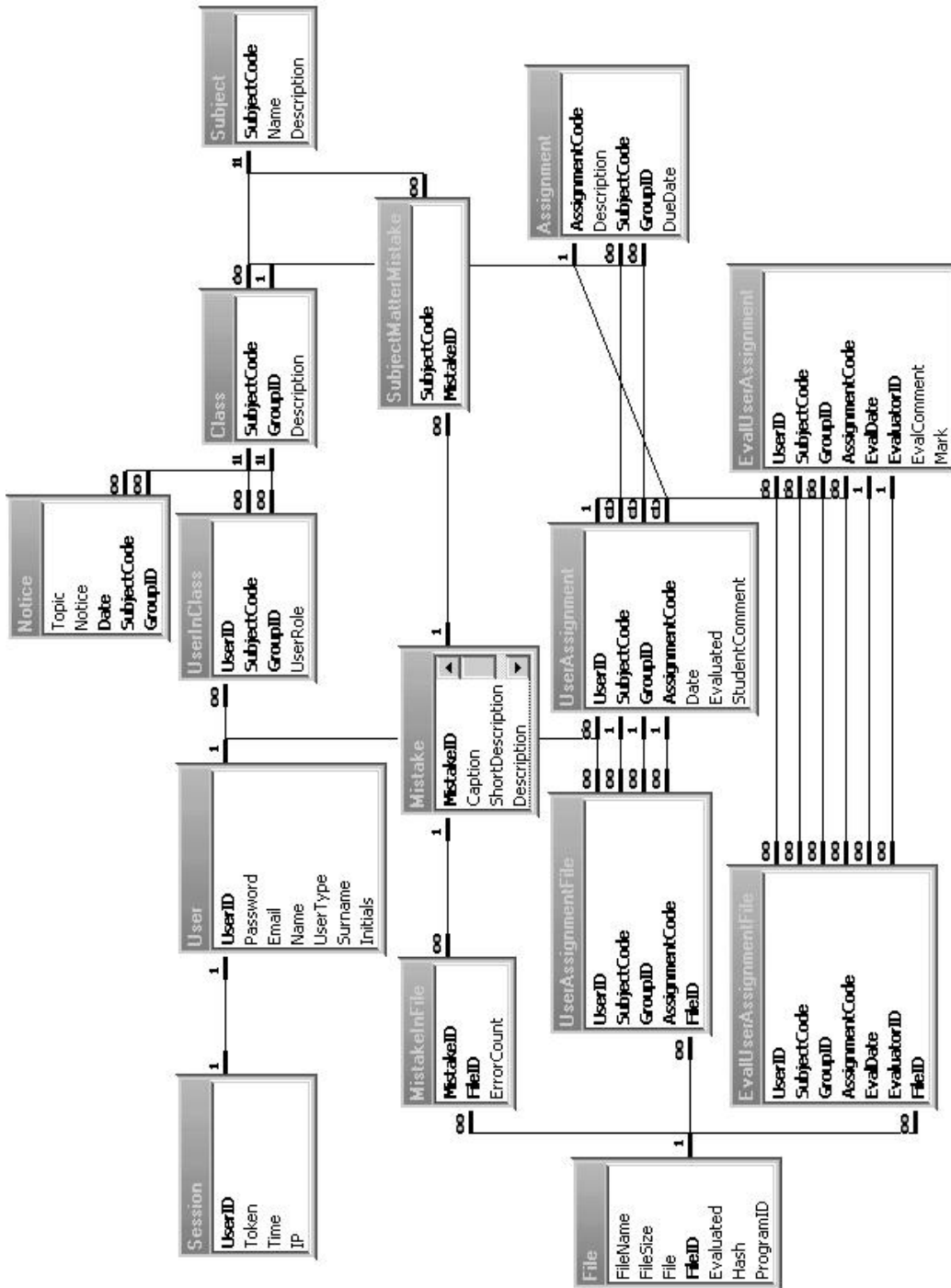


Figure 7.2 An ERD equivalency diagram of the database.

This section will briefly describe the role of the major tables in Figure 7.2:

Each user of the system is stored within the **User** table, which contains the users' personal details. The users can be members of a specific **Class**, with the **UserRole** specifying their role in that class, for example a user could act as a lecturer or a student. The **Class** is a specific group of students studying in a specific subject. The **Notice** table is used to store the noticeboard entries for all of the classes. When a lecturer creates an assignment the assignment details are stored in the **Assignment** table that is related to the appropriate **Class**. The submission of an assignment will result in the assignments submission details, such as the submission date being stored in the **UserAssignment** table. Any files attached to the submission are stored in the **File** table. The file table also stores detailed information about each file, such as its size and hash value. Evaluated assignment details are stored in the **EvalUserAssignment** table. This table includes entries for the date when the assignment was evaluated, the mark attained etc. The evaluated assignment files are also stored within the **File** table. Predefined error types are stored within the **Mistake** table. The **MistakeInFile** table stores how often a particular predefined mistake occurred in a particular evaluated file.

7.3 Management features

7.3.1 Access to the system

Assignment Manger 2000 contains class and user management tools that enable the creation of users and classes (see Appendix B). This includes a registration script which can automatically add students and classes to the system.

Access to the web component is restricted via a login screen. The login screen prompts the user to enter a user ID, password and user type. The web component supports three different types of users: *administrators*, *lecturers* and *students*. A

dropdown list of user types is used to select the role that the user will perform for the duration of the session. The different user types are used to maintain access control to the web component. This enables students and lecturers to only access resources that their user types are authorized to access. Another advantage of using such an approach is that it enables a user to use one user ID and password combination to log in as a student, lecturer or administrator.

When a user logs onto the system a unique token that is valid for a period of one hour is generated. This is stored together with the user's User-ID, IP address and time within the database. (This is shown in the session table in Figure 7.2.) When the user runs any script, the web browser supplies the script with the user's User-ID and token. This is compared to the stored values in the database, if the values match up, and the time from the last transaction is less than one hour, the user is allowed to execute the script. The time value in the database is also updated. The inclusion of the user's IP address ensures that even if other users manage to get the user's token (which should not happen as the communications medium is encrypted with SSL), it will be useless to them because they would have a different IP address. This approach does, however, have a disadvantage. If a user does not log out of the system and tries to log in again from a different IP address within an hour of his or her last transaction, he, or she, will not be allowed back into the system until the previous token has expired.

7.3.2 Assignment creation

A lecturer can add an assignment by clicking on the *create assignment* button. He or she is then presented with a list of all the subjects that have been assigned to them. (See Figure 7.3.) The lecturer must select the classes for whom the assignment is intended, and then fill in the assignment number, description and due date of the assignment. *Note: If multiple classes are selected when creating an assignment, a unique assignment is automatically created for each class. This allows lecturers to change due dates on a class basis if needed at a later stage.*

The system can accommodate multiple lecturers for a particular subject code, by assigning students for a particular subject to a particular class group. This for example, enables lecturers to give their part-time students different assignments from full-time students. Once the assignment has been created, an entry is automatically created on the student's calendar.

Subject	Group	Class description
<input checked="" type="checkbox"/> VB101	1	Visual Basic 1 full time class

Due Date

23 8 2000

Assignment Number

1a

Assignment Description

Write a program that calculates the sum of the odd numbers from 0 to 100.

Submit assignment

Figure 7.3 Assignment creation form.

7.4 Student features

7.4.1 Assignment submission

Once a student has successfully logged onto the system, he or she is presented with the screen depicted in Figure 7.4. The student main screen consists of two calendars, a notice board and a list of recently evaluated assignments. The calendars display the

current and forthcoming month, and indicate if assignments are due within this time period. This is achieved by colouring days on which assignments are due in different colours. In addition to this, if a student moves the mouse cursor over a day when an assignment is due, a window appears with the details pertaining to all of the assignments that are due on that day.

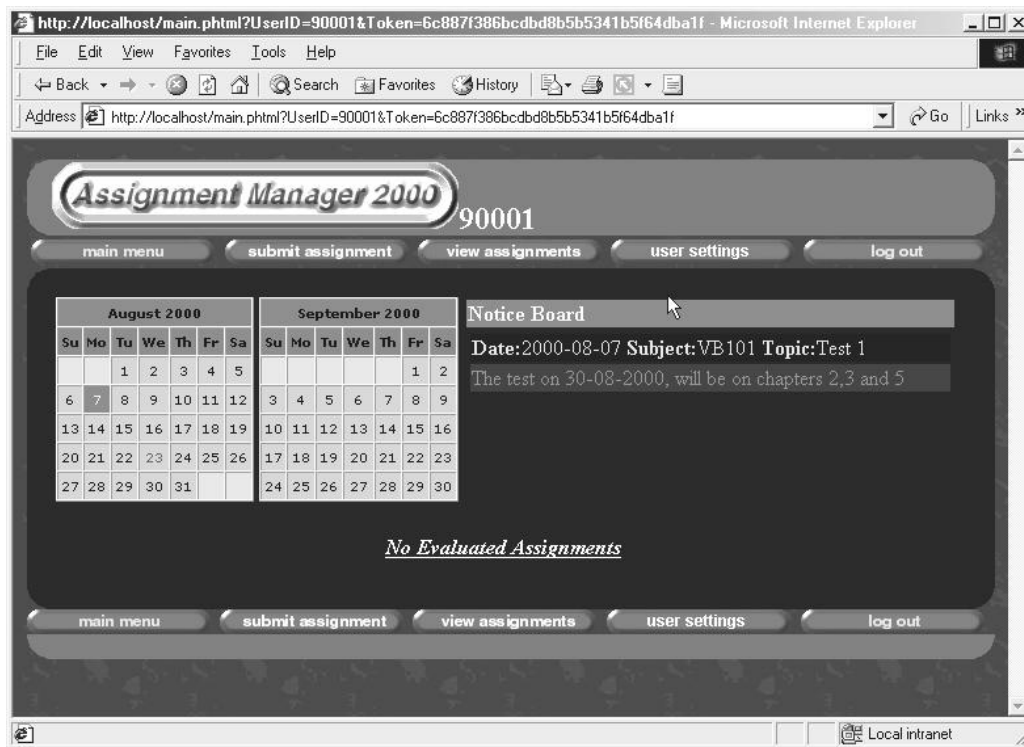


Figure 7.4 The student main menu.

The notice board displays all of the students’ respective notices for their different subjects. The student’s evaluated assignments are displayed below the calendars. If a student wishes to submit an assignment he or she must click on the submit assignment button (Figure 7.4), which will bring up the page that is presented in Figure 7.5.

The student submission screen (Figure 7.5) requires the student to select which assignment he or she is submitting. It also allows the student to attach a comment to the submission. It only allows students to submit assignments before the predefined due-date; it does, however, allow students to resubmit their assignments before the due-date.

A particular problem that had to be overcome when designing the submission component was that of supporting multi-file uploads via forms. This is because web browsers such as Microsoft Internet Explorer and Netscape Navigator only allow users to select single files for an HTML file upload form field. Web-based e-mail systems such as Hotmail (Microsoft Corporation, 2000), have “*overcome*” this limitation by forcing the user to upload each e-mail attachment separately via a separate form. Once all of the files have been attached (uploaded), the user can “send” the e-mail. This approach is, however, time consuming as it forces the user to wait for each attachment to be uploaded until they can click on the send email button. Other Internet sites such as PhotoPoint (PhotoPoint Corporation, 2000) have overcome this problem by asking the user how many files need to be attached and then generating a form with that number of file upload fields. This approach however has two disadvantages. Firstly, it requires the user to select the number of files that they want to upload, and secondly, each file upload field takes up screen space. This is particularly a problem for programming assignments as an assignment normally consists of many files.

Various web sites were investigated to determine whether their approaches to uploading files could be applied within the current system. Their approaches were however, found to be unsuitable. A custom approach specifically optimised for use with programming assignments was therefore developed.

The form in Figure 7.5, allows students to attach multiple files to a submission, and moreover allows them to group files together into units. The file grouping feature was deemed important as student assignments often consist of many programs, each consisting of several files. It is thus important for the lecturer to determine which files belong to which program.

The students would use the following process to submit an assignment: Firstly, the student must select the appropriate assignment that he or she wants to submit. This is done by checking the appropriate radio box that is situated on the top of the form. The student can then enter a brief description of the submission in the comment box.

Submit evaluated student assignment				
Subject	Group	Assignment	Description	Due Date
VB101	1	1a	Write a program that calculates the su...	2000-08-23

Comments

Hope you are happy with this

Attachment:
Program:

File attachments
 D:\TEMP\PicIT\frmMain.frm
 D:\TEMP\PicIT\Module1.bas

Instructions: If applicable, select the appropriate Program from the **Program select** box. Then click the **Browse** button to locate the file you need, and select it. The file path will appear in the attachment field. Next, click **Attach** to attach the selected file to your message. When you send your message, the attached file is automatically enclosed

Figure 7.5 Student assignment submission form.

Files are attached by clicking the browse button and selecting the appropriate assignment file. Once this is done, the student must decide to which group or unit the files belong to. This is done to selecting a “Program” from the program box. The student must then click on the attach button. This will result in the selected file being added to the file attachments list box. After the student has repeated this process for all of the required files, he or she can click on the submit assignment button and the files will be submitted. This method allows the students to submit programming assignments that consist of many programs, which in turn can consist of multiple files. Lecturers are therefore able to discern which files belong to which programs, even if they have the same filename.

Once the assignment has been submitted a token is generated and sent to the student’s e-mail address. The token is a unique text string that is generated by a PHP script residing on the web server. The role of the token is twofold; firstly it acts like a receipt proving to the user and the web component, that he or she has submitted their assignment. Secondly, the token acts like a digital fingerprint (MD5) of the submitted

assignment. In the case of a system failure, the token can be used to prove that the assignment was submitted and the hash values (MD5) can, if need be, be compared to the resubmitted assignments hash value to ensure that the resubmitted assignment is indeed an unmodified version. The submitted assignment files are stored on a secure server, with the **file** field in **File** table of the database storing a pointer to the relevant files.

