Using Collaborative Technology in CS Education to facilitate Cross-Site Software Development

Marie DEVLIN
School of Computing Science
Newcastle University, UK

and

Sarah DRUMMOND
Dept. of Computer Science
Durham University, UK

and

Andrew HATCH
Dept. of Computer Science
Durham University, UK

ABSTRACT

With offshore computing becoming more prevalent, it is essential that we increase our students' employability by providing new and relevant experiences in software development and project management; giving them valuable skills that are essential in an ever-increasing and changing global market. What is new about the work we discuss here is how collaborative technologies have facilitated a year-long cross-site software engineering project between Durham University and Newcastle University students. Our use of various collaboration technologies such as online discussion forums, video-conferencing, company repositories, version control software etc., as part of the collaborative team project has not only encouraged students to develop technical 'transferable' skills but also gain an understanding, through realistic experiences, of how the use of these technologies involves more than just learning their technical aspects and operation, but that it is essential to develop and implement the soft processes and skills required to use them successfully and effectively and hence optimize their cross-site working partnerships and productivity. In this paper we describe the project, the technologies employed by the student teams and the results and anecdotal evidence of staff and students that show the successes and, it must be admitted, occasional failures of this work. We discuss how we have tried to manage the expectations of the students throughout the project, how the technologies we have provided have affected the students' experience of cross-site collaboration and the impact of cross-site collaboration on our assessment strategies and curriculum design.

Keywords Cross-site software development, software engineering student project, innovative curriculum, collaboration technologies.

1. INTRODUCTION

Increasingly in the software industry cross-site development is becoming commonplace [1]. Employers are seeking graduate employees with more than just the technical, analytical and problem-solving skills that are already embedded in the Computer Science (CS) curriculum. In order to compete more effectively in the job market our graduates now need to be inventive and creative and have experience working as part of a team for both co-located and cross-site software development projects.

While most ICS departments provide students with experience of team working, the opportunity for them to adopt cross-site collaboration is rarely taken. Such an undertaking is often seen as being prohibitive, with issues such as assessment, finding a 'window of opportunity' in the curriculum and cohort size being especially problematic. However this type of experiential learning will equip the students with the skills that industry and business now require.

Active Learning in Computing (ALiC) is a Centre for Excellence in Teaching and Learning, (CETL), project funded by the Higher Education Funding Council for England. It is a collaborative effort between four partner institutions: Newcastle University, Durham University as CETL lead, Leeds Metropolitan University and The University of Leeds. Together these institutions provide a broad representation of the student population and the variety of curricula available in Computing Science higher education in the UK today [2].

ALiC have now extended the traditional CS group-work project to include inter-institutional collaboration. Teams are formed from students at Newcastle University and Durham University (geographically separated by 18 miles). These teams collaborate over the course of an academic year in order to produce a software product. This activity mimics cross-site development processes used in the software industry where many companies face the challenge of collaboration across different sites.

2. PROCESS

The aim of the cross-site activity is to align group-work activities in higher education CS to students' future work-based practices. The activity simulates this working practice via a shared Software Engineering assignment between 24 teams of second year students, 12 at Durham and 12 at Newcastle. The cohorts of students are enrolled on Computer Science, Information Systems or Natural Science programmes. Some of the pedagogical aims of this cross-site collaboration activity are to:

- Give students an insight into Software Engineering in an industrial context;
- Make problem-solving more realistic in student team projects;
- Allow staff and students to use and evaluate various technologies for cooperative working ;
- Encourage the development of transferable skills such as communication, organising and team-working.

During this activity 12 'companies' are formed, each comprising a team from each institution. The companies have to work together across the sites in order to manage their project and develop a product together at the end of the academic year. The activity has now run for two academic years and is still ongoing. We are gathering information and student views about the activity as well as assessing projects for feasibility. We are also reviewing and developing suitable assessment mechanisms and tools for ensuring fairness, evaluating the learning outcomes at different stages of the process as well as reviewing the technologies that support learning in this context.

