An investigation into computer support for cooperative work in software engineering groups

Drummond, Sarah A.

University of Durham

Department of Computer Science

Master Thesis in Computer Science

An Investigation into Computer Support for Cooperative Work in Software Engineering Groups

Sarah A. Drummond

April 1999

University of Durham

Sarah Drummond

MSc Thesis in Computer Science

"An Investigation into Computer Support for Cooperative Work in Software Engineering Groups"

ABSTRACT

The research of this thesis relates to Computer Supported Cooperative Work (CSCW) in the context of software engineering, and in particular software engineering education. Whilst research into groupworking has tended to be directed towards CSCW, very little research has been undertaken on group working within software engineering. Linked with CSCW is groupware, which is the class of tools that supports and augments groupwork. This thesis represents an attempt to contribute to the understanding of the groupware needs of software engineers, and to identify and trial groupware that supports software engineering activities.

An infrastructure has been developed providing virtual environments, for use by both collocated and geographically distributed software engineering students, to support their groupwork. This infrastructure comprises of synchronous and asynchronous groupware, in the form of desktop video conferencing, and a shared information workspace. This shared workspace has been tailored from the groupware tool, Basic Support for Cooperative Work (BSCW).

Within this thesis, hypotheses have been formulated as to the student use of these virtual environments. These hypotheses concentrate on the areas of: organisation and coordination of tasks, the level of cooperation that occurs within the phases of the software lifecycle, the usage of the functions within a shared workspace, and what importance is placed on the role of synchronous communication within software engineering student groupwork. Through a series of case studies it was possible to determine the outcome of these hypotheses using various data collection methods. These methods include questionnaires, focus group meetings, observations, and automatic monitoring of workspace activities.

The outcomes of this thesis are that the hypotheses regarding organisation and coordination, and, the role of synchronous communication within software engineering, have been proved. Whilst the determination of the level of cooperation during the phases of the software lifecycle has not been proved, the use of functions within the shared workspace has been partly proved.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. Cornelia Boldyreff for her guidance, encouragement and friendship during the preparation of this work. I would also like to thank friends within the department for their encouragement, especially Liz Burd who patiently read the drafts for this thesis.

Appreciation also goes to Robert Walker and Alex Long who worked with me during the summers of 1997 and 1998 respectively, and of course, the SEG students of 1997-98 without whose feedback this work would not have been possible.

Thanks also go to my colleagues from UMIST and Keele University especially Roj Young and Matt Gumbley whom I worked with on the JISC(Joint Information Systems Committee) JTAP (JISC Technology Application Programme) project, "Developing a Virtual Community for Student Groupwork", which formed the basis for this work.

This thesis is dedicated to my husband Harry, whose love and support throughout my return to higher education has been unfaltering, and also my children Ellen and Harry who have put up with a part-time mother for so long.

STATEMENT OF COPYRIGHT

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

As the cost of computer hardware decreases the result is the proliferation of computer systems into all aspects of life and business, so much so, that personal, corporate, national and international economies are increasingly dependent on computers and their software systems (Sommerville 1995).

Many organisation strategists believe that successful corporations of the future will be those that make the networked organisation a reality, and emphasize that collaboration and coordination activities via flexible communication systems, should be made accessible in the office, at home, and on the factory floor (Bannon 1993).

Employees across multiple sites in various worldwide locations are finding they need to communicate and interwork ever more frequently. Organisations need to assemble teams members with varying skills and experiences in geographically distributed locations. Whilst research into groupworking has tended to be directed towards Computer Support for Cooperative Work (CSCW), which has emerged over the last five years as a research discipline in its own right, very little research has been undertaken on group working within the domain of software engineering.

Linked with CSCW is groupware, which is the class of tools that supports and augments groupworking (Salim 1998), and which the design and development of, have often been focused on by software engineers.

Software development itself consists of specification, design, conceptualisation, implementation, verification and validation. It is intrinsically about team work.

Most of the time it involves the participation of software designers, programmers, reviewers and end users. These people are commonly not located at the same place, especially in large software projects involving different organisations in different countries. Therefore, support for cooperative software development has become a major concern.

This support must take the form of better tools, techniques, methods and perhaps more importantly, better education and training for software developers. Software engineers must be better trained and gain experience in cooperative work if they are to make most effective use of this support.