A problem with form-based file uploads is that errors can occur during the upload that the user is unaware of. This may result in incomplete uploads or zero byte files. The students are, however, presented with tools that enable them to view the file size of the uploaded files and even to download their submissions.

Chapter 6 notes that viruses could pose a substantial risk to a submission system and recommends that steps should be taken to ensure that submitted assignments are virus-free. The web-server is currently installed on a machine that has anti-virus software resident in memory. This should take care of such submissions. Despite this, a facility was created within the submission component to facilitate the detection of viruses within submitted assignments. This was done by creating a script that is automatically invoked whenever a file is uploaded. This script can be configured to use almost any commercial anti-virus software that can be invoked from the command line.

7.4.2 Viewing evaluated assignments

Once the lecturer has submitted a student's evaluated assignment, the students can view the marks that they received, together with the current class average, minimum and maximum marks attained. The students can therefore judge for themselves how well they performed.

A student's evaluated assignments can be accessed using two methods, via the main menu or via the *view assignments* form. If the student clicks on an evaluated assignment, the assignment's details are displayed together with the evaluated

assignment files. If the evaluation application was used to evaluate the assignment, HTML and Rich Text versions of the evaluated assignment are also displayed on the form. Commented regions within the HTML version, are underlined and coloured differently from the rest of the text (Figure 7.6). If the user moves the mouse cursor over such commented regions, detailed commentary may be displayed in the area below the main body text. The students can also easily print out the commented Rich Text versions of the evaluated assignments.

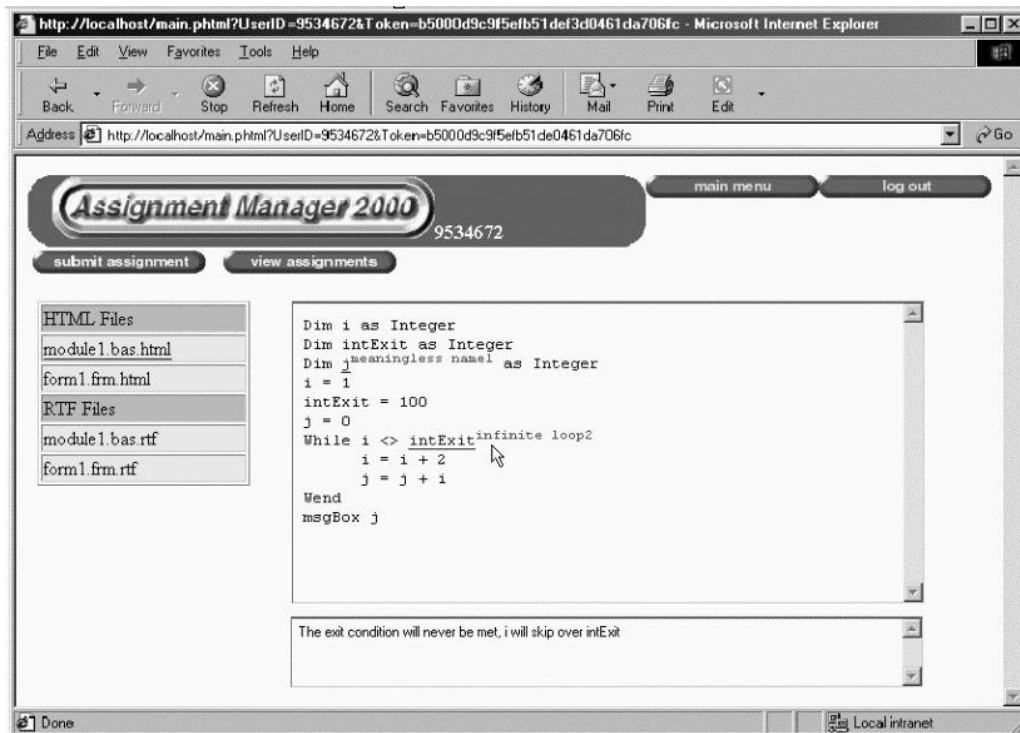


Figure 7.6 Displays the student's view of an evaluated assignment.

7.5 Lecturer features

7.5.1 Lecturer retrieval of submitted assignment

Assignments that were submitted by students can be retrieved using one of two methods. The preferred, and easiest method of retrieving assignments is by using the

main menu page (Figure 7.7). The table on the far right hand side lists the number of submitted assignments for a particular class and assignment. Each table entry has a *view* and *export* button. The export button combines all of the unmarked assignments in a class into a single file. This file is then uploaded by the lecturer, who in turn can decide where to place it on his or her computer.



Figure 7.7 A section of the lecturer's main menu.

The evaluation application (Figure 7.10) is used to extract the combined assignments on the lecturer's computer. The view button can be used to view the details of a particular class, such as the class details displayed in Figure 7.8 that contains three submitted assignments. Each entry in the table represents a user's assignment, offering the lecturer the ability to export and view individual assignments. The view button for each table entry is used to view individual assignment details. Therefore, if the lecturer were to click on a view button in Figure 7.8, Figure 7.9, would be displayed.

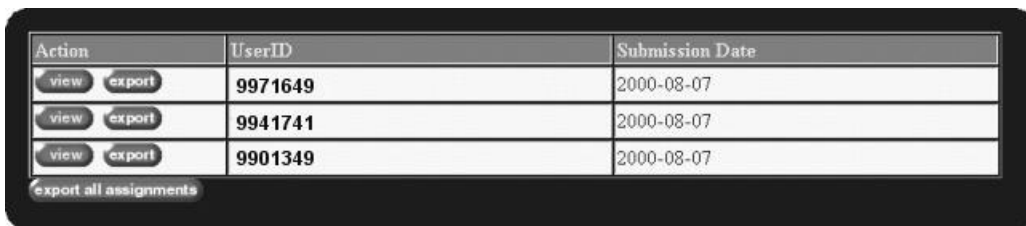


Figure 7.8 Displays the submitted assignments for a particular class and assignment.

Figure 7.9 displays the details of a particular student's assignment submission. The white text area contains the student's comment that was made during submission. The submitted files are listed in a table below this area. The lecturer can download any particular file from the assignment by clicking on the filename, or download them all

as a single file by clicking on the export button; however, using the export function requires the use of the evaluation application for file extraction.

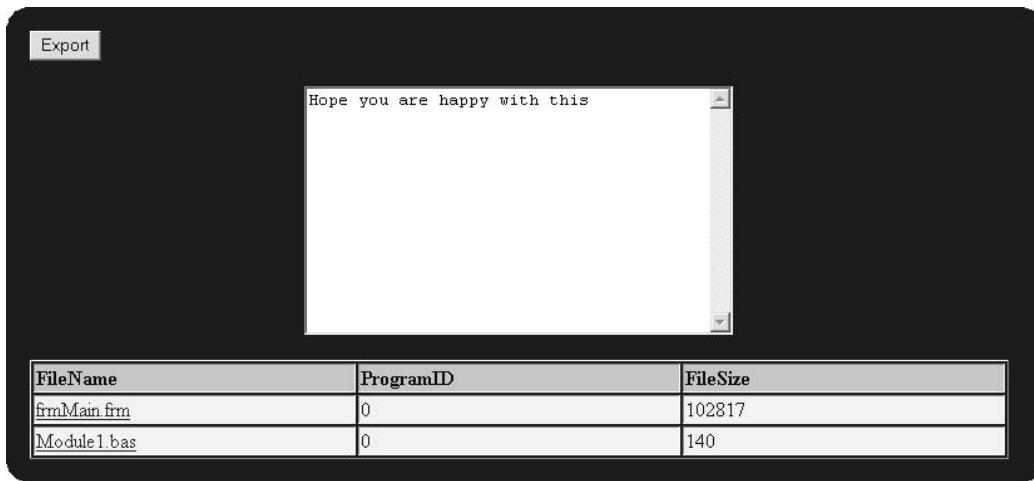


Figure 7.9 Displays the details of a submitted assignment.

Notes about the use of the export functions

As stated above the export function enables lecturers to download groups of assignments as a single file. The evaluation application processes this file, extracting the original files, and placing them within folders, much like a compression program. A problem with this, is that while a third party program such as Word, could be used to load the extracted files, the program would still require the presence of the evaluation application for the initial extraction. A separate utility that only handles the extraction of exported files was therefore developed. This means that lecturers who want to use the export function, but who don't want to use the evaluation application, can use this utility for the extraction of the files.

One may ask why a standard file format such as .zip was not used. A compression format such as zip was not used because the compression process would slow down the server, and because the compression algorithm used within zip files (LZ algorithm) is patented and requires payment of royalties. Furthermore, the exported file contains embedded information that could not be easily added via a normal compression application.

Proponents of web-based standards may ask why a standard file format such as Extensible Markup Language (XML), was not used. XML possesses some major limitations, which prevent it from being extensively applied within the proposed model. The main problem being that it was not designed to enable the embedding of binary data directly within a XML file.

XML places constraints on the types of characters, as well as their syntax within an XML document. This means that before any binary data can be embedded into the document, it needs to be encoded into a valid character format (Rein, 1998). In addition to this special characters need to be represented differently. For example "<" is represented as ("<"). This can be a time consuming operation and also requires the client to do the reverse operation to restore the data to its original format. Another problem with this approach is that changing the data to an ASCII format increases the size of the content, resulting in longer file download times and greater storage requirements. XML does, however, contain two tags that tell the parser to ignore the data contained within them. Namely the COMMENT and CDATA tags (North & Hermans, 1999). A problem with using these tags, is that they require that their closing tags are not included within the content, meaning that the content still needs to be modified, consuming time.

Another problem with using XML as the file format of choice is that certain XML parsers will not work well for this application. Two types of XML parses exist namely tree-based and event-based parsers (North & Hermans, 1999). Tree-based parsers store the content of a XML document in a tree-like structure. Event-based parses generate an event when a tag is found. A problem with using a tree-based parser is that it consumes much more memory than an event-based parser (North & Hermans, 1999). This is a particularly nasty problem as the parser that Microsoft supplies with their development environments is a tree-based parser. The memory requirements of a third party application that uses this type of parser would therefore easily outgrow the available memory. On the other hand, event-based parsers do not have this problem. If XML were to be used within the system, the evaluation application would thus be required to use an event-based parser.

Based on the issues listed above, specifically the lack of binary data storage within XML, it was decided impractical to use XML as the export and import file format of choice within the prototype.

7.5.2 The Evaluation of student assignments

The evaluation application (Figure 7.10) is designed to be flexible in its use and specifically caters to the needs of Visual Basic programming lecturers. The [1], [2], [3] and [4] that are shown in the image were inserted to assist in the explanation of the application. The application operates independently from the web component. This means that other applications could be used for evaluation purposes.

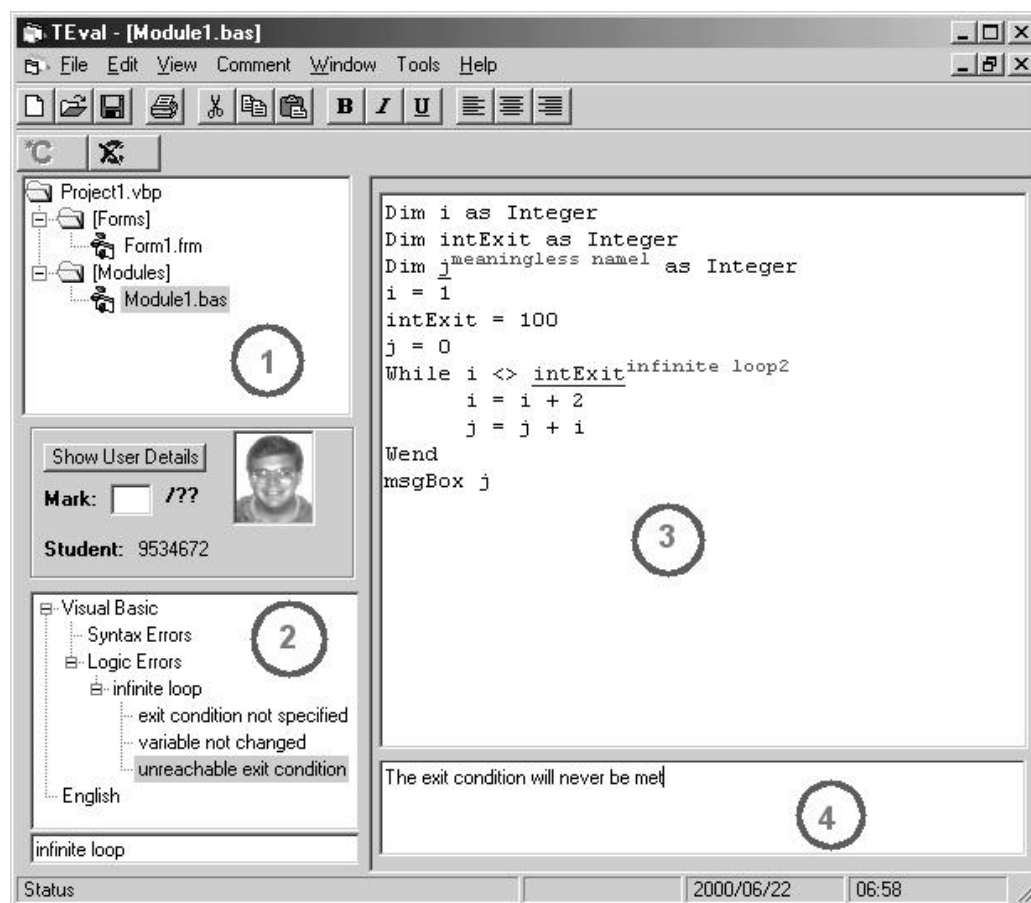


Figure 7.10 The evaluation application.

The application supports any rich text or text-based files. In addition to this it has native support for Visual Basic files (.vbp, .cls, .bas, .frm) and can simultaneously cope with multiple visual basic projects per assignment. If the lecturer opens up a visual basic project file, all of the related files are automatically loaded, and can be accessed by clicking on the appropriate file listed within the treeview [1].