3. THE ASSIGNMENT

In the academic year 2006-2007, the 'companies' were asked to develop a software system for running enthusiasts that had the ability to monitor the user's running performance over distance and time. The system was to be capable of collecting training data via a GPS unit which would allow the user to tailor a training programme. Each company was provided with two DELL AXIM PDAs and two Global Sat Bluetooth GPS receivers - one set of hardware for each local team.
The companies were required to develop two parts to the application 1) a desktop application with backend database to manage the running data and 2) a PDA application with a map or graphing facility. Mandatory requirements of these applications included the ability to gather statistics, plot runs, construct training schedules, record runs (track logs) etc.
There were also some mandatory elements of the specification that had to be developed at a particular site i.e. Durham had to develop the database component of the software to fulfil the requirements for their database module which is synoptically assessed in conjunction with the Software Engineering module.
To ensure that there was a fair division of the workload it was made compulsory for Newcastle to develop the map components for the application. The sharing of the workload for development of the rest of the components was to be decided by the companies themselves. Figure 1 shows the typical components of the systems that the students had to develop. Ideally their desktop application hosted the database for storing running information and the map component. The PDA application had to parse the GPS data from the Bluetooth GPS receiver and record data such as altitude, speed, distance etc. This information would then be transferred to the desktop application via XML.
One of the compulsory components that had to be developed by the companies was a "Ghost Runner" function that enabled the runner to run against a pre-loaded track log. The Ghost Runner could be an ideal personal performance that the runner was aiming to reach or previously recorded by someone else (their favourite athlete or a competitor) who they wanted to beat over the same terrain.

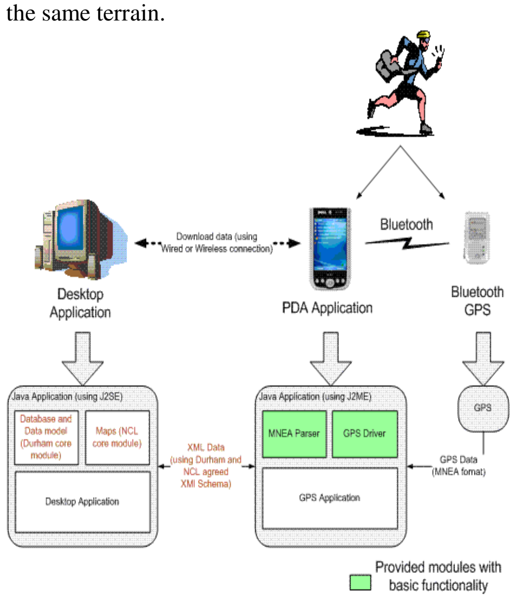

Figure 1: System overview and architectural components

4. FACILITATING COLLABORATION

Supporting the students in this endeavour has involved the wide scale use of technology with internet and web based technologies being key. The design of our institutional support technology is influenced by industrial software engineering practice. "In a virtual learning environment, one is often deprived of direct human interaction, but there are many ways to enrich learning processes through interactive systems, which provide a human-centred component in technology based environments." [3]
We have tried to support and emulate the human interaction that is so important between teams by the use of video conferencing technology. The set up for video conferencing at each site is quite different. Newcastle has a sophisticated video conferencing suite whereas at Durham they have a much simpler microphone and web cam set up. Durham support local team interaction via providing interactive technologies in their Techno-café booths, an example of which can be seen in Figure 2. These technologies include two interactive white boards, an interactive plasma display, two projection systems, a SMART tablet, laptops and tablet PCs. Newcastle also supports local team interaction in their 'web-cam' room. The web-cam room provides laptops, webcams, headsets and microphones for one to one video messaging, two large display screens that are linked to the video conference room next door for larger group participation, height-adjustable tables and soft 'social' seating for co-located team working. Traditional PC labs offer poor

environments for group work due to their layout and lack of relaxed working environment.
The situation is improved by having a relaxed environment designed for creativity and communication. Therefore, the Techno-café allows students to work together in a comfortable environment supported by flexible and time-saving technologies.

4.1 Collaborative Development Technologies

Eclipse was the preferred development environment for the assignment. Eclipse is an open source powerful Java IDE. It was primarily developed by OTI (Object Technology International Inc.), an IBM company. The choice of Eclipse at Newcastle was somewhat controversial as students had become accustomed to using IntelliJ and also to using the Linux operating system. Eclipse also provided the simplest means of facilitating version control for the companies [7].
The companies used Java 2 Micro Edition (J2ME) which is Sun's version of Java aimed at machines with limited hardware resources such as PDAs, cell phones, and other consumer electronic and embedded devices.