Computer systems intended to aid groupwork must be built to fit the needs of groups. To achieve this, a deep, fundamental understanding of how people in groups work, and interact is essential. When group support is the main purpose of a technology, it will only succeed if its developers have an adequate understanding of the application domain and of its activities which the technology is to support. Gaining such understanding, until recently, has not been taught in many Computer Science departments.

This thesis represents an attempt to contribute to the understanding of the groupware needs of software engineers and, in particular, software engineering students, and to identify the gaps in commercially available groupware technology. The requirements and the results presented within this thesis are based on case studies undertaken where the subjects are undergraduate software engineering students.

Experimental research (to be presented in the following chapters), is often dismissed because it is done with students who are plentiful in numbers, but do not possess the skills and knowledge of experienced software engineers. To undertake such research in similar numbers with experienced software engineers is almost impossible, because of the time and cooperation that would be required. However, the qualitative data provided by this research gives an insight into a basic set of requirements needed for software engineering groupwork, and, in addition it has allowed these software engineers of the future to gain invaluable experience in using new technologies, and in understanding the concepts of cooperative work.

1.2 Criteria for success

The main objective of this research is to investigate the provision of groupware based support, specifically for software engineering students working both in collocated and distributed groups. This support takes the form of synchronous and asynchronous technologies supporting computer-based collaboration and coordination.

The criteria for success of this research are to achieve the following:

a) an investigation of the areas of CSCW and associated groupware to determine current commercially available software specific to software engineering.
b) an identification of a set of requirements for collaborative working support for software engineering students, both distributed and collocated, and a formulation of hypotheses regarding software engineering students' use of groupware.
c) an undertaking of case studies to prove or disprove the hypotheses (listed below) using groupware, with software engineering students.

Case studies involving different groupware technologies and different student workgroups are described within this research. Assumptions have been made at the onset of this work relating to the student use of the technologies. These assumptions are presented as a number of hypotheses. The results for these hypotheses are addressed in Chapter 6.

The hypotheses are as follows:

Hypothesis 1

The introduction of an asynchronous shared workspace into software engineering groupworking will aid group members in coordinating and organising their work.

Hypothesis 2

Students undertake more collaboration in the earlier stages of the software lifecycle[1].

Hypothesis 3

Greater use of shared workspace functionality will be made as the project progresses.

Hypothesis 4

Synchronous communication has an important role to play in both collocated and distributed software engineering groupwork.

1.3 Outline of Thesis

The thesis is organised as follows. Chapter two introduces the areas of research that are relevant to the work being undertaken. These areas are software engineering and software engineering education, CSCW and groupware. This chapter first discusses the area of software engineering and its requirements for supporting groupwork and, the problems and needs of software engineers working in distributed locations. Secondly the concepts and background of Computer Supported Cooperative Work (CSCW) are described. Thirdly, groupware which is seen as the technology that provides the support that cooperative work requires is described with examples. This chapter concludes with an overview of software

[1] The basis for this hypothesis is a suggestion made by Peter Wharton of ICL at the Centre for Software Maintenance Advisory board meeting in December 1997.

engineering education, and indicates appropriate educational strategies to ensure students of software engineering, gain the necessary skills to gain experience of this groupware, in order to utilise it effectively.

Chapter 3 describes the background to the work groups that have been used in the case studies. Initial pilot exercises undertaken with distributed student work groups are detailed. It is from the initial results obtained during these exercises that software engineering requirements have been identified. It is these results that have enabled further studies to be undertaken, which are described, for both distributed student projects referred to as the JTAP[2] project, and collocated students in the Software Engineering Group (SEG) project within the Department of Computer Science at the University of Durham.

Chapter 4 discusses the need for network support for groupworking within the SEG work. This support takes the form of the development and implementation of a virtual environment *SEGWorld.* This shared workspace was developed as a result of the earlier JTAP work.

Chapter 5 describes the case studies undertaken. A structure for each case study is detailed to include the design of each task, for both collocated and distributed students using asynchronous and synchronous tools.

Chapter 6 presents and discusses the results from JTAP phase 2, and SEG case studies. The results presented address the hypotheses described above.

Chapter 7 discusses the conclusions which can be drawn from the previous chapter and further work that can be undertaken.