7.5.3 Annotation

A lecturer can comment a particular passage of text, by selecting the text passage in question and either left click on the “add comment” button, or on a predefined comment in the error-tree (Figure 7.10 [2]). If the “add comment” button was used, a form will appear asking for a brief description of the error. This text (together with a comment counter) will be displayed in superscripts adjacent to the selected text. The originally selected text is also re-coloured and underlined so as to make it easy to see (for example, for i = 1 to 10 step -1^{infinite loop1}). Once this is done, the focus will be automatically shifted to the comment text box [4]. A detailed comment or description can be made in this box.

On the other hand, if the user clicks on a predefined error, the error description and detailed commentary is automatically filled in. This is described in more detail in the next subsection.

7.5.4 Error Types

Chapter 6 stated that students often make commonly recurring mistakes. For example, many students may make mistakes that cause infinite loops. It is undesirable for the lecturer to repeat his or her description of such commonly recurring mistakes, therefore a feature was added to make it possible to add pre-defined commentary for commonly occurring errors. The evaluation application supports a customisable hierarchy of error types. (See [2] in Figure 7.10) This enables lecturers to pre-define and categorise commonly occurring errors.

A pre-defined error has two fields associated with it. The first field gives a brief description of the error type (superscript) and is used within the body of the evaluated text (in the aforementioned example this would be “infinite loop”). The second field contains a detailed description of the error.

Predefined-errors can be easily added, edited or removed from the error tree, by simply right clicking on the tree ([2] in Figure 7.10), if the user right clicks on a specific error type, he or she is given the option to edit the error type or create a subcategory of that error type. This enables a lecturer to add or modify error types as he or she is marking.

7.5.5 Historical assignment information

As stated in Chapter 1, programming classes are becoming larger placing more strain on lecturers. This has resulted in lecturers been unable to give students individual attention, and by so doing, get to know each student’s particular problems. The evaluation application attempts to alleviate some of these issues by providing the lecturer with as much student information as possible. When an assignment is loaded in the evaluation application, a photo of the student is presented to the lecturer. This helps to the lecturer to put a face to a name or number, making the evaluation process more personalised.

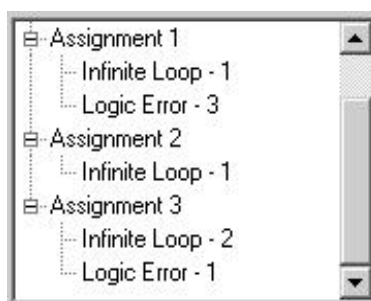


Figure 7.11 Error history tree.

Another tool that was added can be accessed by clicking on the “Show user details” button. Clicking on this button results in the treeview [2] being populated with historical assignment details of previous assignments (Figure 7.11). These details show how often various pre-defined errors occurred in the previous assignments. For example Figure 7.11 shows that assignments 1 and 2 contained an infinite loop, while assignment 3 contained 2 infinite loops. If a student consistently has problems with infinite loops, for example, the lecturer should be able to identify this from the historical details, and address the problem appropriately.

7.5.6 Packaging evaluated assignments

If the web component is used in conjunction with the evaluation application, the lecturer must use the export function to save the evaluated assignment in a format that is readable by the web component. An exported assignment is saved as a single file, which not only contains the evaluated assignment, but also HTML and rich text versions of it. The evaluation application also provides evaluators with a tool to combine all the evaluated assignments into a single file.

7.5.7 Evaluation limitations

As mentioned in Chapter 1, evaluators would often draw lines and scribble over the paper. The evaluation application presented is however incapable of performing any drawing-based functions. This was done for two reasons. Firstly it would have taken a great deal of time and effort to implement such a solution from scratch.

The second and more important reason is that such an approach would require the students to have access to specialised software to view the evaluated assignments. This is not the case with the current method of evaluation, as the students can currently view the evaluated assignments within a normal web-browser.

7.5.8 Lecturer evaluated assignment submission

Once a lecturer has evaluated an assignment, he or she must submit the evaluated assignment. Two forms of assignment submission exist, namely manual submission and automated submission. Automated submission can be used to submit individual or groups of assignments that were evaluated in the evaluation application. The lecturer simply has to select the file(s) with the feedback within the evaluation application, and add them to the form. The web component will automatically determine the assignment details (student ids, marks, etc.) from the embedded information contained within the file(s). In addition to this, the system keeps a record of what types of predefined errors occurred in each evaluated file. (This information is stored in the “Mistake” table in Figure 7.2)

The manual submission should be used when the evaluation application is not used to evaluate the assignment. Consequently certain details about the assignment, such as the errors made, cannot be determined. The evaluator must therefore manually select the user and class to which the assignment belongs to, together with the mark attained. The assignment files must also be attached to the form.

7.6 Summary

This chapter described how a secure web-based assignment management system was implemented. The implementation allows large groups of users to submit and retrieve programming assignments. This is done while maintaining a large degree of security, providing assignment non-repudiation (token), data confidentiality and integrity (SSL), access control and authentication (user-id password and token). In addition to this, the system allows for multi-file assignments and stores historical assignment information within a database. Chapter 8 will conclude this dissertation by evaluating this implementation of the model and discussing further areas of research and development.

Evaluation & Conclusion

8.1 Introduction

The first objective of this dissertation was to determine how the flow of electronic assignments between students and the lecturers could be automated in a secure manner. Furthermore, the study tried to determine how intuitive feedback and assessment could be attained within a virtual learning environment. Chapter 5 determined that currently available commercial solutions could not sufficiently meet the needs of a programming environment. Chapter 6, therefore proposed a model for managing the process of electronic assignment submission and evaluation, which, if implemented correctly, could attain the objectives set out in the first chapter.

The model was tested by developing a prototype that was discussed in Chapter 7. The prototype was specifically customised for use in programming classes and consisted of two components. The first component dealt with the management issues of electronic assignment, creation, storage and retrieval, while the second component dealt with the process of electronic evaluation. This chapter will attempt to do a preliminary evaluation of the prototype to determine how effective it was at attaining the goals set out in Chapter 1. The chapter will then use the information gained from the evaluation and discuss various areas of improvement and research.

8.2 Method of the evaluation

The prototype (Assignment Manager 2000) was used within a second year Visual Basic programming class that consisted of thirty-seven students. Twenty-seven of the students, partook in the evaluation of the prototype by means of a questionnaire, and discussions were held with the lecturer. It was only used for one-cycle of assignment submission, evaluation and feedback. Some small bugs were found, but these were quickly resolved. These factors therefore influenced the outcome of the preliminary investigation.

The students were given three assignments, all due on the same day. The lecturer used Assignment Manager's export function to retrieve all the submitted assignments, which were then evaluated within the evaluation application. The evaluated assignments were then submitted to Assignment Manager 2000. Once the assignments were submitted the students were able to view the evaluated assignments online, or download them for printing.

8.3 Results of the questionnaire

As stated before, the students were given a questionnaire to complete. This can be found in Appendix A. The questionnaire consisted of seven multiple-choice questions, and three long questions. The following section will discuss the results of the questions contained within the questionnaire.

8.3.1 "Did you have any difficulties logging onto the system?"

This question was deemed necessary because it was reasoned that if a student had difficulties logging onto the server, it would influence the rest of the questionnaire. Figure 8.1 shows that 51.85% of the students had difficulties logging onto

Assignment Manager 2000. A number of problems caused students to have difficulties logging onto the system. Firstly, the machine on which Assignment Manager 2000 was installed seemed to be randomly invisible to the network. Secondly, users had problems logging onto the server from outside the Technikon. Troubleshooting with the Computer Services departments did not resolve this problem, and thus outside access to the system was not available.

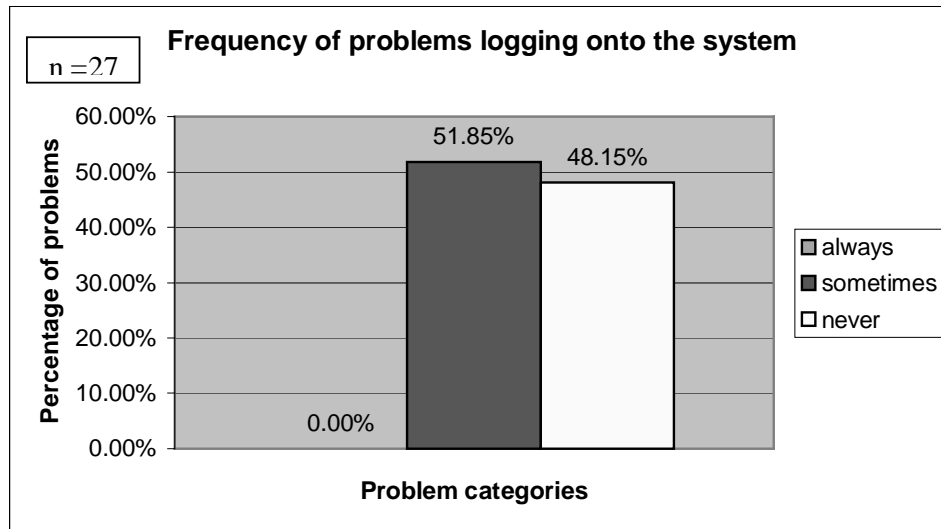


Figure 8.1 Displays how often students had problems logging onto the system.

Thirdly, an additional reason why students had problems logging onto Assignment Manager 2000 was because they did not log out of the system in the first place. This problem arose when they would attempt to log onto the server within an hour of last accessing the server from a different machine. (Refer to Chapter 7). Finally, a power failure the day before an assignment was due, resulted in students being unable to submit their assignments. The lecturer overcame this problem by changing the assignment's due-date.

8.3.2 "How easy or difficult was it to learn how to use the system?"

This graph (Figure 8.2) shows that some 66.7% of the students found the system to be easy (33.33%) or very easy (33.33%) to use, while only 3.7% found it difficult to use.

The remainder of the users (29.64%) found the system's ease of use to be average.

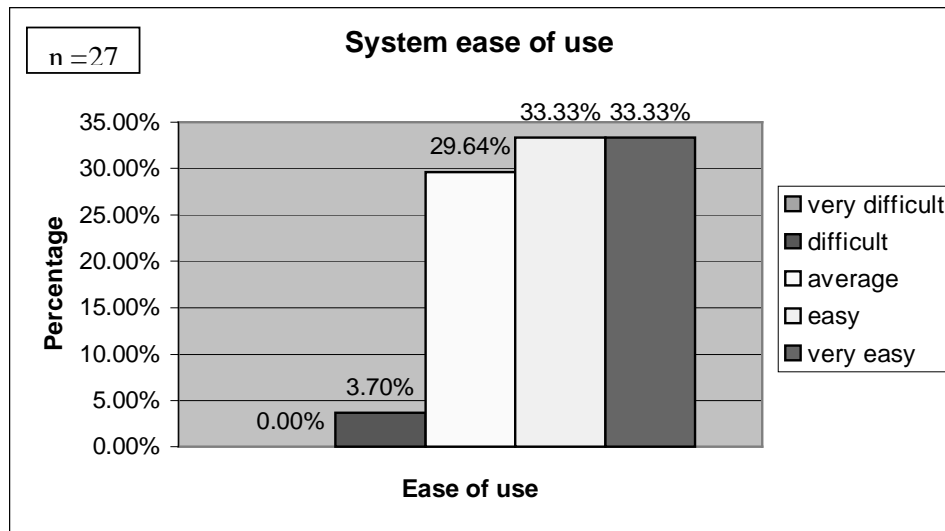


Figure 8.2 System ease of use.

8.3.3 "How would you compare electronic to paper-based assignment submission"

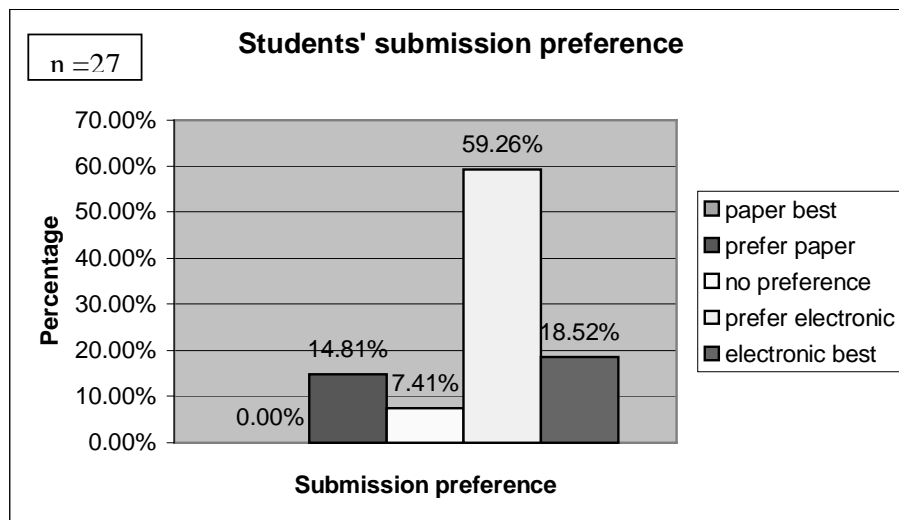


Figure 8.3 Displays the students' submission preference.

This graph (Figure 8.3) shows that some 77.8% of the students were more inclined towards electronic submission with 59.26% preferring it and 18.52% finding it to be

the best submission method, while only 14.81% of the students preferred paper-based submission. The remainder of the users (7.41%) had no preference.

8.3.4 “How do you feel about receiving evaluated assignments electronically compared to paper?”

