December 1997

Utilising the WWW for Assignment Management: The Development of a Working Prototype at Southern Cross University

Rod Bymes
Southern Cross University

Julian Dimbleby
Southern Cross University

Bruce Lo
Southern Cross University

Sue Plunkett
Southern Cross University

George Stuart
Southern Cross University

Utilising the WWW for Assignment Management: The Development of a Working Prototype at Southern Cross University

Rod Byrnes, Julian Dimbleby, Bruce Lo, Sue Plunkett, George Lee Stuart
(School of Multimedia and Information Technology)
***Peter Vitartas** (School of Management and Marketing)*

Southern Cross University
P.O. 157 Lismore 2480, NSW Australia.
Fax: (066) 221 724

rbyrnes@nor.com.au, jdimbleb@scu.edu.au, blo@scu.edu.au, splunket@scu.edu.au, gstuart@scu.edu.au, pvitarta@scu.edu.au

Executive Summary

PURPOSE of PROJECT..... To develop a WWW based assignment management system (AMS) that offered all client groups:

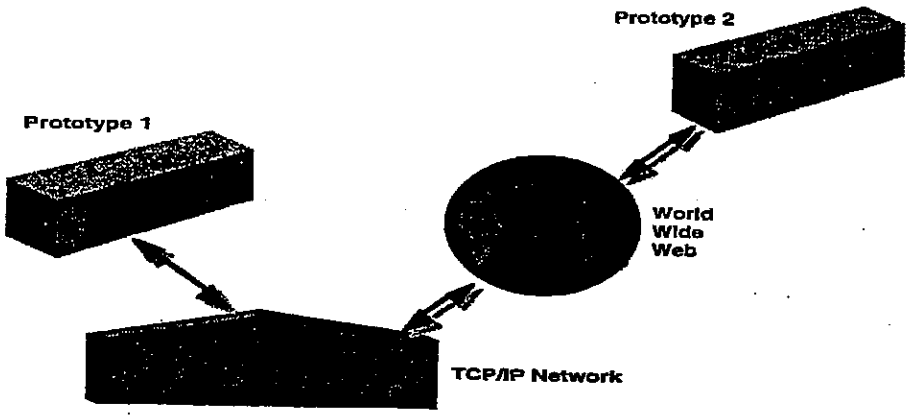
⇒ **cost effective** - The Information Technology Directorate at SCU spent $46,850 on paper for assignment printing in 1996. Any scheme that would encourage such usage to be reduced would be welcomed in the short term;
⇒ **security** - Compared to other electronic systems, a Web based AMS guarantees security for all client groups;
⇒ **fast response** - no postal and administration delays
⇒ **human resource efficient** - Aside from ongoing maintenance, a working version of a Web based system requires no staffing; and,
⇒ **a paper-less environment** - No paper used to submit assignments. No paper required to receive and view marked assignments returned from the instructor.

Students are offered:
⇒ the ability to submit up to 8 documents for each assignment;
⇒ the opportunity to submit Rich Text Format documents.

Instructors are offered:
⇒ automatic student records;
⇒ two methods of assignment marking;
⇒ the opportunity to insert marking templates into student assignments or marking 'sheets'.

PURPOSE of CONFERENCE PAPER ..... To share with colleagues our experience of two electronic prototype AMS systems and the critical issues that required resolve in developing a Web based AMS.

Diagram 1. Comparison of two prototype systems



Southern Cross University
Lismore, NSW

Abstract

The following paper describes the development of two prototypes (the first a network based model; the second a working WWW based prototype called LODGEiT) for an electronic assignment management system at Southern Cross University. It is asserted that a WWW based system is both resource efficient and less costly than conventional hard copy paper-based systems. The paper describes system development issues; describes the functionality of LODGEiT; provides a brief review of proactive and summative evaluation of the working prototype and concludes with a statement regarding future extension of the functionality of the WWW based system.

Introduction

Australian tertiary institutions, like those world-wide, have the challenge of evolving from an organisational stance purely oriented towards excellence in teaching/learning to one which embraces key business principles such as generation of funding, the paring away of inefficient work practices and costs and the development of 'leading edge' teaching/learning support processes. Such processes may include those associated with assignment management, the distribution of learning material and assessment tasks, record keeping and the dissemination of assignment feedback. Significant funds are expended by many universities in employing staff to fulfil such roles (one conservative estimate suggests $50,000 per annum, per faculty on such employ) and in developing mechanisms to facilitate the professional 'appearance' of assignment management systems (AMS).

In the main, hard copy submission of assignments remains the key practice at Southern Cross University (hereafter referred to as SCU). The process for dealing with external students submitting hard copy assignments requires the relevant AMS staff member to perform many activities which are summarised as follows:

1. key all student details into a database;
2. print off individual student address and assignment labels;
3. post labels and assignment cover sheets (with appended instructions) to students;

4. receive submitted assignments;
5. correct any errors or issues of changed addresses, incorrectly selected (by student) assignment labels, incorrect or missing labels because student had altered learning program;
6 scan labels on submitted assignments to 'check off' database record;
7. post receipt slips to students;
8. distribute assignments to lecturers;
9. receive marked assignments back from lecturers;
10. record assignment grade;
11. place assignment in envelope, print off label, post.