[2] JTAP: JISC(Joint Information Systems Committee) Technology Application Program. This work involved the University of Durham, UMIST and Keele University.

Chapter 2

Research Areas

This chapter attempts to provide a framework for the study of software engineering in the context of Computer Support for Cooperative Work (CSCW), and groupware - the technologies that enable collaborative working. Section 2.2 discusses software engineering, and the problems developers encounter in communicating and coordinating work developed in collocated and geographically distributed locations. Section 2.3 describes the background and growth of the research area of CSCW. Section 2.4 describes the background of groupware with examples of groupware and associated communication modes. The final section 2.5 discusses the area of software engineering education.

2.1 Introduction

The main thrust of this research is applying CSCW to software engineering. In doing so, groupware is seen as a subset of CSCW, and software engineering education as a subset of software engineering. Figure 2-1 shows how these four areas intersect with one another, i.e. groupware is seen as a component of CSCW even though it represents the technological side of groupworking. The central intersection of Figure 2-1 is groupware support for software engineers, and, in particular, software engineering students.

The remainder of this following chapter is divided into four sections. These sections represent the areas of research described above.

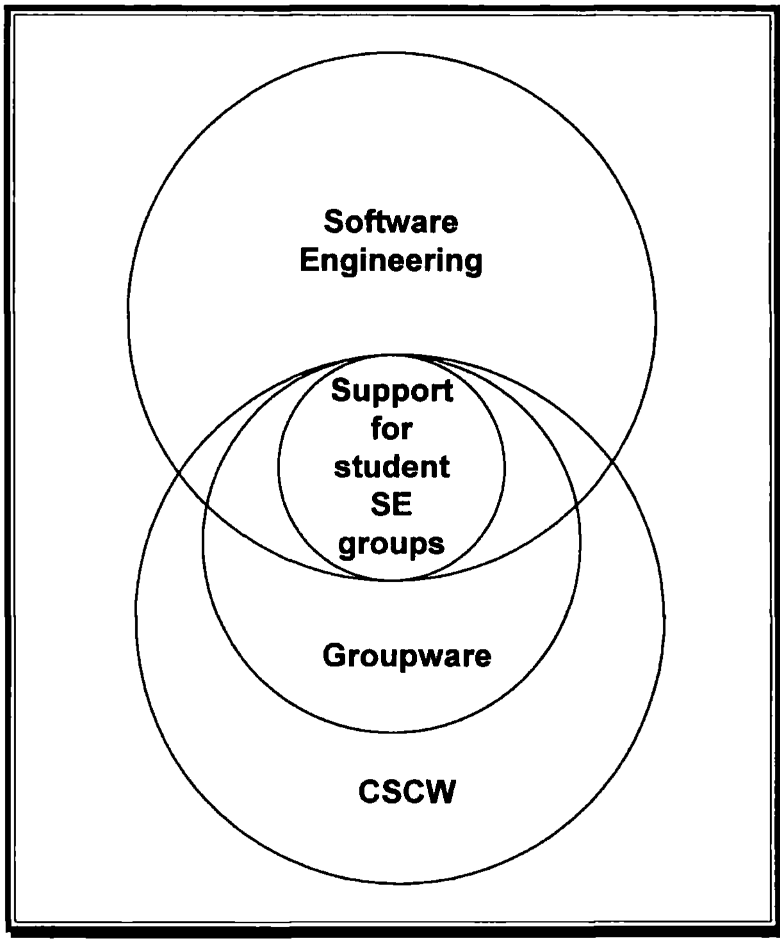

Figure 2-1: Areas of Research

2.2 Software Engineering

> *"The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software"*
>
> IEEE definition of Software Engineering

Software Engineering is an engineering discipline concerned with the practical problems of developing large software systems. It refers to the process of establishing the requirements for such systems and the designing, building, testing and maintaining of these systems. This process may involve a number of generic activities such as writing documents, brainstorming, prioritisation and reviewing. It could therefore be described as a collection of activities, many of which are common to other areas of team working.

System development is essentially a team process where developers spend a relatively large proportion of their time working with others. Estimations of this proportion of time have ranged from 50%, from an early study undertaken in 1978 by IBM(cited in (Sommerville 1992)) (Figure 2-2), with DeMarco and Lister (DeMarco and Lister 1987) seeing this figure as nearer 70% and Jones (Jones 1986) citing 85% for larger complex systems. What all these studies show is that system development is intrinsically a team activity.