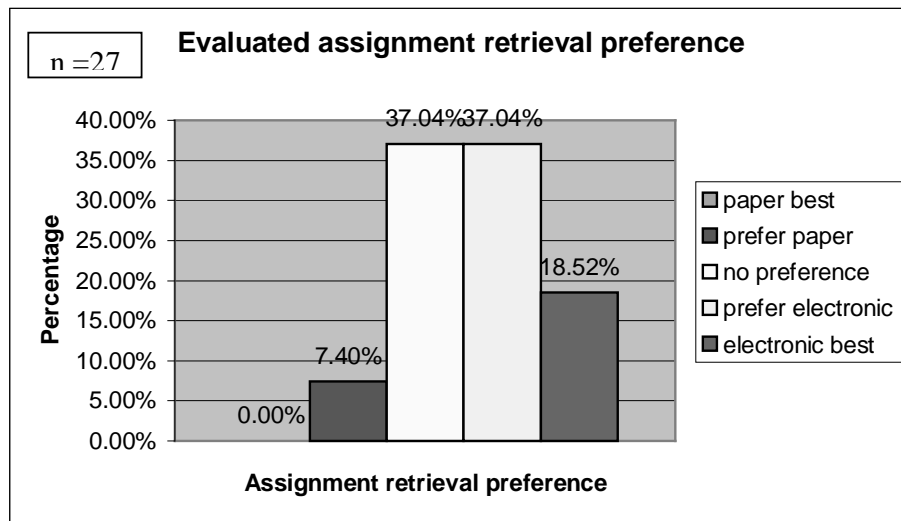


Figure 8.4 Evaluated assignment retrieval preference.

This graph (Figure 8.4) shows that some 55.56% of the students prefer (37.04% + 18.52%) to receive their evaluated assignment electronically, while 7.4% prefer paper-based returns. When one compares this graph to the previous graph it becomes evident that the students don't feel as positive towards electronic returns as they do with submission. It is therefore beneficial to have the ability to print out the submitted assignments for those who prefer to do so. The remainder of the users (37.04%) had no preference with respect to electronic versus paper-based assignment retrieval.

The following questions are open-ended questions. The numbers in the brackets represent the number of students who raised the same issues.

8.3.5 “What did you like the most about the system?”

A large number of respondents (11) commented positively on Assignment Manager 2000’s ease of use and performance (3). Many students additionally found the method of submission to be easy and time saving (9), with several students commenting on how they would not have to wait for the lecturer to submit their assignments (5). Finally, some of the students (2) favoured the fact that their marks are more private compared to the traditional method.

8.3.6 “What did you dislike the most about the system?”

The most common complaint about Assignment manager 2000 was that the students had problems logging onto the system (6). As stated before this was mainly due to network related issues and a power failure. The second complain that the students had with the system was the tedious method of attaching files to a submission (3). In particular, students disliked the fact that they had to click the browse button, and browse for every file to be attached individually.

8.3.7 “What changes should be made to improve the system?”

The major improvement that students wanted was the ability to submit entire directories in a single action (3). Students also suggested the addition of online help, and links to Visual Basic sites (2). Finally it was also suggested that an e-mail style chat facility be integrated into the system (1).

8.3.8 “Do you see a potential use for electronic-based assignment submission?”

Figure 8.5 illustrates that the overwhelming majority (92.6%) of the students felt that there was great potential for the use of electronic-based assignment submission.

Only 7.4% felt that paper-based submission was the way to go.

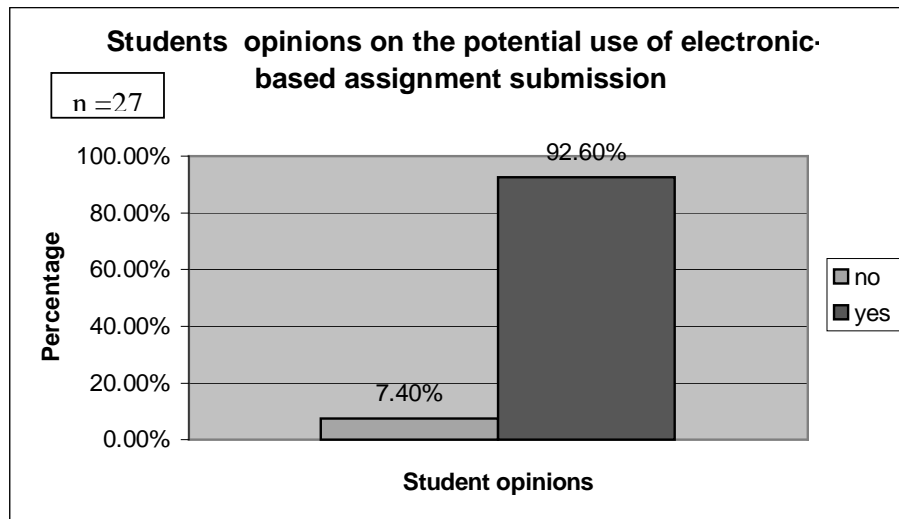


Figure 8.5 Displays the students opinions, on the potential use of electronic-based submission of assignments

8.4 Evaluation by the lecturer

The lecturer was generally satisfied with the way Assignment Manager 2000 worked. The main problems or limitations that the lecturer had with the system were related to the evaluation application. In particular the lecturer stated that he found a need for a facility whereby the lecturer can determine which assignments have already been evaluated, via a menu. With the current implementation, the evaluator has to browse the directory structure to determine this.

As stated in Chapter 7, students can attach any types of files to an assignment, the evaluation application however only supports certain file types. Therefore when an assignment is loaded within the evaluation application, it will only load those supported files (for example Visual Basic and Rich Text files) ignoring all of the unknown file types. The lecturer could still access the unsupported files, this however entailed the lecturer traversing the directory structure to find the appropriate file. The

lecturer found this to be cumbersome and suggested that the unknown files also be displayed within the file-treeview (Figure 7.9). Furthermore, he suggested that if he were to right click on one of these files, the evaluation application would automatically load the file within the application that Microsoft Windows associates with that particular file type. For example, if the lecturer clicks on a Visual Basic file, the evaluation application would load the file within Visual Basic. This would help him to easily run the programs.

8.5 Meeting the requirements

Chapter 4 discussed the major issues that should be taken into account when designing or choosing an assignment management system. This information was used in Chapter 5, to evaluate various software solutions that could be used for assignment management and evaluation. This section will attempt to evaluate Assignment Manager 2000 and the evaluation application (Table 8.1) against the same criteria used in Table 5.1 in Chapter 5.

	Assignment Manager 2000
Platforms	
Unix	X
Windows NT	√
Mac	X
Security	
Access Control	√
Authentication	√
Integrity	√
Confidentiality	√
Non-repudiation	√
General	
Training	X
Support	X
Annual license fee (unlimited number of students)	free

	Assignment Manager 2000
Communication	
Calendar	√
Noticeboard	√
Assignment reminders	√ (the calendar highlights when the next assignments are due)
E-mail	X (the system e-mails tokens of submission to the students)
Newsgroups	X
Submission	
Assignment creation tools	√
Multiple file submission	√
Submission method	
• E-mail	X
• Web-interface	√
• FTP	X
• Mapped drive	X
Storage	
Submitted assignment storage	√
Evaluated assignment storage	√
Evaluation	
Assignment annotation	√
Online annotation	X
Offline annotation	√
Multiple file assignments	√
Programming specific features	
• Source code analysis	X
• Automated program testing	X
Feedback and retrieval	
Access to own grades	√
Assignment progress tracking	√ (tracks whether assignment has been submitted or evaluated)
Summative calculations	√

Table 8.1 Compares the features of the classroom management solutions.

8.5.1 Multiple file attachments

Assignment Manger 2000 allows students and lecturers to attach multiple files to a single submission. Furthermore, the attachment facility allows students to associate

the assignment files with an attachment group. This allows for the submission of assignments that consist of many programs, which in turn consist of many files. This enables lecturers to determine which files belong to which programs.

8.5.2 Security

Chapter 3 went into great detail explaining the role and importance of information security. Chapter 5 summarised that the main security requirements that need to be met are, authentication, nonrepudiation, access control, integrity and confidentiality. This section will briefly examine these requirements to determine how, or if, they were attained.

- **Integrity** - Integrity mechanisms ensure that the data has not been modified in any form during its storage or transit. Mechanisms must identify and correct or report incidents where the integrity of data was breached so that appropriate actions can take place (Bruce & Dempsey, 1997; CPS4001, 1998).

Assignment Manager 2000 uses Secure Sockets Layer to provide data integrity over the communications medium. This only checks to see if the submitted information is modified during transit and does not ensure that complete files were submitted. For this reason the students can verify the integrity of their submission using two methods. They can check to see if the submitted files, file sizes correspond to the original files, or they can download their submitted assignment files and compare them to the original files.

- **Access Control (Authorisation)** - Once an entity has been identified and authenticated, access control mechanisms ensure that an entity is only allowed to access authorised system resources. (Bruce & Dempsey, 1997; CPS4001, 1998). Assignment Manager 2000 uses a user ID, user role and a time dependant token to provide access control.

- **Confidentiality** - Confidentiality services ensure that the data is protected from disclosure to unauthorised entities. Students are often concerned about the public availability of their marks. Mechanisms that ensure the privacy of the students' information, including assignment content, should therefore be put in place. SSL provides for confidentiality over the communications medium.
- **Authentication** - Authentication mechanisms are used to verify that the users are who they claim to be. (Bruce & Dempsey, 1997; CPS4001, 1998). Assignment Manager 2000 uses a User ID and Password combination to authenticate the users. SSL could additionally provide for the authentication of the server, however this feature was not used.
- **Non-repudiation** - Mechanisms that support non-repudiation are used to prevent entities from denying that an action was taken or that a message was sent or received (Bruce & Dempsey, 1997; CPS4001, 1998). Non-repudiation mechanism must be in place to prevent the often-occurring problem of students who claim they have submitted their assignments, but evaluators who claim they never received them. Assignment Manager 2000 provides for non-repudiation by means of a token. When a student submits an assignment, a token is generated and e-mailed to the student as proof of submission.

8.5.3 Assignment annotation

Chapter 5 stated that the solution must be able to annotate assignments that consist of many files. The evaluation application discussed in Chapter 7 can be used to evaluate assignments that consist of many files. Moreover, the evaluation application enables evaluators to provide lengthy in-context commentary to assignment files. The evaluation application also allows evaluators to define generic error types for commonly occurring errors. This feature can save the evaluator time, as he or she is liberated from having to type out and explain a certain commonly occurring error to each student.

8.5.4 Assignment storage

Assignment storage refers to the permanent storage of student assignments. Assignment Manager 2000 stores both submitted and evaluated student assignments on a secure server and is additionally capable of storing multiple evaluations of an assignment. Furthermore, Assignment Manager 2000 records in a database how often pre-defined errors occurred within individual assignments.

8.5.5 Conclusion

The model described in Chapter 6 is highly flexible and generic. Most of the problems that Assignment Manager 2000 exhibited are implementation related and should not negatively reflect upon the model itself.

This section discussed whether the prototype met with the requirements which were used in Chapter 5's evaluation of the commercially available solutions. Consequently, the prototype does meet with these requirements. The remainder of this chapter will therefore discuss further areas of research and development.

8.6 Future Research and Development

The following section will concentrate on some of the areas of research and development that could positively influence future implementations and refinements of the model.

8.6.1 Integrated mark-sheets

Systems such as Mindtrail (2000) (Chapter 5) make use of mark sheets to provide feedback on evaluated assignments. In the traditional sense, a marksheet can be described as a page that contains set criteria according to which an assignment is evaluated. For example, a marksheet may define that a user can attain between 1 and 10 marks for the style of their submitted program. Each of these criteria is then added to produce a final mark. Such systems, on their own, are inadequate for use within programming assignments. This is because they do not provide facilities for in-context commentary on the code. It would, however, be beneficial to integrate electronic mark-sheets into the evaluation application. Implementing a generic mark-sheet enabled system within Assignment Manager 2000, would, however, require one to embed the mark sheet within each assignment. This is because one may evaluate different assignments according to different criteria, which means that one would not want a “fixed” mark sheet. Research into how one should do this is needed.

8.6.2 Online evaluation

One of the requirements that the lecturers set was the ability to evaluate assignments offline. However, online evaluation does have various advantages over offline evaluation. For example lecturers don't have to worry about managing assignment files on their computers, and the submission process is greatly simplified. It would therefore be useful if lecturers had the facility to evaluate assignments online as well as offline. Online evaluation could be implemented by using an Active-X control or a Java applet-based version of the evaluation application

8.6.3 Plagiarism

Students often copy assignments from one another. It would therefore be beneficial if such acts of plagiarism were automatically detected by the system. This feature can be

implemented, because, all of the submitted assignments are permanently stored on the server. Furthermore, plugging such a facility into the current system should be relatively easy, as all the files are stored in their original format.

8.6.4 Graphics capabilities

Although the evaluation application provides evaluators with in-context annotation facilities, the application still lacks drawing features. Such a feature would greatly enhance the level of feedback that evaluators can provide to the students. For example, such a feature would enable evaluators to draw a circle around a concept and an arrow showing where it should be moved to. A problem with implementing such a feature is that it would, in most cases, require the students to use specialised software to view the evaluated files instead of a web-browser. A possible method of doing this is by converting the actual drawings that the evaluators make into .GIF files. The .GIF files can then be overlaid over normal HTML text by using Dynamic HTML (DHTML). This approach could however consume a lot of space if the evaluator made a large number of drawings. Research into compression techniques might help overcome these problems.

8.6.5 Trend analysis

As illustrated in Chapter 7, the database is used to store detailed information pertaining to the mistakes that students have made in their assignments. Different queries could be developed to discover trends in the mistakes that students make. Evaluators could use such facilities to discover general problem areas that most students share, and specific problems of individual students. Data mining methods might be used here.

8.6.6 Standardisation of file formats

As noted in Chapter 7, there are numerous problems with using XML as the file format in which the web component exchanges assignments with the evaluation application. Further investigation should therefore be undertaken to determine how a standard file format such as XML could be implemented within the current solution.

8.6.7 File Uploading

The questionnaire made it apparent that students felt that the current method of attaching assignments is still too time consuming. As stated in Chapter 7 this is mainly due to limitations within current web browsers. A possible way to enable selecting of multiple files, in a single step, is to use an Active-X control to upload the files. This was originally considered, however a problem with using Active-X controls is that they offer potential security risks to users' machines as they can access all of the resources on them. The onus of verifying the authenticity of the control is thus put on the user.