Although the process for on-campus students is somewhat truncated, the sequence is none-the-less time consuming. As SCU learning programs increased in popularity in the early 90's, expanding to include distance learners (for example, MBA students from Singapore), it became clear that the costs associated with hard-copy based assignment management would become prohibitive in the medium to long term. This primary concern acted as catalyst for the identification of an AMS that was both cost and time efficient. If, at the same time, that method would be capable of extending the functionality of hard copy AMS methods, then so much the better.

In order to clarify the general features necessary for the creation of a cost efficient system, a comparison was made between traditional methods of assignment submission and those which utilised a form of electronic medium such as E-mail, a 'folder' or file on a network, or a server such as the WWW acting as 'mediator' for submission into an on-line browser. Table 1 compares the selected features of differing forms of assignment submission. Although electronic systems tend to be less favourable than hard copy systems because of possible loss in integrity of the original documents due to a mismatch between instructor and student software applications, nevertheless electronic systems overall were seen as more desirable than hard copy methods because of greater cost efficiency (no staffing, postage or stationary costs), reduced paper usage (accounting for the environmental factor), security (utilisation of passwords) and the potential for automatic creation of system records (thus reducing instructor workload in record keeping).

Table 1. Comparison of Selected Features of Differing Assignment Management Systems

Feature	Hard Copy	E-mail	Folder/File on Server	WWW
Security	Available only if 'locked' boxes are provided. Usually requires staff to clear and distribute	Not secure as E-mail addresses can be 'faked'	Available if student knows how to 'lock' personal folder	Yes. Available via password
Suitability for Distance Education	Costly because of postage and staffing	Available anywhere at any time wherever computer has modem. Must have interface application.	Service not always available for off-campus students	Available anywhere at any time wherever computer has Internet access and modem
Ability to submit Rich Text Format documents	Unable to support video/music clips within documents taken from the Internet	Yes, although documents tend to lose integrity	Yes	Yes
Paperless	No	Yes	Yes	Yes
Requires AMS staffing	Yes	No	No	No
Facility for system records	Only if keyed into database by AMS staff	No	No	Yes

Early Background of an Electronic Assignment Management System
In 1994, an honours student, Rod Bymes, at SCU (in the Centre for Computing and Mathematics) developed a prototype of a network based electronic AMS. The features that were included in this first prototype are displayed in Table 2.

Table 2. Functionality Available to Instructors Using a Network Based AMS

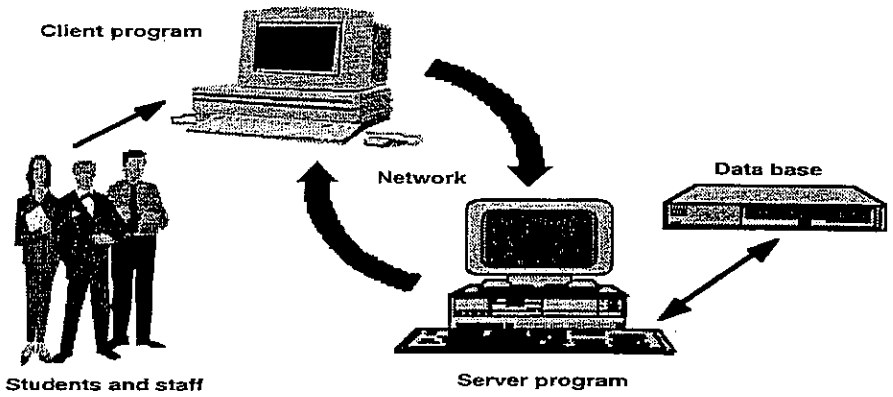
Function Name	Description
Mark assignments	The staff member selects which unit and assignment number they wish to mark. A list of students is presented, with their assignment submission status which can be: 1. unsubmitted; 2. submitted but not marked; 3. marked; or 4. marked and returned to student. When an assignment is returned to the student, the mark they receive is automatically recorded in the database.
Administration - Student Information or Password	* Instructors can change a student's password if the student forgets it. This removes some workload from the administrator. * Instructors can add, modify or delete student information
Administration - Tutorial groups	Instructors can enrol students in tutorial groups and 'transfer' them between groups if necessary
Administration - Assignments	Instructors can enter unit assignment details for their cohort

The key principles which underpinned the development of the first prototype were:

- User Friendliness. This was a priority to ensure that the system could be applied university wide. The computing literacy of students from different professional fields needed to be taken into account.
- Expandability. The system was designed to readily incorporate new features. It was anticipated that the system could eventually provide computer based testing and a facility for learner-learner interaction.
- Accessibility. There were three locations from which the system had to be accessible:
  - On campus access from computers attached to the University network;
  - Open Learning Access Centres geographically distant from the two main campuses of SCU at Lismore and Coffs Harbour;
  - Off campus access in general via the Internet.