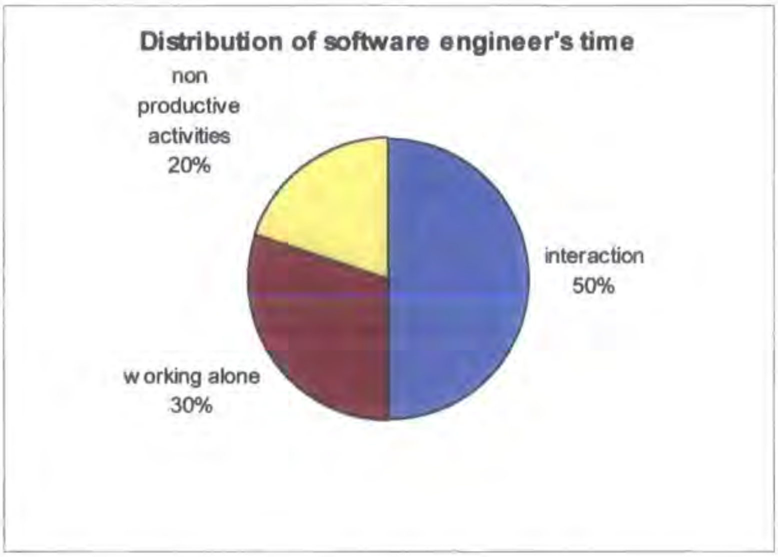

Figure 2-2: Distribution of Software Engineers time

Much of this team interaction is spent in an effort to maintain communication and control, especially on complex systems which require the collaboration of specialists over a sustained period of time (Forte and Norman 1992). While specialists also perform independent tasks, they also need to share information with one another through interaction. This interaction accounts for a significant part of the total cost of a system. Tasks such as design, programming, debugging, testing, and inspection typically require the participation of multiple engineers. Independent tasks occur more frequently in the specification and design phases of development where the work can be decomposed into individual tasks. What is necessary during these phases is to provide support for managers to be able to coordinate these individual tasks and to assist with project management in general.

Whilst there have been considerable recent advances in Computer Support for Cooperative Work (CSCW) technologies, the advancement of tool support for collaborative software engineering has been neglected. There are prototype tools that have been developed within research environments e.g. ISM (Rodden and Sommerville 1994), Flecse (Flexible Environment for Collaborative Software Engineering) (Dewan and Riedl 1993), ICICLE (Intelligent Code Inspection Environment in a C Language Environment) (Brothers, Sembugamoorthy et al. 1990), a generic, cooperative software engineering platform (Hawryskiewicz and Gorton 1996), but as yet none of these tools are commercially available. There are groupware tools available which can be used to support various aspects of software engineering, but these tools are generic and not specially designed nor developed for this domain. Typical software engineering activities are listed below (Table 2-1), together with an indication of how they can be supported by current generic groupware tools.

Software Engineering Activity	Groupware
formulate and exchange ideas	shared workspaces, shared whiteboard, video conferencing
hold meetings - formal and informal	shared whiteboard, video conferencing, chat tools
develop and edit graphical designs	shared drawing tools, shared workspaces
develop shared documents and reports	group document handling, shared workspaces, application sharing
presentations and demonstrations	video conferencing
track work in progress	workflow technology

Table 2-1: Software Engineering activities aided by groupware

To begin to develop tools for software engineering, it is necessary to first look at how teams of software engineers cooperate, and to understand their communication processes. Once this is understood, more specific software engineering groupware tool requirements can be developed. It is these requirements that can then be used as a foundation for future software engineering CSCW tools.

2.2.1 Characteristics of Software Development

A factor associated with many of the problems faced in the development of software systems such as cost overruns, late delivery, difficult maintenance etc., is poor coordination of development activities (Harrison, Osser et al. 1990; Kraut and Streeter 1995). Software development is a complex activity because of the multiple components a software system contains, each component often built by different people and needing to be integrated, which results in coordination overheads. This problem is compounded further as the project increases in size and complexity.