Figure 2: Techno café at Durham: Students working in a booth

The companies were provided with sample software to get them started on the project. This software included a MIDlet which is a Java program for embedded devices, more specifically the J2ME virtual machine for use on the PDA. Also provided was a sample project - the MobileGPSDemo. This is a project for the Eclipse 3.2 IDE. The project was designed to allow the students to write software for the Dell Axim X51 PDA coupled with the GlobalSat bluetooth GPS receiver. The Dell Axim needed to have IBM J9 runtime environment installed. Students were also provided with XML examples and two modules for the PDA - the MNEA Parser and GPS Driver. Another version of the demo application was written in order to allow for emulation in Linux and was necessary because not all of the programming techniques or technologies used for the assignment are part of the normal programming curriculum at either university. This gave the companies a starting point to being programming their applications Students had to research these technologies in the domain analysis phase of the assignment and learn how to develop software using them throughout the implementation phase with some limited assistance from staff.

4.2 Central Repositories

Deliverables from the companies centre around documents, source code and executable software, so the main supporting technologies that were provided are NESS (Newcastle E-learning Support System, Figure 3) and a Subversion repository. NESS is a web-based e-Learning system developed by Computer Science at Newcastle that has been in use for a number of years and which allows students to submit coursework, view results, receive feedback from their tutor etc. NESS also hosted forums and FAQ section.

Figure 3: NESS - Newcastle E-Learning Support System - Team Repositories

The cross-site team project requirements meant that it was necessary to provide new features specifically put in place to support both sites in this cross-site development activity. The changes to NESS were company forums where students could discuss developments, arrange meetings, post web links etc. and company repositories for sharing documents and data. A FAQ section was provided so that staff and sometimes students could post detailed answers to questions. It was important that students at each site received the same answer to various questions posed. NESS also provided a management interface to Subversion [7]. Subversion is an open-source revision control system, allowing students to share their code and impose version control. Some companies did however augment the technology provision with their own solutions. These solutions include bulletin boards and online discussion forums. Other technologies provided and supported included MSN, GOOGLETalk, and SKYPE [5]. Some teams also relied on regular face-to-face meetings and used SMS or mobile phone calls to communicate. These were not provided or supported by us.

4.3 Cross-Site Communication Technologies

We use Access Grid software to facilitate video conferencing. Access Grid provides multimedia capability that allows the interconnection of a high number of geographically distributed groups that can videoconference and display shared documents at the same time [3]. The students can also share applications such as PowerPoint or edit code online.
Shibboleth was used for access to NESS. Shibboleth is an architecture that enables organisations to build single sign-on environments that allow users to access web-based resources using a single login. Shibboleth uses open standards (such as SAML) and was developed by the Internet2 middleware group. Shibboleth was mainly used in this case to authenticate the Durham students' remote login to NESS in order to use and share the repositories, forums etc. [4]

5. ASSESSMENT METHODS

In any group activity, assessment of both the group and the individual can be problematic. This has been addressed in a

number of ways in previous work [9], [10]. In addition to the known problems of group assessment, it is imperative in this cross-site collaboration that each institution would be performing its own assessment. It was necessary to do this as Durham and Newcastle have different ways of assessing their module. The module at Durham is worth 40 credits whereas at Newcastle it is worth 20.
Companies had to define an organisational structure, choose a software design methodology, plan the software design, estimate the amount of effort needed, consider the schedule for implementation to meet deadlines and project milestones and allocate the work. Companies also needed to plan for software integration, testing and the implementation and demonstration of their product. Throughout the whole process the companies needed to manage the planning and allocation of documentation and report writing.

The assessment of these company deliverables presented a challenge as staff had to firstly agree what was to be delivered, which often meant a change from their own current format, and secondly the subsequent agreement on marking criteria. This in itself is often difficult to agree 'in-house' without having another institution involved. What was necessary to ensure was that we had agreed comprehensive marking criteria coverage, that the individual and team efforts at each site were acknowledged and rewarded appropriately, and that a team's assessment would not be compromised by a poorly performing team in the other University. It was necessary to reassure students that poor collaboration would not necessarily be detrimental to them.