The First Prototype: A Network Based Assignment Management System
The prototype consisted of a client program and a server program (both written specifically for the prototype in C). The server interfaced with a commercial relational database. Diagram 1 illustrates the main components of this first prototype.

Diagram 1. First prototype: a network based system.



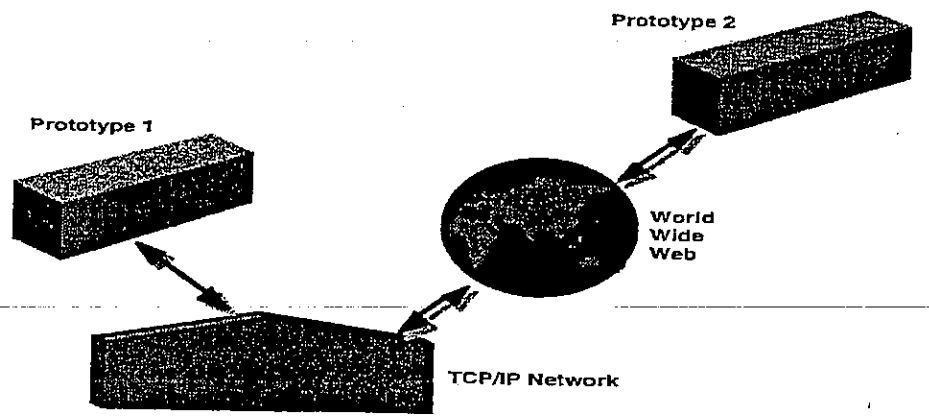
The workload involved in developing the first prototype was large. One drawback for this system was its lack of portability, ie. a client program had to be developed for each type of PC or Macintosh computer that a user may utilise to access the AMS. It was apparent that if the workload was going to be manageable, an existing multi-platform client program would need to be utilised. There were two major contenders:

1. WWW browsers, or
2. SQL clients.

The first option (Web browsers) was selected for the second prototype due to the fact that the interface was already familiar to most of the users, ie. students and staff, who would be using the AMS. (This familiarity arising as a consequence of the tremendous growth of Internet access in the last two years). Additionally, web browsers and servers were already available to the development 'team'[2] thus minimising expenditure at this point of the project.

The Second Prototype - A Web Based Assignment Management System - LODGEiT
The essential difference in schema between the first and second prototypes may be represented as in Diagram 2.

Diagram 2. Comparison of two prototype systems



It was felt desirable to give this second prototype a meaningful and distinctive name. Because the heart of the developed system is concerned with students lodging assignments, the name "LODGEiT" was chosen.

This second prototype, or LODGEiT system, provides comprehensive support for virtually all activities associated with the submission and marking of assignments. Table 3 provides a summary of the LODGEiT activities to be performed by each student, each instructor and by the system administrator. The activities in this table are listed in the approximate sequence in which they would normally be performed and are grouped together into logically related clusters in this table. The student submit assignment activities and the instructor mark assignment activities are naturally repeated many times for each unit (subject).