8.6.8 Non-repudiation

The model proposed in this dissertation makes use of tokens that are e-mailed to the students to prevent non-repudiation. This approach does not however take into account the possibility of delivery failure of the token. Research should therefore be undertaken to determine possible solutions to this problem. A possible workaround would require the students to acknowledge that they have received the token. The students could do so by signing the token with their private key, and then by e-mailing the signed token back to the assignment management system. A problem with this approach is that it requires the students' direct participation within the non-repudiation system. Furthermore the students may purposely decide not to reply. Therefore if such an implementation were to be used, it should be made policy that

only those assignments belonging to student who have acknowledged the receipt of the token be marked.

8.6.9 Extending the user base

Assignment Manager 2000 was especially customized for use with programming assignments. The system should however be expanded to add support for non-programming related assignments. The only alteration that the existing system would need is to rename some of the terms used within Assignment Manager 2000. For example, instead of stating that a group of attachments belong to a specific program, they could belong to an attachment group. Furthermore, research should be undertaken to determine how the system could be customised for use in distance education-based classes.

8.6.10 Extra features

Assignment Manager 2000 was developed to meet the basic needs of an assignment management system within programming classes. A more refined version therefore needs to be developed. In particular it does not support group submission of assignments. This is due to various issues that need to be resolved. The first issue is whether groups are predefined within the database, or whether students must specify authors of the submission during the submission process. This approach brings another point of contention and that is assignment ownership. This becomes important when certain members of a group want to resubmit an assignment before the due date. Finally the distribution of marks between members of a group becomes difficult. Will all of the members receive the same mark, or will each member receive a mark according to their contribution to the assignment? This would require the students to specify which components of the assignment each of the groups' members contributed to. An evaluation application would therefore have to recognise that a particular assignment had for example five authors, and provide the lecturer with the tools to give each of the authors a particular mark.

8.7 Summary

The model presented in Chapter 6, provides an infrastructure for the secure management and evaluation of electronically submitted assignments. As describe in the chapter, it can be implemented in numerous ways each of which can provide the needed level of security and functionality. The viability of the model was tested by implementing a prototype of a web-based version. The prototype was evaluated by students and a lecturer. Their evaluations were generally positive.

The objectives of this dissertation, as set out in Chapter 1 were:

- *“Determine how the flow of programming assignments between students and lecturers can be streamlined using current technologies;”*

The PAMM model described a method of streamlining the submission, evaluation and feedback of assignments. The process of assignment submission, storage and retrieval has been simplified in Assignment Manager 2000. Students and lecturers can submit and retrieve assignments at all times. Furthermore, submitted and evaluated assignments are automatically stored for later retrieval. The students and the lecturers are also freed from having to print out the assignments, thus saving time and money.

- *“To ascertain how rich, intuitive feedback can be attained in programming assignments that are evaluated electronically;”*

The PAMM model and the prototype evaluation application are capable of handling multiple file assignments and can be used to provide in context annotation. The evaluation also allows lecturer to predefine commonly occurring errors, saving the lecturer time. The lecturer is also presented with historical information about how often certain pre-defined errors occurred in individual students assignments. This feature can aid the lecturer in the evaluation process,

thereby improving feedback. Students can also privately retrieve, view and print their evaluated assignments over the Internet.

- *“To determine how the submission, storage and retrieval of assignments can be secured within an electronic environment which is trusted by lecturers and students.”*

The entire process of assignment submission, storage and retrieval takes place within a secure environment. Section 8.5.2 described how access control, authentication, integrity, confidentiality and nonrepudiation are all maintained within Assignment Manager 2000.

The PAMM model and Assignment Manager 2000 therefore meet the objectives set out in Chapter 1, and proves the viability a secure assignment management model.

References

Adelsberger, H. H., Körner, F. & Pawlowski, J. M. (1998). A Conceptual Model for an Integrated Design of Computer Supported Learning Environments and Workflow Management Systems. Proceedings of Teleteaching '98: Distance Education, Training and Education, XV. IFIP World Computer Congress. Vienna. pp. 55 – 64.

Apache. (2000). Apache.org. Retrieved 2 October 2000 from the World Wide Web:<http://www.apache.org>

Associated Press. (1999). U.S. relaxes encryption restrictions. Retrieved 3 March 2000 from the World Wide Web: <http://www.usatoday.com/life/cyber/tech/cte066.htm>

Arzt, N. H. (2000). Facilitating Instructor-Student Collaboration on the Internet. Retrieved 22 August 2000 from the World Wide Web: <http://www.upenn.edu/computing/help/collab/index.html>

Baker, C. & Buller, T. (1995). The global classroom why K-12 needs it. In Berge, Z.L. & Collins, M.P. (Eds.). Computer mediated communication and the online classroom volume III: Distance learning. Cresskill, New Jersey : Hampton Press, INC. p. 195.

Berge, Z. L. & Collins, M. P. (1995a). Computer mediated communication and the online classroom volume I: Overview and perspectives. Cresskill, New Jersey : Hampton Press, INC. p. 3.

Berge, Z.L. & Collins, M. P. (1995b). Computer mediated communication and the online classroom volume III: Distance learning. Cresskill, New Jersey : Hampton Press, INC. pp. 13 – 30.

Bernstein, T., Bhimani, A. B., Schultz, E. & Siegel, C. A. (1996). Internet security for business. New York: John Willey & Sons, Inc.

Blanchard, E. (2000). Newsgroups for education. Retrieved 22 August 2000 from the World Wide Web: <http://www.cadvision.com/blanchas/education/news/news0.htm>

Bock, G. E. & Marca. D. A. (1995). Designing groupware.: A guidebook for designers, implementors, and users. New York: McDraw-Hill. p. 12.

Borja, H. M. (1998). Automating the Support of Collaboration in Distance Education. Retrieved 9 July 1999 from the World Wide Web: <http://www.cs.ucf.edu/~morelos/prospectus/Chapter3.html>

Brinck, T. (1998). Introduction to groupware. Retrieved 8 July 1999 from the World Wide Web: <http://www.usabilityfirst.com/groupware/intro.html>

Britain, S. & Liber, O. (1999). A Framework for Pedagogical Evaluation of Virtual Learning Environments. Retrieved 21 June 2000 from the World Wide Web: <http://www.jtap.ac.uk/reports/htm/jtap-041.html>

Bruce, G. & Dempsey, R. (1997). Security in distributed computing : Did you lock the door. New Jersey : Prentice-Hall.

Ciolek, T. M. (2000). The Internet in 2000: Opportunities and disadvantages of scholarly work. Retrieved 7 March 2001 from the World Wide Web: <http://www.ciolek.com/PAPERS/brainstorm-mar00.html>

CNET News.com. (1997). By Lash, A. 40-bit crypto proves no problem. Retrieved 24 May 1999 from the World Wide Web: <http://www.news.com/News/Item/0,4,7483,00.html>

Coleman, D. (1995). Groupware: Collaborative strategies for corporate LANs and intranets. Cambridge, Ontario : Prentice Hall

Commonwealth of Pennsylvania. (2000). Beginner's guide to the Internet for educators. Retrieved 23 December 2000 from the World Wide Web: <http://205.146.39.13/linktuts/bgchat.htm>

Cooper, F. J., Goggans, C., Halvey, J. K., Hughes, L., Morgan, L., Siyan, K., Stallings, W. & Stephenson, P. (1995). Implementing Internet security. Indianapolis: New Riders Publishing

Counterpane Internet Security, Inc., (2000). Counterpane Labs Releases Windows 95-compatible S/MIME 40-bit RC2 Cracking ScreenSaver. Retrieved 29 August 2000 from the World Wide Web: <http://www.counterpane.com/smime.html>

CPS4001. (1998). ISO 7498-2 Information Technology Committee. (1998). Managing Security of Information. Retrieved 19 June 1999 from the World Wide Web:<http://www.ifac.org/StandardsAndGuidance/InformationTechnology/ManagingSecurityOfInfo.html>

Day, M. & Batson, T. (1989). The network-based writing classroom: The ENFI idea. In Berge, Z.L. & Collins, M.P. (Eds.). Computer mediated communication and the online classroom volume II: Higher education. Cresskill, New Jersey : Hampton Press, INC. pp. 25 – 46.

December, J. (1997). Notes on defining of Computer-Mediated Communication. Retrieved 19 February 1999 from the World Wide Web: <http://www.december.com/cmc/mag/1997/jan/december.html>

Dumas, A. (1995). Programming Winsock. United States of America: Sams Publishing

Dwyer, D., Barbieri, K. & Doerr, H., M. (1995). Creating a Virtual Classroom for Interactive Education on the Web. Retrieved 19 February 1999 from the World Wide Web:

<http://www.igd.fhg.de/www/www95/proceedings/papers/62/ctc.virtual.class/ctc.virtual.class.html>

Evard, R. (1993). Collaborative Networked Communication: MUDs as Systems Tools. Proceedings of the Seventh Systems Administration Conference (LISA VII), Monterey, CA. pp. 1 – 8.

Fanderclai, T. L. (1995). MUDs in Education: New Environments, New Pedagogies. Computer-Mediated Communication Magazine, 2(1), p.8.

Flake, J. L. (1996). The World Wide Web and Education. Retrieved 22 February 1999 from the World Wide Web: <http://mailer.fsu.edu/~jflake/WWWEd.html>

FutureU. (2000). Comparative Features Analysis of Leading Course Management Software. Retrieved 9 April 2000 from the World Wide Web: http://www.futureu.com/cmscomp/cms_comp.html

Gibson, E., Brewer, P., Dholakia, A., Vouk, M., & Bitzar, D. (2000). A Comparative Analysis of Web-Based Testing and Evaluation Systems. Retrieved 17 June 2000 from the World Wide Web:
<http://renoir.csc.ncsu.edu/MRA/Reports/WebBasedTesting.htm>

Gibson, J. (1997). Evaluation of a Trial of Internet Teaching. Proceedings of Third Australian World Wide Web Conference, Retrieved 23 May 2000 from the World Wide Web:

<http://ausweb.scu.edu.au/proceedings/gibson/paper.html>

Gold, J. A. (1996). Enhancing Professional Development in Special Education Through the Web. CMC magazine October 1996. Retrieved 22 February 1999 from the World Wide Web: <http://www.december.com/cmc/mag/1996/oct/gold.html>

Gottschalk, T. H. (1995a). Guide #6 Computers in Distance Education, Distance Education at a Glance. Retrieved 22 February 1999 from the World Wide Web: <http://www.uidaho.edu/evo/dist6.html>

Gottschalk, T. H. (1995b). Guide #7 Computers in Distance Education, Distance Education at a Glance. Retrieved 22 February 1999 from the World Wide Web: <http://www.uidaho.edu/evo/dist7.html>

Hammond, M. (2000). Oracle, Sybase move to boost database encryption. Retrieved 30 September 2000 from the World Wide Web: <http://www.zdnet.com/zdnn/stories/news/0,4586,2422423,00.html>

Hazari, S. I. (1998). Evaluation and selection of web course management tools. Retrieved 17 June 2000 from the World Wide Web: <http://sunil.umd.edu/webct>

Hills, M. (1997). Intranet as groupware. 605 Third Avenue, New York: Wiley Computer Publishing, John Wiley & Sons, Inc.

Hiltz, S. R. (1995). Teaching in a Virtual Classroom [TM]. International Conference on Computer Assisted Instruction, ICCAI'95. Retrieved 19 February 1999 from the World Wide Web: <http://www.njit.edu/CCCC/VC/Papers/Teaching.html>

Holmes. (1999). Markin. Retrieved 2 November 1999 from the World Wide Web: <http://www.cict.co.uk/software/markin/index.htm>

Hughes, Jr. L. J. (1995). Actually useful Internet security techniques. New Riders Publishing.

Human-Computer Interaction. (2000). Theories and Algorithms in Virtual Reality and Multi-Dimension Information Visualization. Retrieved 21 April 1999 from the World Wide Web: <http://www.imd.cims.edu.cn/English/tech/hci.htm>

Ihrer, K. (2000). Database security: Securing Oracle: New utilities and third-party "add-on" tools take security in Oracle 8.1.6 to the next level. Retrieved 30 September 2000 from the World Wide Web:
<http://www.infosecuritymag.com/sep2000/databasesecurity.htm>

Information technology security team. (1998). The University of Iowa enterprise security architecture. Retrieved 4 September 1999 from the World Wide Web:
<http://www.uiowa.edu/~security/docs/esa.html>

Jaspan, B. (1995). Kerberos Users' Frequently Asked Questions 1.14. Retrieved 14 September 1999 from the World Wide Web: <http://www.faqs.org/faqs/kerberos-faq/user/>

Jones, D. & Buchanan, R. (1996). The Design of an Integrated Online Learning Environment, Proceedings of ASCILITE 96. Retrieved 29 July 1999 from the World Wide Web: <http://www.ascilite.org.au/conferences/adelaide96/papers/11.html>

Jones, D. & Jamieson, B. (1997). Three Generations of Online Assignment Management, Rod Kevill, Ron Oliver, Rob Phillips (Eds.), What Works and Why. Proceedings of ASCILITE 97.
Retrieved 25 November 1999 from the World Wide Web:
http://webfuse.cqu.edu.au/Information/Resources/Readings/Assignment_Submission/tree/

Jones, D. & McCormack, C. (1997). Class management: The forgotten task. In H. Greenberg & R. Hall (Eds.), Shortening the Distance to Education, Proceedings of the 3rd International North American Web Conference. pp. 109-125.

Joy, M. & Luck, L. (1995). On-line submission and testing of programming assignments. Retrieved 29 July 1999 from the World Wide Web:
<http://www.dcs.warwick.ac.uk/cobalt/publications/seda.html>

Kaplan, H. (1998). Building Your Own Web Course: The Case for Off-the-Shelf Component Software. CAUSE/EFFECT journal, 21(4). Retrieved 4 September 1999 from the World Wide Web: <http://www.educause.edu/ir/library/html/cem9849.html>

Kaufman, C., Perlman, R. & Speciner, M. (1995). Network security: Private communication in a public world. Indianapolis: New Riders Publishing

Kinner, J. & Coombs, N. (1995). Computer access for students with special needs. In Berge, Z.L. & Collins, M.P. (Eds.). Computer mediated communication and the online classroom volume I: Overview and perspectives. Cresskill, New Jersey : Hampton Press, INC. pp. 34 -52.