Our initial work in 2005-2006 was to assess the feasibility and benefits of cross-institution software collaboration. In this pilot study we coupled the assignments very loosely. The product was in two clearly defined sections - Durham developed an application for a mobile phone and Newcastle developed a similar application for deployment on a PDA. The teams that made up each company were working to the same scenario but the deliverables were to be implemented using different IDEs and development technologies.

The collaborative element of the work was based on the remit that the final systems had the same basic functionality and a similar 'look and feel' to the interface. Each team was asked to document and build prototype software systems to be used on a PDA or a mobile phone. One of the outcomes of this loose coupling was that there was very little motivation for collaboration [9] because the deliverables and schedules were quite separate and hence a team could effectively undertake little communication with the other site and still effectively pass the module. This lack of motivation to collaborative brought about major changes to the academic year 2006-2007 assignment. The assignment was tightly coupled where all deliverables would result from a combined effort.

It was necessary to assess their success as a company based on what they produced. However now there was team, individual and company assessment making it more difficult to assign credit to individuals for their contributions and also ensuring teams did not suffer if their collaboration was not successful.

5.1 Student Involvement in Assessment

To aid in the individual assessment process students at both sites undertook self and peer assessment (only within their own teams). Self and peer assessment is a valuable skill that students needs to be able to do especially in the development of their own judgement skills.

Sections	Joe	Kirill
1.0 Introduction	Newcastle	Newcastle
1.1 Purpose	CMR	R
2.1.1 PC Modules	CM	RM
2.1.2 PDA Modules	Newcastle	Newcastle
3.1.1 PC Modules	CMR	CMR
3.1.2 PDA Modules	Newcastle	Newcastle
3.2 Inter-process deps.	CMR	R

Key C = Create M = Modify R = Review

Table 1: An abridged sample contribution matrix from Durham

At Newcastle team members were asked to distribute 100 marks between their team members. At Durham team members where asked to rank themselves and other team members contribution on a scale of 1 – 15 (1-5 being a good contribution [11]. This ranking allowed Durham students to specify other tasks in the development of the deliverable e.g. a managerial or communication role that they were involved in from a team and subsequently a company perspective. Each team also completed a contribution matrix (Table 1) that was to be submitted with each company deliverable. This contribution matrix provided the students at each site the opportunity to illustrate precisely who was responsible for the various sections of a deliverable. Each section identified who was the creator, modifier or reviewer. Once each team understood the rationale behind completing a matrix for each deliverable they were happy to complete these as it clearly shows site contributions overall.

6. RESULTS AND EXPERIENCES

At the end of the academic year we conducted module questionnaires in order to evaluate our students' experiences during the project. This is standard practice in most UK universities. We also anonymised student individual and company reports and combined the student feedback from these and the questionnaire results together in order to find out about our students' experiences of the cross-site team project and determine if our changes to the curriculum had given the students a useful experience. We also included staff comments and observations in our evaluation.

6.1 Technology Challenges

Staff noted that the amount of time it took to set up authentication to NESS for Durham students and staff was underestimated. This resulted in a delay of 5 weeks in setting up the company forums and repositories. It was necessary to change the logging-in system for all staff at Newcastle and provide an institutional interface for authentication for Durham. The main concerns were security. Some of the problems were related to us not realising the security implications early on in the planning stages for the module. Both universities rely on their Information Service for network security and user authentication and we had to get their cooperation and advice on connecting Durham students and Newcastle students together in NESS which is a secure, virtual space. The time delay did not prevent the students from collaborating and setting up their own forums but the adverse affect of this was that it made the students less interested in moving to the NESS system when it became available.

Introducing hardware components to the assignment was a new undertaking by each site. More often in Computer Science students simulate the hardware and test their applications on a simulator. Staff noted that when presenting the students with the PDAs and GPS receivers the assignment was viewed as difficult from the outset because this is not what the students were used to. Students did however quickly begin to enjoy working with the hardware as it gave the whole assignment a greater air of realism.

Some of the technical problems the students had with the software libraries etc. were resolved mostly by posts to the FAQ section of NESS, support from our computing officers if there was a problem with accessing repositories, and the use of public key/private keys. This was a learning curve for both staff and students. Occasional 'failures' in the technologies and the experimentation with many different tools to communicate provided the students with the experience of determining where the technologies best facilitated and supported the cross-site software engineering process. Intermittent failures of the video conferencing (VC) system encouraged the students to employ contingency plans in the event of technology failures. Students also began to recognise that were certain stages of the process e.g. integration, where the benefits of face to face communication outweighed the use of technology, "The video conferences are very convenient because we did not have to travel every time we needed to work together. But there are times that it is essential to work together (during Integration of the applications) and we did not get to do that." This student is making the distinction between VC and face-to-face.