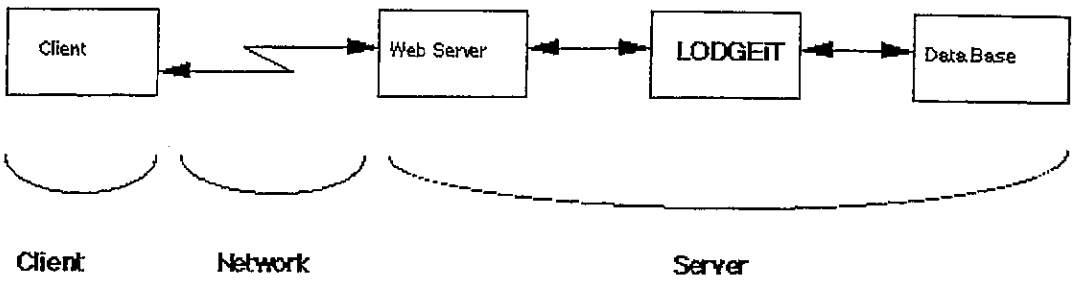
Table 3. Summary of LODGEiT Activities

Group	Assignment Management System (AMS) or LODGEiT Activity	Performed by		
		Student	Instructor	Admin
Setup	Enter instructor name and details			1
	Enter unit (subject) name and details			1
	Enter student name and details	1	1	
	Enter tutorial details for each unit		1	
	Enter tutor for each tutorial		1	
	Enrol student into a unit	1	1	
	Enrol student into a tutorial in a unit	1	1	
	Enter assignment details for each unit		1	
	For teamwork asgs, enter groups (optional)		1	
Submit	Submit Asg - 1 select unit and asg number	1		
	Submit Asg - 2 select file(s) and set file type	1		
	Submit Asg - 3 send comments to lecturer (optional)	1		
	Submit Asg - 4 submit asg file(s)	1		
	View asg (optional)	1		
Mark	Mark Asg - 1 choose unit, asg number and student		1	
	Mark Asg - 2 select each submitted file		1	
	Mark Asg - 3 enter feedback comments in original student file or separate html file		1	
	Mark Asg - 4 final mark		1	
	Release marks (when finished all asgs)		1	
	Grant extension to asg due date for one student		1	
View	View asg mark	1		
	View asg feedback comments from instructor	1		
	View class minimum, average, maximum marks	1		
Other	Resubmit assignment (optional)	1		
	Remark assignment (optional)		1	
	Export marks to spreadsheet for manipulation and to submit final grades to Administration		1	

Search, modify and delete functions exist for many of the "setup" activities which are listed in the Table of LODGEiT activities.

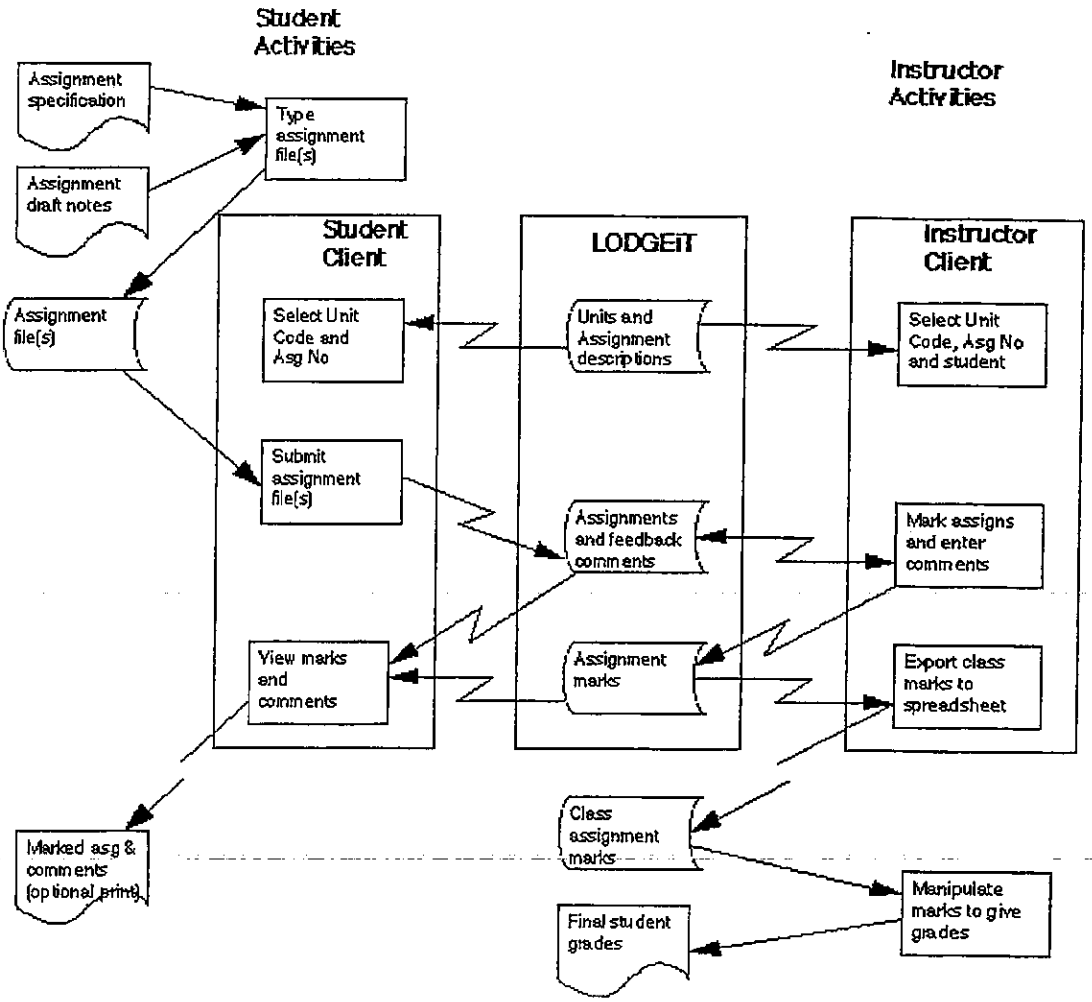
A visual representation of the main components of this Web based client / server LODGEiT system is shown in Diagram 3.

Diagram 3. Client / server components of LODGEiT



The flowchart in Diagram 4 seeks to show how a selection of some of the main activities performed by students and instructors are linked together in the client and server components of this LODGEiT system.

Diagram 4. Flowchart of student and instructor client activities



As indicated by the start point in the flowchart in Diagram 4, the specification for each assignment and the associated marking scheme is currently forwarded to each student by conventional means e.g. paper-based, email. A future enhancement to LODGEiT will incorporate this information.