Kristapiazzi, G. (1998). Compare Web Tools for Course Authoring. contracted report for Daytona Beach Community College. Retrieved 7 December 1999 from the World Wide Web:
http://www.geocities.com/Eureka/Gold/6012/compare_web_tools.htm

Lennon, J. & Maurer, H. (2000). Applications and impact of hypermedia systems: An overview. Retrieved 14 August 2000 from the World Wide Web: http://www.jucs.org/jucs_0_0/applications_and_impact_of/html/paper.html

Lewis, J., Whitaker, J. & Julian J. (1995). Distance education for the 21st century: The future of nation and international telecomputing networks in distance education. In Berge, Z.L. & Collins, M.P. (Eds.). Computer mediated communication and the online classroom volume III: Distance learning. Cresskill, New Jersey : Hampton Press, INC. pp. 13 – 30.

Lotus Corporation. (1999). Lotus Learningspace. Retrieved 10 March 1999 from the World Wide Web : URL <http://www.lotus.com/learningspace>

Martin, L. (1997). WWWAssign. Retrieved 8 October 1999 from the World Wide Web: <http://www.admin.northpark.edu/lmartin//WWWAssign/>

Mason, D. & Woit, D. (1999). Providing Markup and Feedback to Students with Online Marking. Retrieved 17 June 2000 from the World Wide Web:
<http://www.sarg.ryerson.ca/sarg/papers/199903-sigcse99/index.phtml/sarg.html>

Microsoft Corporation. (2000). Microsoft home page. Retrieved from the World Wide Web: <http://www.microsoft.com>

MindTrail. (2000). MindTrail home page. Retrieved from the World Wide Web:
<http://www.mindtrail.com>

Murray, P. J. (1997). A rose by another name. CMC magazine January 1997. Retrieved 29 July 1999 from the World Wide Web:
<http://www.december.com/cmc/mag/1997/jan/murray.html>

Netcraft. (2000). The Netcraft web server survey. Retrieved from the World Wide Web: <http://www.netcraft.co.uk/Survey/>

North, S. & Hermans, P. (1999). SAMS teach yourself XML in 21 days. Indianapolis, Indiana, USA: SAMS.

Nouwens, F. & Towers, S. (1997). Assessment in distance education, Cited in S. Dixon, G. Lefoe, F. Nouwens and S. Wills (Eds.) Teaching at a Distance [CD-ROM]. PAGE: Melbourne. Retrieved 17 March 2000 from the World Wide Web:
<http://cedir.uow.edu.au/programs/flexdel/resources/AssessmentDistEd.html>

O'Hara, K. & Sellen, A. (1997). A Comparison of Reading Paper and On-Line Documents. Retrieved 17 September 2000 from the World Wide Web:
<http://www.acm.org/sigchi/chi97/proceedings/paper/koh.htm>

Parkland College. (2000). Modules for Internet course development: Integrated course management software. Retrieved 2 July 2000 from the World Wide Web:
<http://online.parkland.cc.il.us/modules/m2/integrated.htm>

Paulsen, M. F. (1995). The Online Report on Pedagogical Techniques for Computer-Mediated Communication. Retrieved 14 February 1999 from the World Wide Web: <http://www.nettskolen.com/alle/forskning/19/cmcped.html>

PhotoPoint Corporation. (2000). PhotoPoint home page. Retrieved 3 January 2000 from the World Wide Web: <http://www.photopoint.com>

Rada, R. (1999). Workflow Management in Virtual Education. Retrieved 13 June 2000 from the World Wide Web: <http://www.enable.evitech.fi/enable99/papers/rada/rada.html>

Rein, L. (1998). Handling Binary Data in XML Documents. Retrieved 2 December 1999 from the World Wide Web: <http://www.xml.com/pub/98/07/binary/binary.html>

Rohfeld, R. W. & Hiemstra, R. (1995). Moderating discussions in the electronic classroom. In Berge, Z.L. & Collins, M.P. (Eds.). Computer mediated communication and the online classroom volume III: Distance learning. Cresskill, New Jersey Hampton Press, INC. pp. 91-104.

Rowley, K. (1995). Understanding Software Interoperability in a Technology-Supported System of Education. CAUSE/EFFECT journal, 8(3). pp. 20-26.

Rowntree, D. (1995). Teaching and learning online: A correspondence education for the 21st century? British Journal of Educational Technology. pp.205-215.

RSA Data Security, Inc. (1997). Swiss-based Ph.D. Student Solves 48-bit Key in RSA Data Security's Secret-Key Challenge. Retrieved 4 June 1999 from the World Wide Web: <http://www.rsa.com/pressbox/html/970214.html>

RSA Data Security, Inc. (1999). S/MIME frequently asked questions. Retrieved 4 June 1999 from the World Wide Web: <http://www.rsasecurity.com/standards/smime/faq.html>

Schneider, D. K. (2000). Educational Technology: Educational VR (MUD) sub-page. Retrieved 2 July 2000 from the World Wide Web: <http://tecfa.unige.ch/edu-comp/WWW-VL/eduVR-page.html>

Sellen, A. & Harper, R. (1997). Paper as an Analytic Resource for the Design of New Technologies. Retrieved 13 August 2000 from the World Wide Web: <http://www.acm.org/sigchi/chi97/proceedings/paper/ajs.htm>

Shield, G. (1996). Formative influences on technology education: The search for an effective compromise in curriculum innovation. Journal of Technology Education. Fall 1996, pp. 50-60.

Shimabukuro, J. N. (1995). CMC and writing instruction: A future scenario. In Berge, Z.L. & Collins, M.P. (Eds.). Computer mediated communication and the online classroom volume I: Overview and perspectives. Cresskill, New Jersey : Hampton Press, INC. pp. 37-52.

SSH Communications Security Ltd. (1999a). Cryptographic Algorithms : Public key algorithms. Retrieved 20 June 1999 from the World Wide Web: <http://www.ssh.fi/tech/crypto/algorithms.html>

SSH Communications Security Ltd. (1999b). Introduction to cryptography : Basic cryptographic algorithms. Retrieved 20 June 1999 from the World Wide Web: <http://www.ssh.fi/tech/crypto/intro.html>

Szabo, M. (1998). Survey of instructional technology research. Retrieved 17 June 2000 from the World Wide Web: <http://www.quasar.ualberta.ca/edmedia/readingsnc/Nrefsza.html>

Szabo, M. & Fuchs, A. (1998). Computer Managed Instruction. Retrieved 7 March 1999 from the World Wide Web: <http://www.quasar.ualberta.ca/edmedia/TIES/I5.html>

TechNet. (2001). Windows 2000 security technical overview. Retrieved 5 March 2001 from the World Wide Web:

<http://www.Microsoft.com/technet/win2000/win2ksrv/sectech.asp>

Thomas, R. N. (1997). The paperless classroom? In Duber, J. (Eds). TESL-EJ Vol. 3 No. 1, November 1997. Retrieved 17 February 1999 from the World Wide Web:

<http://www-writing.berkeley.edu/TESL-EJ/ej09/int.html>

van der Veen, J., Jones, V. & Collis, B. (1998). Workflow Applied to Projects in Higher Education. Proceedings of COOPIS'98. Retrieved 21 April 1999 from the World Wide Web: <http://wwwhome.ctit.utwente.nl/~vdveenj/ny/anaxny.html>

Warschauer, M. (1997). Computer-mediated collaborative learning: Theory and practice. Retrieved 2 July 2000 from the World Wide Web:

<http://www.lll.hawaii.edu/web/faculty/markw/cmcl.html>

WBT systems. (2000). TopClass overview. Retrieved 19 June 2000 from the World Wide Web: <http://www.wbtsystems.com/products/overview.html>

WCB. (2000). WCB home page. Retrieved 21 February 2001 from the World Wide Web: <http://madduck.mmd.vcu.edu/wcb/wcb.html>

WebCT. (2000). WebCT home page. Retrieved 21 February 2001 from the World Wide Web: <http://www.webct.com/>

Webopedia. (1998). Digital certificate. Retrieved 3 February 2000 from the World Wide Web: http://webopedia.internet.com/TERM/d/digital_certificate.html

Wegener, D. P. (2000). Basic terms of computer based training. Retrieved 21 April 2000 from the World Wide Web : <http://www.delweg.com/dpwessay/cbterms.htm>

WFMC-TC-1011. (1996). Workflow Management Coalition terminology & glossary. Workflow Management Coalition. Retrieved 13 March 1999 from the World Wide Web : URL <http://www.aiim.org/wfmc/standards/docs/glossary.pdf>.

Woodruff, M. & Mosby, J. (1996). A brief description of videoconferencing. Videoconferencing in the classroom and library. Retrieved 3 July 2000 from the World Wide Web: <http://www.kn.pacbell.com/wired/vidconf/description.html#what>.

Yankelovich, N., Meyrowitz, N. & van Dam, A. (1991). Reading and writing the electronic book. In Barrett, O.B. & Scanlon, E. (Eds.). Computers and learning. Cresskill, New Jersey : Hampton Press, INC. pp. 235-249.

Zakon, R. H. (2000). Hobbes' Internet timeline. Retrieved 27 November 2000 from the World Wide Web: <http://www.isoc.org/zakon/Internet/History/HIT.html>

Zin, A. M. & Foxley, E. (2000). Automatic program assessment system. Retrieved 10 August 2000 from the World Wide Web:
<http://www.cs.nott.ac.uk/~ceilidh/papers/ASQA.html>

Appendix A

Questionnaire

This appendix contains a copy of the questionnaire given to the students to help evaluate the prototype system.

Electronic Assignment Submission and Retrieval Questionnaire

1. Did you have any difficulties logging onto the system?

always	sometimes	never
--------	-----------	-------

2. How easy or difficult was it to learn how to use the system?

very difficult	difficult	average	easy	very easy
----------------	-----------	---------	------	-----------

3. How would you compare electronic to paper based assignment submission?

paper best	prefer paper	no preference	prefer electronic	electronic best
------------	--------------	---------------	-------------------	-----------------

4. How do you feel about receiving electronically evaluated assignments compared to paper ?

paper best	prefer paper	no preference	prefer electronic	electronic best
------------	--------------	---------------	-------------------	-----------------

5. What did you like most about the system?

6. What did you disliked the most about the system?

7. What changes should be made to improve the system?

8. Do you see a potential use for electronic-based assignment submission?

yes	no
-----	----

Thank you for completing the questionnaire

Appendix B

User Manual: Assignment Manager 2000

This appendix includes a user manual for Assignment Manager 2000.

B.1 Introduction

Assignment Manager 2000 is a software package which can be used to facilitate electronic assignment submission and evaluation. The package consists of two independent components, a web component and an evaluation component. The web component is used for the storage, management and retrieval of assignments. The evaluation component can be used to help evaluate assignments. Although the components are independent of one another, certain time saving features require the use of both products.

The manual will firstly deal with the installation and configuration of the web component. This will be followed with the discussion of the administrative functions of the web component, after which the lecturers' and students' interfaces will be looked at. Finally the basic operation of the evaluation application will be explained. Some headings will have a diagram next to them. The purpose of these diagrams is to show the user what button needs to be pressed to activate that function.

B.2 Installation and configuration of Apache and PHP

The web component was written in a server side scripting language PHP and therefore requires PHP to reside on the web server. PHP can be obtained as a CGI executable or as a module for Apache (Win32 and Unix) or IIS. The following section will explain how Apache needs to be configured for use with the web component. The settings listed below are the changes that need to be made to the Apache configuration file (httpd.conf). The values displayed are those used to install the prototype at PE Technikon, and should be changed accordingly.

```
#ServerName IP or DNS of the web server
ServerName http://submit.petech.ac.za/
```

DocumentRoot The location of the web pages on the server
DocumentRoot "C:/Program Files/Apache Group/Apache/htdocs"

This tells the web server that .phtml files are executed with php.exe
AddType application/x-httpd-php .phtml
Action application/x-httpd-php "/php/php.exe"

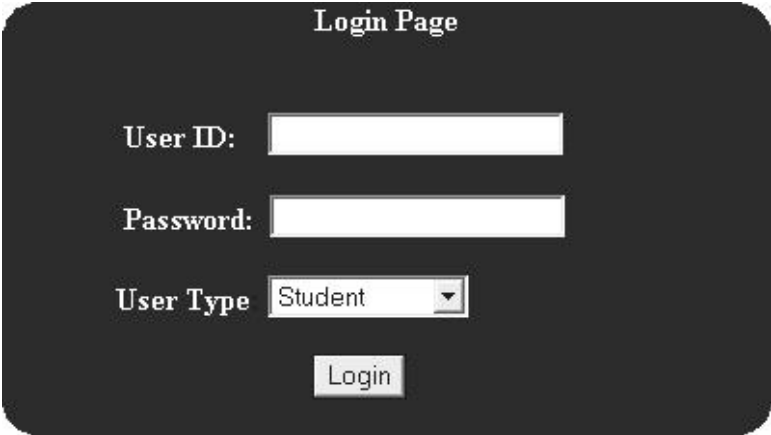
PHP's installation and configuration is well documented, however it is worth noting that the following changes must also be made. PHP's config file (for win32) is located within the windows directory.

doc_root The location of the web pages, and scripts on the server
doc_root = C:\Program Files\Apache Group\Apache\

upload_max_filesize Used to specify maximum file upload size, set to 2 Meg
upload_max_filesize = 2097152

SMTP This is used to specify the SMTP server that will be used for e-mail
SMTP = mail.petech.ac.za ;for win32 only

B.3 Login



The image shows a dark-themed login form. At the top, the title "Login Page" is centered. Below the title, there are three rows of input fields. The first row is labeled "User ID:" followed by a white text input box. The second row is labeled "Password:" followed by a white text input box. The third row is labeled "User Type" followed by a dropdown menu with "Student" selected. Below the dropdown menu is a "Login" button.

Figure B.1 A section of the login screen.