Collaboration Method / Technology	Main Method	Second	Tried by
Video Conferencing	7	3	12
NESS forums	0	0	6
Instant Messaging	1	3	6
Phone/Text	0	0	3
Email	4	6	12
Skype	0	0	4
Google Talk	0	0	7
Face to Face	0	0	6
NESS Repositories	12	12	12

Table 2: Collaboration technologies used for cross-site working during the project.

As can be seen from the results in Table 2, all the companies tried video conferencing and used email to work together during the project. Seven companies opted to use video conferencing as the main technology to work together between sites and these companies held video conferences once or twice each week, on average. All the companies used the NESS repositories to share documents and source code when they became available but their uptake of these was slow at the beginning.

It is hoped that the students have realised that effectively functioning technology cannot compensate for poor communication skills and that planning meetings properly be they virtual or face to face can avoid misunderstanding and stress. During VC meetings some students were reluctant to speak and others dominated – the outcome of this was that some failed to see the value of attending a video conference. One student commented "I attended 2/3 of the video conferences. I didn't think they were particularly helpful because I'm pretty sure both teams were trying hard not to offend the other team and perhaps didn't say everything they wanted to. I know for instance that I wanted to ask the Durham team what they had done and when could we see it… but refrained since they were sounding particularly stressed in the emails."

Students had been given training on how to conduct meetings e.g. preparing agendas and taking minutes. The importance of a clearly structured agenda to maximise the time they had during a VC was stressed to them. The problem raised above could have been avoided if an agenda item has been on progress reporting. However on reading final reports some companies failed to construct and circulate any formal agenda.

6.2 Process Challenges

Whilst the stronger coupling of the assignment this year has given the students greater impetus to collaborate it has also brought with it problems for both staff and students.

With the difference in module size at each site (essentially Durham is a double–credit module and Newcastle single) the number of timetabled lecture and practical sessions is significantly different between sites and therefore the depth of Software Engineering material covered in the module varies at each site. In addition to this there is a difference in emphasis on SE topics. Whilst it is difficult (or even unlikely) that this will change, a closer alignment of the syllabus is required to manage the expectations of the quality and quantity of work that is expected from the students. Students constantly worried about the difference in workloads i.e. as Durham SE is a double module they wondered if they should do twice the amount of work even through there were less of them in the team than the Newcastle team? Getting the balance right to cater for the differences between sites is something that we are still working on. We do not want to compromise the nature of the module at each site but we recognise that a closer alignment will prove more reassuring and perhaps easier to manage for students than at present.

In a similar vein, at Newcastle the teams were mainly made up of CS and Information Systems (IS) students and generally the IS students did all the documentation whilst CS students tended to do the more technical work. We feel it is important that whilst students are encouraged to work to their skills this does not preclude them from improving skills they view as weak. We have tried to overcome this tendency of students to divide the work this way (largely based on ideas about which degree program is better placed to do which task) with various exercises throughout the year that encourage students to recognise their strengths and weaknesses and to work on their weaker areas. However, without becoming too prescriptive this is a difficult problem to overcome and we are still working on assignment examples, case-studies and assessment strategies that could make it easier for students to do this.

During the year it was necessary to compromise on the number of deliverables for the assignment. Durham increased their number whilst Newcastle reduced theirs. However, it is still felt that the number of deliverables can be combined and reduced in size. Whilst assessment is vital it was felt by staff that the assessment was overshadowing the student learning experience. This problem is more difficult for Newcastle as this module is based solely on coursework assessment and therefore there has to be a frequent level of assessment throughout the whole year for both the individual and the team. Durham's assessment division is based on 60% coursework (with 25% of that being an individual reflective report) and 40% unseen written exam.

One of the larger deliverables during the project is a project plan. Each company must produce a Gantt chart indicating all relevant tasks and milestones. For each task the person/people responsible for ensuring that the task is completed on time and those individuals who will be engaged on that task are identified. This is completed but very rarely adhered to mainly because the emphasis on planning throughout the project is lessened as the date of 'deliverables' is laid out explicitly in the assignment. In future versions of the cross-site work staff agree that greater emphasis must be placed on the creation and deployment of the plan to alleviate problems students encountered. Whilst it is recognised that this is the students' first attempt at a project plan more thought needs to go into it. Students totally underestimated the time they allocated for implementation and integration of the system under construction.