LODGEiT benefits

This system provides significant time savings to students in the submission and return of assignments. This is especially true for off-campus, distance learning students where postal delays can run into many days. The system will automatically generate email messages to students when all assignments have been marked. At the same time, a statistical profile of marks (minimum, average and maximum) is available for all students in the class. This further saves time for students where their usual means of obtaining this information is direct from the instructor or by reference to a notice board.

Instructors have commented that they value the fact that the onus of responsibility for successfully submitting assignments now clearly lies with students and not with instructors (for example, with assignments previously submitted on floppy disk, frequently there have been files missing or empty files or obscure named files). Instructors have recognised the advantage of being able to perform keyword searches on individual assignments. They have also been able to speed up their provision of feedback comments by building a growing list of assignment-related points from which they may copy and paste to provide rich feedback to individual students. They may also wish to provide a sample framework assignment solution to students. In this electronic medium, the used of different sized fonts and colours and graphic images can readily be used to visually enhance the feedback provided.

A significant benefit to instructors is the retention of all student marks by the system. At any time, an instructor may export these marks to a spreadsheet for manipulation. This facility would certainly be used at the end of each semester to produce final student grades to submit to Administration. It is Iso worth noting that, as well as retaining all student marks, the system retains all feedback comments. Instructors may find it valuable to build up a history of comments for future reference. For example, if a particular student is experiencing difficulty with a current assignment, it may prove beneficial to quickly glance at their earlier submitted assignments and review the feedback comments provided. In this way, a student's individual performance may be closely followed and encouraged.

Another benefit provided by LODGEiT is that more than one instructor, possibly in different geographical locations, may review the work of one student at the same time. This may be beneficial in cases where, for example a second opinion is sought on a 'border-line" assignment. LODGEiT also allows assignments to be remarked. For example, where it transpires that an assignment specification may be ambiguous in parts, allowance can be made for this by subsequently remarking previously marked work (and also providing additional feedback comment to this effect). On a related matter, LODGEiT looks after the bureaucracy associated with handling assignments which are submitted after the due date. Instructors may choose to set a penalty for each day overdue.

Some instructors have expressed discomfort at marking many student assignments on-screen. Certainly the facility to print each assignment exists, but this approach would be 'at odds"with one of the objectives for this paper-less system. Other instructors however, have said how they enjoyed marking assignments on their computer. This issue appears to be quite a personal matter, and there may be a transition period for some instructors to move through in order to adjust to this new way of handling assignments.

LODGEiT size and screens

To provide a rough measure of the size of the current LODGEiT system, there exists a total of approximately 60 different screen formats. This is made up of 11 administration screens, 35 instructor screens and 14 student screens. Although to a new user this size may seem formidable, practiced LODGEiT users have found the system to be very intuitive. A small sample of the content and appearance of these LODGEiT screens is provided in the Diagrams 5 and 6 for student assignment submission, and in Diagram 7 for instructor mark assignment..

Diagram 5. Student submit assignment screen - document name(s) and type(s)