To build software efficiently, developers must have a common view of the software they are constructing. They must agree upon common definitions, share information and agree their activities. For instance, they must share detailed design specifications and information about the progress of software modules and how they should be integrated with other software systems. The coordination of these activities can be very difficult as software development has several characteristics (Kraut and Streeter 1995) which make this coordination problematic, such as:

- *Scale* - the project may be large, involving millions of lines of code and therefore beyond the capability of one person to understand the system.
- *Uncertainty* - in software development, many software systems are 'one-of-a-kind' and often the specification changes throughout the life of the software development. The specification may be incomplete because of the limited knowledge of the developers in the domain area in which they are working. Sub-groups of a team may also have different ideas about what the software should do.

- *Interdependence* - poor coordination amongst subgroups involved in a system comprising of thousands of modules which have to be integrated for the system to work properly, can cause major disruption. For example, the AT&T long distance network disruption which occurred in 1989 was due to unanticipated interaction between modules.
- *Communication* - is a vital factor within development teams for the success of a project. Both formal means (e.g. written requirements specification, reviews, tracking of program errors), and informal means (e.g. unscheduled meetings) are valuable for achieving communication. In formal project meetings, different stakeholders may make or review decisions on, for example, functional requirements. A major problem with this type of meeting is that there is often a disproportionate attendance of local representatives and few domain experts or users. Informal communication may entail a software developer asking for help from someone within the same locality as a matter of convenience, rather than the person having the more appropriate competence.

Whilst individual staff understand different components of an application, the deep application-specific knowledge required to successfully build most large, complex systems is often thinly spread through software development staff (Curtis, Krasner et al. 1988). There is a need, therefore, to be able to increase the amount of domain knowledge across the entire development staff, so that they can all share and integrate this knowledge.

2.2.2 Distributed and collocated teams

The software engineering process involves the participation of software designers, programmers, end-users and domain experts. These people are commonly not located at the same place, especially on large software projects. Groups no longer need to meet in the same location; new technologies are allowing us to relax the constraint of collocation. Modern communications make available an interesting array of options such as teleconferencing, video conferencing and synchronous interactions over computer networks. These options are not designed to replace the rich face-to-face interaction which is a central component of cooperative work (Tang and Isaacs 1993). But face-to-face interactions is expensive and at time, logistically impossible, and these options provide a viable alternative in many instances.

Distribution is often an unavoidable consequence of organisational factors e.g. technical resources may only be available at specific locations, skilled workers are distributed across sites etc. There are both advantages and disadvantages to distributed work. It can be advantageous for the development team to be separate from the testing team so that testing is not influenced by the development team. In addition to this, because distribution can make it difficult for relationships to be built up over distributed sites, it becomes necessary for processes and interactions to be rigidly formalised and documented. By the same token, the lack of regular face-to-face meetings and therefore the lack of good working relationships can be problematic to a project.

Introducing software development onto the Internet is steadily growing because of the potential advantages it offers. By getting teams of people together who are situated at different locations as well as collocated, it is possible to have specialist centres, i.e. one site concentrates on design, another development and a third on maintenance. Opportunities such as this can be obtained if group work can be fostered across distance and group support tools are available.

Currently there are no specific groupware products commercially available for software engineers. Technology does provide many tools i.e. CASE tools, for software engineering, but these are mainly designed for the single user. Vessey and Sravanapudi (Vessey and Sravanapudi 1995) carried out initial investigations into the desired features of a collaborative support CASE tool. They looked at four commercially available CASE tools and in general, found that the strongest support that these tools offered, was for information sharing.

A relatively simple approach to group support is to provide a virtual group environment that supports face-to-face interaction, and in particular groupware tools such as desktop video-conferencing, and shared information spaces such as Lotus Notes™[3] and BSCW (Basic Support for Cooperative Work) (GMD-FIT). It is the use of such groupware tools that have the potential to support groupworking.

The purpose of the next two sections is to introduce the research field of CSCW and review the major classes of groupware.

2.3 Computer Supported Cooperative Work

2.3.1 Background

Technological developments of the 1980s have resulted in the personal computer bringing computer power onto the desk and into the hands of a variety of end users (Plattner 1994). This wide-scale introduction of personal computing was followed closely by the trend to network these systems together, allowing users access to a variety of services. Developments in telecommunications infrastructure, such as Integrated Services Digital Networks (ISDN) and the Internet (TCP/IP) has made access to these services widely available.