7. FUTURE WORK

The project will continue in the next academic year and follow a similar structure to that reported here. However we are introducing a "real" customer. A large international company are proposing a real world problem in supply chain logistics and are prepared to meet the students on a number of face-to-face events and answer typical questions they would receive after putting out to tender. It is envisaged that this will provide the students with experience of producing documentation and a prototype system that is acceptable to a real customer. The technology infrastructure that is now in place will be used again. This time therefore the technology will be more stable and available from the onset of the project. The student uptake will depend on how we introduce the technologies and highlight their potential.

The students will be encouraged to investigate and use other communication and collaboration technologies they feel are more convenient and useful to them with e.g. MSN, mobile phones and email. Skype conference internet phones will be made available during practical labs so that students do not have to bear the cost of mobile phone calls.

Assessment, security issues, sharing practice, change in working practices, deadlines and instructing students on how to manage meetings will all be revisited. Clearer marking criteria will be provided to staff and students. Students also need to be made very aware that the process of software development does not begin and end with the actual implementation of the system but equally important is the whole process. This was often forgotten by the students who spend excessive amounts of time coding.

8. SUMMARY

The technologies provided during the project have supported communication and collaborative working and allowed us to engage students, capture their interest and make the work more enjoyable and realistic. Students have however struggled with elements of the assignment not least, the technical aspects of it but also with of the experience of how difficult maintaining good communications across-site is and how the quality of communication can greatly influence the outcome of the project.

The technologies provided have enabled our students to stretch themselves. They have had to communicate, coordinate and organise themselves more and to tackle new technologies, make plans for when things go wrong, realise dependency between their work and someone else's etc. throughout this project.

The introduction of a real customer next year has generated lots of interest as this adds a further dimension of realism. The interest in this work from our industrial contacts only encourages us to pursue cross-site development further and we hope the work outlined in this paper may serve to give guidance to others in academia contemplating undertaking similar projects. The risks that have been taken and the benefits to the students are hard to measure and quantify and we are currently assessing these. At present we have many observations and a lot of anecdotal evidence to offer the wider higher education community and our ongoing work in this area is proving very interesting.

"It was my first time to have team members from a different site so it really was a new experience for me. I found it very useful because now I know how complicated it is to work with someone who is not physically present" Student Comment

9. REFERENCES

[1] Offshore Outsourcing: current conditions & diagnosis, E. Ferguson, C. Kussmal, D. McCracker, M. Robbert, **Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education**, Virginia, USA, 2004, pp 330-331.

[2] CETL-ALiC (2005), "Centre for Excellence in Teaching and Learning: Active Learning in Computing." from http://www.dur.ac.uk/alic

[3] Supporting students to develop collaborative Learning skills in technology-based environments, Anne Nevgi, Paivi Virtanen & Hannele Niemi, **British Journal of Educational Technology**, Vol. 37, No. 6, 2006, pp 937-947.

[4] Access Grid http://www.accessgrid.org/home <Accessed 23/05/07>.

[5] Shibboleth http://www.matu.ac.uk/shibboleth_intro.html <Accessed 23rd May 2007>.

[6] SKYPE: http://www.skype.com/intl/en-gb/ <Accessed 23rd May 2007>.

[7] Eclipse: http://www128.ibm.com/developerworks/opensource/library/os-ecl-subversion/

[8] Subversion: http://subversion.tigris.org/ <Accessed 28th May 2007>.

[9] S. Drummond, M. Devlin, Software Engineering Students; Cross-site Collaboration: An Experience Report. **Proceedings of 7th Annual Conference of HEA-ICS,** Dublin, Eire, 2005**.**

[10] E. L. Burd, S. Drummond, et al., Using Peer and Self Assessment for Group Work. **Proceedings of 4th Annual LTSN-ICS Conference,** Galway, Eire, 2003**.**

[11] P. Race, A Briefing on Self, Peer and Group Assessment. **LTSN Generic Centre Assessment Series 9**, Learning and Teaching Support Network (LTSN), 2001.