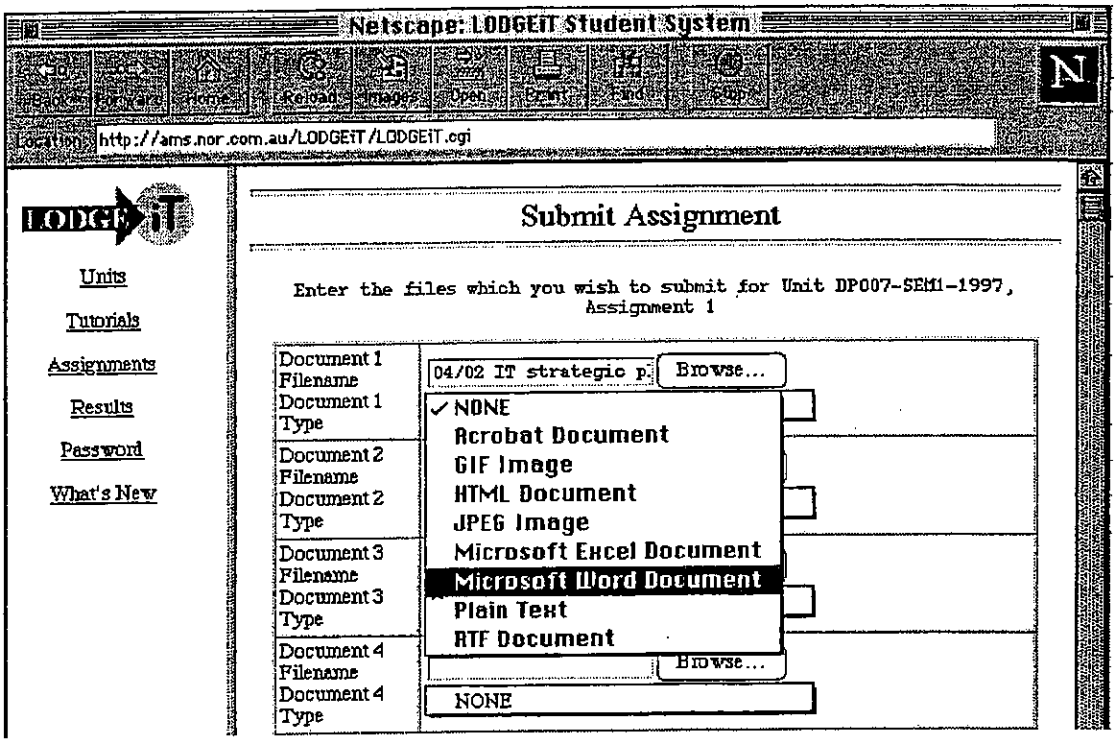
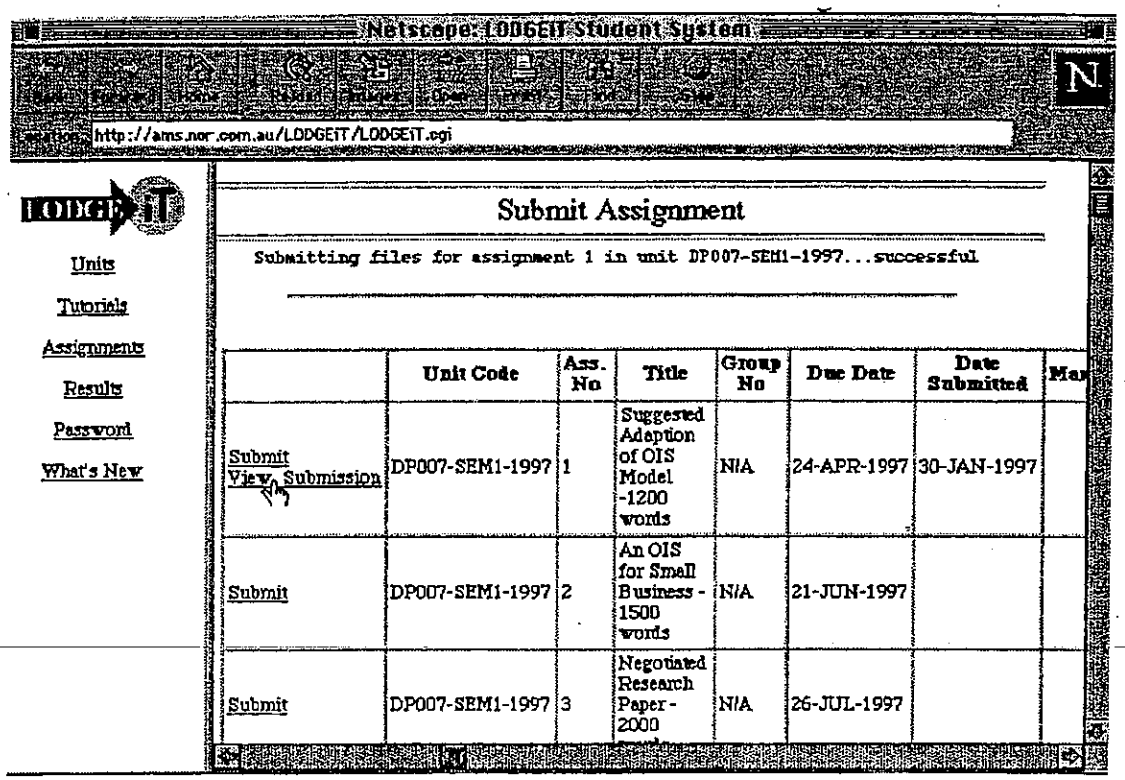
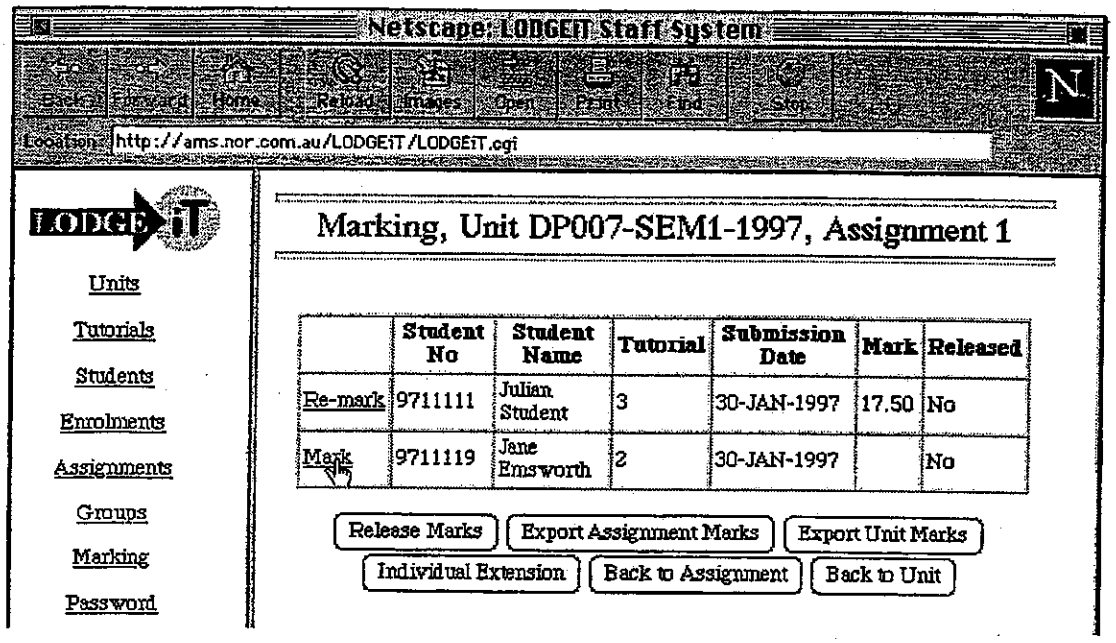