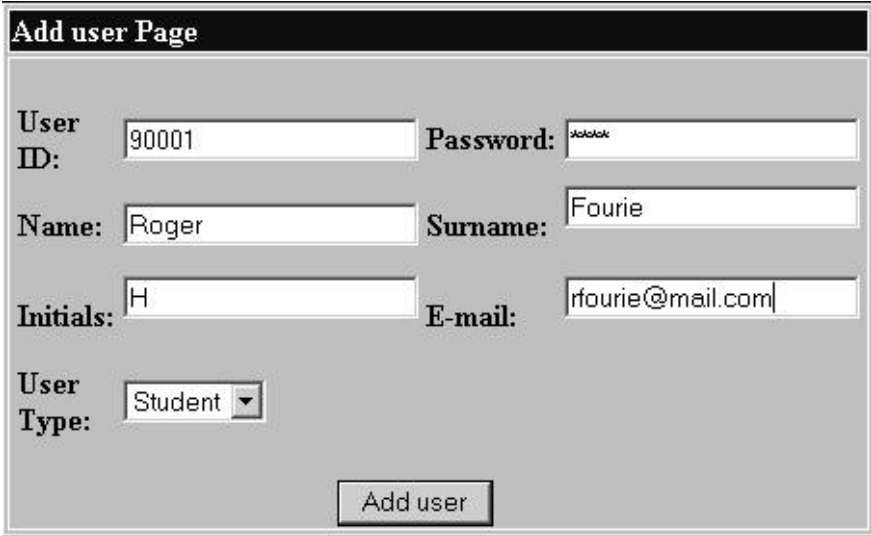
Access to the web component is restricted via a login screen (Figure B.1). The login screen prompts the user to enter a user ID, password and user type. The web component supports three different types of users: *administrators*, *lecturers* and *student*. A dropdown list of user types is used to select the role the user will perform for the duration of the session. The different user types are used to maintain access control to the web component. This enables students and lecturers to only access resources that their user types are authorized to access.

B.4 The Administrative Functions Of The Administrator

A user in the role of administrator can create and edit users and classes. The following section will briefly describe how to perform the main administrative functions of the administrator.

Add User 

An administrator can add users one at a time, by using the *add user* form (Figure B.2). The form requires the users details, together with a unique user id and user type. The user type field is used to determine the users overall access rights. The access rights are hierarchical in nature, meaning than an administrator could log in as a lecturer or student, or a lecturer could log in as a student.



Add user Page

User ID: Password:

Name: Surname:

Initials: E-mail:

User Type:

Figure B.2 Add user form.

Add Class

Create class

Class Group ID: SubjectCode:

Class Description:

Note, the SubjectCode is case sensitive, if it does not exist in the subject table an entry will be automatically created

Figure B.3 Add class form.

Classes are used to group users for a particular subject. Multiple classes are allowed for a particular subject, each with its own lecturer. The add class form, shown in Figure B.3, requires the subject code and the id for that particular class. If a subject with that particular subject code does not exist, one is automatically created.

Register User

A user can be placed in class by making use of the *register user* form (Figure B.4). The form requires the administrator to type in the users id, and to select the correct class group from the dropdown menu. The role that the user will play in the class is also specified.

Add user to class

User ID: Subject-ClassGroup:

UserRole:

Figure B.4 Register user in class form.

Automated User Registration

The initial registration of students and the subjects, for which they are registered, is automated by means of the auto registration function. This form requires the user to provide a comma-delimited file, which contains all of the needed particulars of the students for registration. The file's format must be as depicted in Table B.1. The students default password and e-mail address are automatically calculated from the User ID field.

User ID	Name	Surname	Initials	Subject Code	Class Group
---------	------	---------	----------	--------------	-------------

Table B.1 Required format for imported user registration.

B.5 The Lecturer

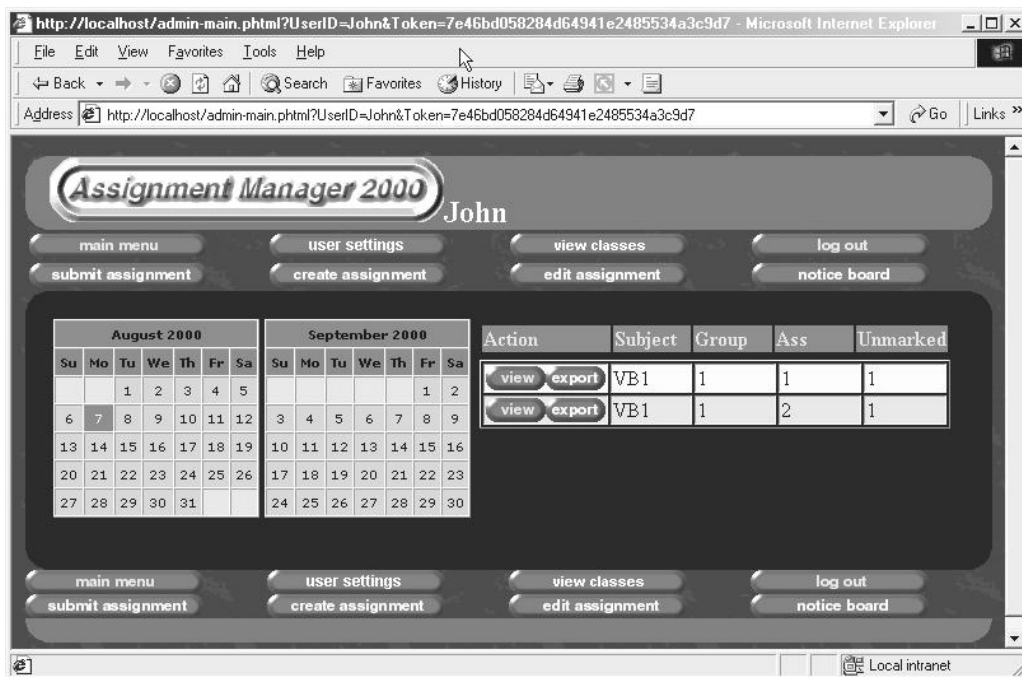
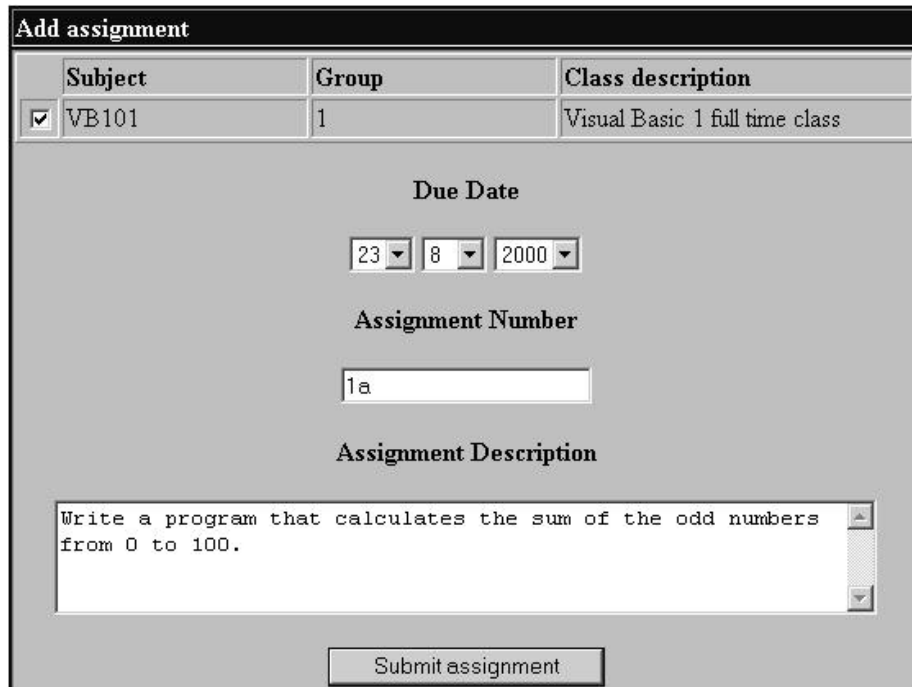


Figure B.5 Lecturer main menu.

The lecturer main menu, displayed in Figure B.5, consists of two calendars and a table, which lists the number of submitted assignments for a particular class assignment. The calendars display the current and forthcoming month, and indicate assignment due dates, as set by the lecturer. This is achieved by colouring days on which assignments are due, differently. In addition to this, if a lecturer moves the mouse cursor over a day when an assignment is due, a popup window appears with the details pertaining to the assignment(s).

Add Assignment

A lecturer can add an assignment by clicking on the *create assignment* button. He or she is then presented with a list of all the subjects that have been assigned to them. (See Figure B.6) The lecturer must select the classes for whom the assignment is intended and then fill in the assignment number, description and due date of the assignment. *Note: If multiple classes are selected when creating an assignment, a unique assignment is automatically created for each class. This allows lecturers to change due dates on a class bases if needed at a later stage.*



Subject	Group	Class description
<input checked="" type="checkbox"/> VB101	1	Visual Basic 1 full time class

Due Date

23 8 2000

Assignment Number

1a

Assignment Description

Write a program that calculates the sum of the odd numbers from 0 to 100.

Submit assignment

Figure B.6 Assignment creation form.

The system can accommodate multiple lecturers for a particular subject code, by assigning students for a particular subject to a particular class group. This enables lecturers to give their part-time students different assignments from full-time students, for example. Once the assignment has been created, an entry is automatically created on the students' calendars.

Retrieve Submitted Assignment


Assignments that were submitted by students can be retrieved using one of two methods. The preferred, and easiest method of retrieving assignments is done from the *main menu* page (Figure B.7). The table on the far right hand side lists the number of submitted assignments for a particular class and assignment. Each table entry has a *view* and *export* button. The export button  combines all of the unmarked assignments in a class into a single (.cpx) file. This file is then uploaded to the lecturer, who in turn can decide where to place it on their computers.



Figure B.7 A section of the lecturer's main menu.

The evaluation application is used to extract the combined assignments on the lecturer's computer. The view button is used to view the details of a particular class, such as the class details displayed in Figure B.8 that contains one submitted assignment. Each entry in the table represents a users assignment, offering the lecturer the ability to export and view individual assignments. The view button for each table entry is used to view individual assignment details. Therefore if the lecturer were to click on the view button in Figure B.8, Figure B.9, would be displayed.

Action	UserID	Submission Date
view export	9971649	2000-08-07
view export	9941741	2000-08-07
view export	9901349	2000-08-07

export all assignments

Figure B.8 Displays the submitted assignments for a particular class and assignment.

Export

Hope you are happy with this

FileName	ProgramID	FileSize
frmMain_frm	0	102817
Module1.bas	0	140

Figure B.9 Displays the details of a submitted assignment.

Figure B.9 displays the details of a particular student’s assignment submission. The white text area contains the student’s comment that was made during submission. The submitted files are listed in a table below this area. The lecturer can download any particular file from the assignment by clicking on the filename, or download them all as a single file by clicking on the export button (creating a .svl file); however, as stated before, using the export function requires the use of the evaluation application for file extraction. This is explained in B.7.

Submit Evaluated Assignment submit assignment

Once a lecturer has evaluated an assignment, he or she must submit the evaluated assignment. Two forms of assignment submission exist, namely manual submission and automated submission (Figure B.10). Automated submission can be used to

submit individual or groups of assignments that were evaluated in the evaluation application. The lecturer simply has to select the file(s) that he or she has evaluated and add them to the form. The web component will automatically determine the assignment details from the embedded information contained within the file(s).

Automated assignment submission

Attachment:

Browse...

File attachments

Instructions: Click the **Browse** button to locate the file you need, and select it. The file path will appear in the attachment field. Next, click **Attach** to attach the selected file to your message. When you send your message, the attached file is automatically enclosed

Submit assignment

Figure B.10 Automated evaluated assignment submission form.

The manual submission form (Figure B.11) is accessed by clicking on the manual submit button on the automated submission form. Manual submission should be used when the evaluation application is not used to evaluate the assignment. Consequently certain details about the assignment, such as the errors made, cannot be determined. The evaluator must select the user and class to which the assignment belongs, together with the mark attained. The files out of which the assignment consists must also be attached to the form.

Submit evaluated student assignment

Student ID: 90001-VB101-1-1a Mark 66

Remarks

Well done

Attachment: Browse... Attach

Program: - Program 1 -

File attachments
D:\TEMP\note.txt

Instructions: If applicable, select the appropriate Program from the **Program select** box. Then click the **Browse** button to locate the file you need, and select it. The file path will appear in the attachment field. Next, click **Attach** to attach the selected file to your message. When you send your message, the attached file is automatically enclosed

Submit assignment

Figure B.11 Manual evaluated assignment submission form.

Notice Board notice board

Add Notice

	Subject	Group	Class description
<input checked="" type="checkbox"/>	VB101	1	Visual Basic 1 full time class

Notice Topic


Test 1

Notice

The test on 30-08-2000, will be on chapters 2,3 and 5

Add notice

Figure B.12 Add notice form.

The notice board, as the name implies, acts as an electronic version of a notice board. Lecturers can use it to notify students about various events such as tests or class cancellations. To add a notice the lecturer must click on the *add notice* button,  located within the notice board screen. The add notice form (Figure B.12) requires the lecturer to select the class group(s) for whom the notice is intended, as well as a topic and the actual notice text. Once a notice has been added students will see it in their main menu.