Diagram 6. Student submit assignment screen - overview of assignment details



	Unit Code	Ass. No	Title	Group No	Due Date	Date Submitted	Mar
Submit View Submission	DP007-SEM1-1997	1	Suggested Adaption of OIS Model -1200 words	N/A	24-APR-1997	30-JAN-1997	
Submit	DP007-SEM1-1997	2	An OIS for Small Business - 1500 words	N/A	21-JUN-1997		
Submit	DP007-SEM1-1997	3	Negotiated Research Paper - 2000	N/A	26-JUL-1997		

Diagram 7. Instructor mark assignment screen



With the introduction of the WWW as a critical element within the second prototype, a number of issues pertaining to the use of Web based applications must be acknowledged and accommodated within the system design. These are reviewed in the following section.

Web Based Applications

The major problem with web based applications is that the web is 'stateless', ie. after performing one operation from the browser (for example, looking up the details of a student) the server promptly forgets everything about the transaction. So, if the instructor then wanted to modify the student's details, the browser must tell the web server everything it needs to know to perform the transaction. This does not just include the new details (as would be the case in a system that maintained state) but also:

1. the person performing the operation (for reasons of security); and,
2. the operation to be performed (eg. modification of student details).

This state information must be passed from the client to the server for every transaction. The second prototype implemented this by using both fields in a form and the query string parameter in the URL itself.

It must be noted that Netscape (v.2) (the browser utilised at SCU) was the first version to support file uploads, an essential function of the second prototype. Netscape 3.0 and 3.0 Gold appear to have some implementation problems which make these versions unable to be utilised by the second prototype at this time.

Database Changes and Allied Programming 'Status'
The first step taken to improve the first prototype was to re-design the database. Several problems that were discovered in the first prototype required modification. For example:

1. The first prototype was unable to store more than one document per submission. The second prototype allows up to 8 documents, of any type, to be submitted for each assignment;

2. There was a need to introduce extra fields and tables into the second prototype in order to allow:
- multiple tutorial groups with separately assigned tutors who can mark only their own students,
- group or teamwork assignment submission;

3. There was a need to incorporate "blob" fields to store entire documents within the database itself rather than referencing them externally within a filename; and,

4. In the first prototype the database integrity was not properly enforced. The use of triggers and other DBMS (database management system) facilities were previously not used adequately.

The programming effort for the second prototype operated at a much higher conceptual level than for the first prototype. This resulted in faster development times and reduced system maintenance (to both correct perceived flaws or shortfalls in the system, and to effect modifications).

Program development, for this second prototype, was achieved by an application of iterated prototyping driven by functional specifications.

Initially a function list was provided and a set of screen designs worked out by the project team in group sessions. After an initial prototype was completed, subsequent prototypes were created on the basis of additional functional requirements until the system "went live" in its first useable state by instructors and students.

Thereafter system changes were implemented on the basis of a Request For Change (RFC) system where bug fixes and functional enhancements were implemented. RFC's were logged and prioritised. Some RFC's were considered essential, bug fixes for example, and others were less important enhancements.

The Parse Library
The parse library in essence 'sits' between the http server and the LODGEiT program. The library takes the data from the http server (which the browser sent as a result of the user's request) and puts it into a form more easily accessible to the LODGEiT program. For example, data might come from the instructor in a form such as this:
username=staff
function=modify-student
student_no=123456, (and so on).

The parse library interprets this and makes it available to the LODGEiT program in the form of "easy to use" function cells. So, for example, to find which function is being asked for, the LODGEiT program will simply do: function=get_variable("function")

The 'library' also ensures that if the LODGEiT program asks for something which was not sent from the client, a valid result is returned, hence, a lot of error checking is done automatically, and the programming effort is reduced. Finally, the parse library also parses uploaded files which are in a more complex format than the above.

System Hardware
The second prototype server was a 486 DX2 66 running SCO Unix, the NCSA http server and Interbase 3. In order to increase server speed and to cope with an increased load of student assignments, a Pentium system with a 2gb hard drive is intended for purchase in 1997.

A Critique of the Second Prototype - A Word From Instructors
Throughout the trial period of the second working prototype (approximately three months) a database was developed to record all queries, problems and requested changes (for the system) posed or forwarded by the five instructors utilising the system.

This 'database' of change was regularly reviewed by the development team and suggestions for system 'amendment' awarded a priority in consultation with the systems administrator. A time-line in which changes would be made operative was developed to ensure system amendments would be completed by the commencement of First Trimester 1997.
In an early version of this second prototype, some instructors expressed concern about the "setup"time required to enter all students onto the system. This issue was addressed in one of the enhancements to LODGEIT by allowing students to enrol themselves into each unit of study. Instructors can still perform this task in the small percentage of situations where student difficulties arise, but they are now relieved of the burden of entering this information for all students.

A Critique of the Second Prototype - A Word From Students
Prior to the second working prototype being 'installed' for assignment submission, students (n=103) were asked to rank, in order of preference, the assignment submission methods they preferred. Twelve of these students came from the Coffs Harbour Campus, the remainder from the Lismore Campus. Whilst 87% of students were undertaking a Bachelor's Degree in Computing or Information Technology, the remaining 13% were undertaking various computing and/or marketing units as electives within their base program of Paralegal Studies, Human Movement or a general Bachelor of Arts Degree.

When asked to rank selected assignment lodgement methods from most preferred to least preferred the survey sample provided the following sequence clearly indicating a preference for a secure method of submission (although the cost involved in postage seemed to outweigh this factor when it came to a choice of postage or leaving an assignment in an open box accessible to anyone):

Most Preferred
1. Place assignment in locked assignment box
2. Hand assignment to instructor at office
3. Hand assignment to instructor in class
4. Place assignment under instructor's door
5. Submit assignment via E-mail
6. Place assignment in open box outside instructors door

Least Preferred 7. Deliver assignment by post or courier.

With reference to perceived advantages of the second prototype, the opportunity to submit richly formatted documents, the functionality of off-campus submission and the return of marked assignments via electronic means were the salient features considered most important to the survey cohort. Though pleased with this system, a number of students expressed concern about having to separately print their instructor's comments. This prompted a further revision to the system which allowed instructor's to choose whether to directly annotate a student's original work (in a different font size and colour), or to give their feedback comments in a separate html document.

At the time of writing, a post-test has been distributed amongst the survey cohort to determine whether the working prototype met student expectations and whether students perceived they had difficulties with any aspect of assignment submission using this medium.

A Critique of Both Prototypes - A Word from the Developer

It is evident from experience with both prototypes that neither addresses all instructor/client needs. The first prototype was too much of a burden in programming effort whereas the second prototype fell short of expectations in terms of the quality of the user interface. As an example of the second point, when uploading a file, the client is offered no progress information by the browser.

A possible method of solving both these problems is the use of Java applets and Javascript. This approach would allow the development of a user interface that was as good as a specific client program, but with far less programming effort. The programming effort however would be higher that needed for the second prototype.

Future Visions

A proposed enhancement to this LODGEiT prototype is to incorporate links to skill or criterion referenced based multiple choice testing SCU browsers with a facility to provide students with automatic feedback regarding their level of competency on any given test.

Finally, the development team is keen to interest other learning institutions in trialing and thereafter utilising the Web based system to afford the said institution the same cost and resource efficient benefits that have been gained at SCU.

References

Byrnes, R. 'Computerised assignment management for distance students" Proceedings of Access Through Open Learning Conference, 1994, pp. 42-45

Byrnes, R. and Lo, B.W.N. 'A computer-aided assignment management system: Improving the teaching-learning feedback cycle" Proceeding of the 5th International Conference on Computer Managed Learning, 1994

Byrnes, R., Lo, B. and Dimbleby, J. 'Flexible assignment submission in distance learning"Proceedings of the IFIP World Conference for Computers in Education, 1995, pp. 305-315.

Hagan, R. and Trathen, C. 'Multimedia E-mail for computer mediated education materials" Proceedings of ASCILITE, 1993, pp. 316-323

Hasan, H.M. "The paperless classroom" Proceedings of ASCILITE, 1991,
pp. 267-276.

Kaye, A. 'Computer-mediated communication and distance education"in Mindweave (eds. Mason, R. and Kaye, A.), Pergamon Press, Oxford, 1989, pp. 439-449.

Ross, R. 'Structure and functions: Implications of access" Open Learning and Technology, in Occasional Papers in Open Learning, (ed. Ellis, A.) 1, 1991,
pp. 18-25.

Scriven, B. 'Perceptions of quality in distance education"Proceedings of the ASPESA Biennial Forum, N.S.W: Charles Sturt University, Bathurst, 1991, pp. 439-449