B.6 The Student

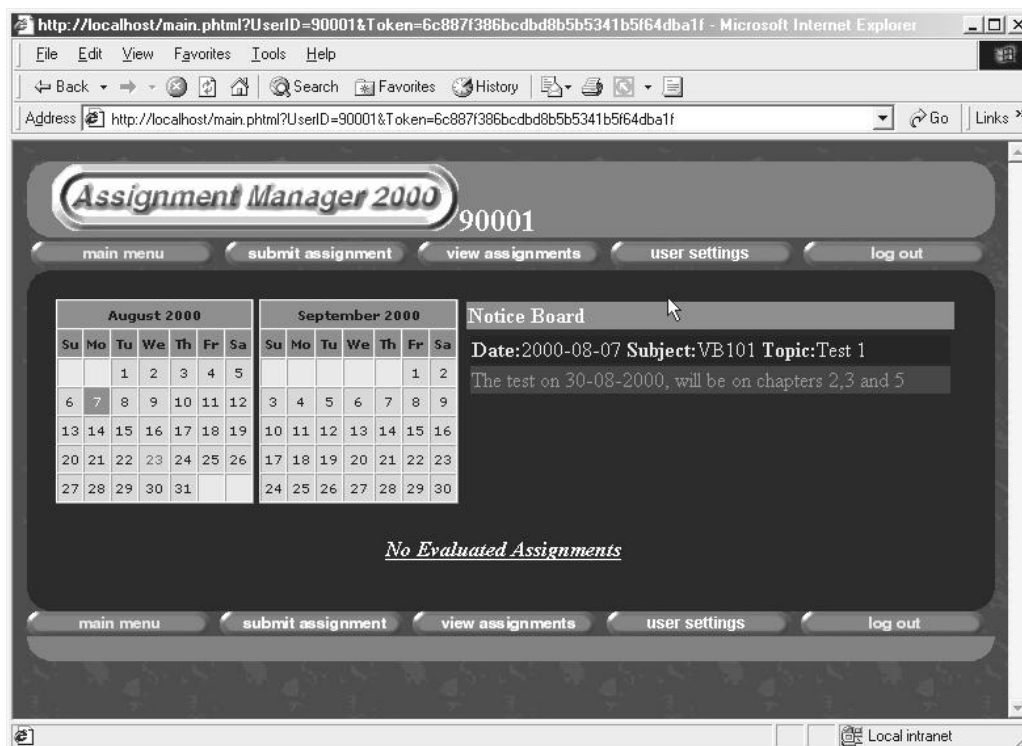


Figure B.13 The student main menu.

The students' main menu (Figure B.13.) consists of two calendars, a notice board and a list of recently evaluated assignments. The calendars display the current and forthcoming month, and indicate if assignments are due within this time period. This is achieved by colouring days on which assignments are due, differently. In addition to this, if a student moves the mouse cursor over a day when an assignment is due, a popup window appears with the details pertaining to the assignment(s). The notice

board displays all of the student's respective notices for their different subjects. The student's evaluated assignments are displayed below the calendars.

Student Assignment Submission submit assignment

Subject	Group	Assignment	Description	Due Date
<input checked="" type="radio"/> VB101	1	1a	Write a program that calculates the su...	2000-08-23

Comments

Hope you are happy with this

Attachment: **Program:**

File attachments

- D:\TEMP\Pict\frmMain.frm
- D:\TEMP\Pict\Module1.bas

Instructions: If applicable, select the appropriate Program from the **Program select** box. Then click the **Browse** button to locate the file you need, and select it. The file path will appear in the attachment field. Next, click **Attach** to attach the selected file to your message. When you send your message, the attached file is automatically enclosed

Figure B.14 Student assignment submission form.

The student submission screen (Figure B.14.) requires the student to select which assignment he or she is submitting. Once the assignment has been submitted a token is generated and sent to the users e-mail address. A token is a unique text string that is generated by a script residing on the web server. The role of the token is twofold; firstly it acts as a receipt proving to the user and the web component, that he, or she has submitted their assignment, secondly the token acts like a digital fingerprint of the submitted assignments. In the case of a system failure the token can be used to prove that the assignment was submitted and the hash values can, if need be, be compared to the resubmitted assignment to ensure that the resubmitted assignment is indeed an unmodified version.

View Assignments [view assignments](#)

A student's evaluated assignments can be accessed using two methods, via the main menu or via the *view assignments* form. If the student clicks on an evaluated assignment, the assignments details are displayed together with the evaluated assignment files. If the evaluation application was used to evaluate the assignment, HTML and Rich Text versions of the evaluated assignment are also displayed on the form. Commented regions within the HTML version are underlined and coloured differently from the rest of the text (Figure B.15). If the user moves the mouse cursor over such commented regions, detailed commentary may be displayed in an area below the main body text.

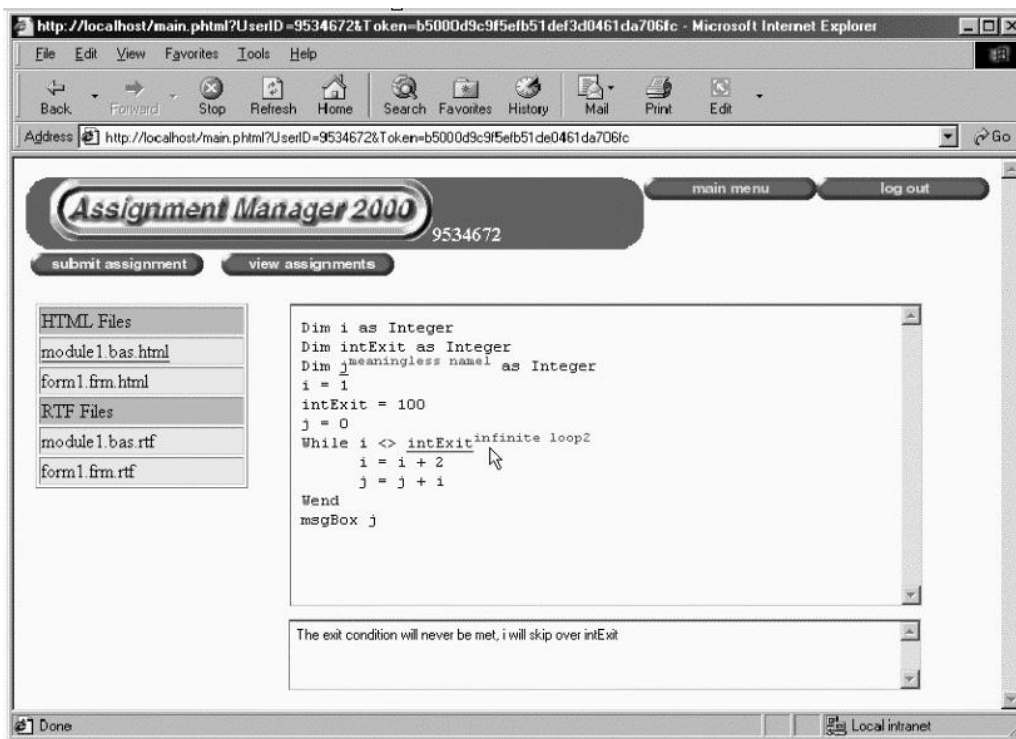


Figure B.15 HTML view of an evaluated assignment.

B.7 The Evaluation Application

B.7.1 Installation and Configuration

The evaluation application was written in Microsoft Visual Basic 6.0 SP3 and therefore requires the visual basic runtime libraries. The application consists of a single executable file, which can be placed anywhere on the user's computer. The application will automatically create import, export and save sub-folders within its folder. The import folder is used to store all of the imported assignments, while exported, evaluated assignments are stored within the export folder. The save folder is primarily used to store assignments that have not been completely evaluated.

B.7.2 Lecturers view of the application

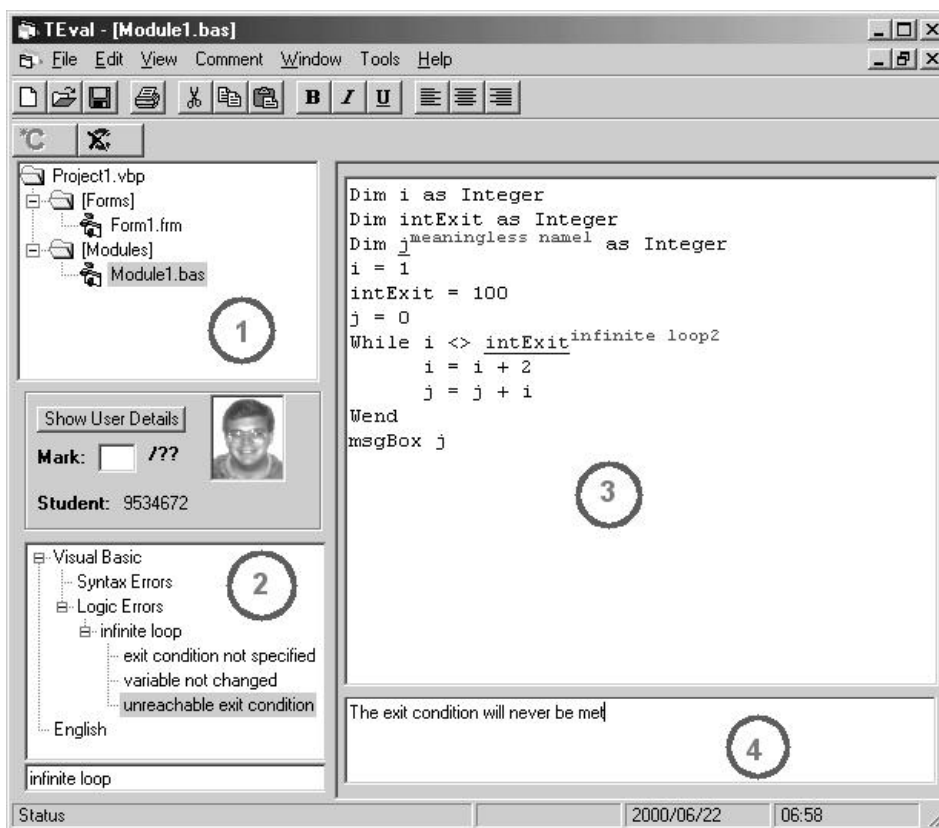


Figure B.16 The evaluation application.

The evaluation application (Figure B.16) is designed to be flexible in its use and specifically caters to the needs of Visual Basic programming lecturers. The

application operates independently from the web component, therefore other applications could be used for evaluation purposes. The [1], [2], [3] and [4] that are shown in the image were inserted to assist in the explanation of the application.


The application supports any rich text or text-based files. In addition to this, it has native support for Visual Basic files (.vbp, .cls, .bas, .frm) and can simultaneously cope with multiple visual basic projects per assignment. If the lecturer opens up a visual basic project file, all of the related files are automatically loaded and can be accessed by clicking on the appropriate file listed within the treeview [1].

The following sections will discuss the main features of the application, and how to perform basic operations within the application.

Importing Files

The import function is used to import assignments that were exported from the web component. The evaluation application can import two types of assignment files. These have been given extensions of .cxp and .svl. The .cxp files can contain entire classes of assignments, while .svl files contain individual assignments. If the lecturer imports a .cxp file, the assignments that it contains are extracted as .svl files, and stored within the import folder. The actual assignment is loaded within the evaluation application when an .svl file is imported.

Annotation

To comment a particular passage of text, the user must select the text passage in question and either left click on the add comment button , or on a predefined comment in the error-tree (shown in [2] in Figure B.16). If the add comment button was used, a form will appear asking for a brief description of the error. This text (together with a comment counter) will be displayed in superscripts adjacent to the selected text. The originally selected text is also re-coloured and underlined so as to make it easy to see (for example, for i = 1 to 10 step -1^{infinite loop¹}). Once this is done

the focus will be automatically shifted to the comment text box [4]. A detailed comment or description can be made in this box.

On the other hand, if the user clicks on a predefined error, the error description and detailed commentary is automatically filled in. This is described in more detail in the next subsection.

Error Types

It is not uncommon for students in a particular subject to make commonly recurring mistakes. For example, as in Figure B.16, many students may make mistakes which cause infinite loops. It is undesirable for the lecturer to repeat his or her description of such commonly reoccurring mistakes; a feature was therefore added to make it possible to add pre-defined commentary for commonly occurring errors. The evaluation application supports a customisable hierarchy of error types. (See [2] in Figure B.16) This enables lecturers to pre-define and categorise commonly occurring errors.

A pre-defined error has two fields associated with it. The first field gives a brief description of the error type (superscript) and is used within the body of the evaluated text (in the aforementioned example this would be “infinite loop”). The second field contains a detailed description of the error. Errors can be easily added or edited from the error tree by simply right clicking on the tree ([2] in Figure B.16), if the user right clicks on a specific error type, he or she is given the option to edit the error type or create a subcategory of that error type.

Student information

When an assignment is loaded in the evaluation application, a photo of the student is presented to the lecturer. This helps the lecturer to put a face to a name or number, making the evaluation process more personalised. Another tool that was added can be accessed by clicking on the “Show user details” button. Clicking on this button results

in the treeview [2] being populated with historical assignment details of previous assignments (Figure B.17).

These details show how often various pre-defined errors occurred in the previous assignments. For example, Figure B.17 shows that Assignments 1 and 2 contained an infinite loop, while Assignment 3 contained 2 infinite loops. If a student consistently has problems with infinite loops, for example, the teacher should be able to identify this from the historical details, and address the problem appropriately.

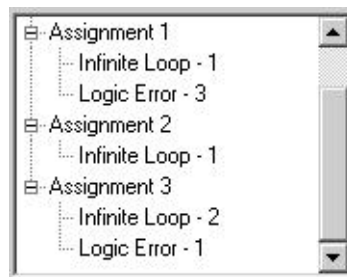



Figure B.17 Error history tree.

Editing and Deleting Comments

To delete a comment, click anywhere within the commented area and then click the remove comment button . The evaluation application only permits the user to edit the detailed commentary associated with a commented area. If the user clicks on a comments text area the detailed commentary is displayed in the comment window below. The user can then, if he or she wishes edit this text. To change the superscript text, the user must delete and create a new comment with the desired text.

Exporting Files

If the web component is used in conjunction with the evaluation application, the lecturer must use the export function to save the evaluated assignment in a format that is readable by the web component. An exported assignment is saved as a single file,

which, not only contains the evaluated assignment, but also HTML and rich text versions of it. The evaluation application also provides evaluators with a tool to combine all the evaluated assignments into a single file.

Combine Exported Assignments

This function (situated under *tools*) allows the lecturer to select a group of exported evaluated assignments (.evl) and combine them into a single file (.gvl). This function can therefore shorten the submission process as only one file needs to be submitted.

Export Error Definition List

This tools based function exports all of the defined error types within the error-tree (see [2] in Figure 15) into a single file. This file can then be imported into the web component to reflect the changes in the database.

Import Error Definition List

This option is used to import predefined errors that are retrieved from the web component.

Support

All problems or queries can be directed to Grant Pullen at grant_pullen@hotmail.com.