Running along a parallel but separate path, investigations of how individuals and groups functioned in computer mediated working environments were being undertaken by a variety of disciplines. This, together with the technological advances, resulted in a new research area centered around the use of computers to support human communication - especially group communication. This research area is Computer Support for Cooperative Work (CSCW).

[3] Lotus Development Corporation: http://www.lotus.com/home.nsf

The concept of computerised support for cooperative work was pioneered by Douglas Englebart in 1968 with the development of an experimental meeting environment known as On-line System or NSL (Englebart and H.Lehtman 1988), which allowed office workers to communicate either by exchanging documents or by interaction in real time through a shared window. This system underwent additional development and was made commercially available under the name Augment. The actual term Computer Supported Cooperative Work (CSCW) can be traced back to 1984 when Irene Greif and Paul Cashman used it as a shorthand way of referring to a set of concerns about supporting people working together with computer systems.

Since the mid-80s, the area of CSCW has been investigated more intensively, and CSCW has emerged as an identifiable research area. The research focuses on how people work together in groups, and on how computer and related technologies have impacts on group behaviors (Bannon 1993). Since this time, conferences both in the USA (CSCW 1986, 1988, 1990,1992, 1994, 1996) and Europe (ECSCW 1989, 1991, 1993, 1995, 1997) have attracted an every increasing number of delegates from many disciplines with differing perspectives.

There have been various attempts to form a definition of CSCW but no universally accepted definition has been accepted (Kling 1991; Wilson 1994). This is due to the diverse range of disciplines involved, and hence the differing perspectives. For instance, the sociologists and psychologists carry out research into groups and their dynamics, whereas the computer scientists have interests in distributed multimedia applications and networks.

Bannon (Bannon and Schmidt 1991) perceives CSCW as *"an endeavor to understand the nature and characteristics of cooperative work with the objective of designing adequate computer-based technologies"*. Grudin (Grudin 1991) on the other hand, defines CSCW by the specification of features such as group size, composition, organisation, time course and physical location. These statements show there is no agreement on a common definition, but both highlight the problem of having to consider not just the technology, but the whole socio-technology system. Conversely most authors agree on the following principles (Williams, Blair et al. 1994):

1. work is a cooperative activity, generally involving groups of people interacting to achieve common goals, and
2. the designers of supporting computer systems must address this cooperative nature of work.

Checkland (Checkland 1997) believes that it is unfortunate that CSCW even begins with the word - Computer - as the initial focus should be on the group activity not the technology. In Checkland's opinion it would have been more appropriate to call this area of research "Cooperative Work Aided by Computers": CWAC.

The term CSCW is therefore a misnomer. It implies a more limited field of study than the diverse range of disciplines which are involved. Contributions come from Sociologists, Anthropologists, Psychologists, Computer Scientists, Human Factor Specialists and many more (Wilson 1994). CSCW can be seen as a variety of disciplines coming together, all with some overlapping interests concerning people, computers and cooperation. Howard (Howard 1988) sees these communities as either the "strict constructionists", whose focus is on developing computers systems - the tool builders, and the "loose constructionists", who are the heterogeneous group of people who see the area of CSCW as an opportunity for them to prescribe how these groupware systems should be designed.

Therefore CSCW can be seen as involving a paradigm shift wherein, the emphasis within the design of people-technology systems must be on understanding the social organisation of work, how an individual communicates and works, and similarly the way group dynamics affect the way people collaborate. The implications of this paradigm shift need to be reflected into the design of the technologies such as email, video conferencing, databases, and shared information space facilities to support cooperative work.

In this research, CSCW is viewed as the study of people working together using computer technology. Typical topics of study include the use of shared workspaces, monitors providing awareness of the activities of other users, email, videoconferencing, chat systems, and real-time shared applications such as, collaborative writing or drawing systems.

CSCW is often divided into the domains of synchronous (or real-time) work, which considers people who are working together at the same time (such as with videoconferencing), and asynchronous work, which considers people coordinating their efforts across longer periods of time (such as with shared workspaces and email). A useful classification of CSCW is given by considering the location of the cooperative work in terms of time and place (Table 2-2).

	Same Place	Different Place
Same Time	face-to-face and synchronous	synchronous:
Different Time	asynchronous	asynchronous:

Table 2-2: Time and Space Taxonomy

The research described here is concerned with supporting software engineers who carry out both synchronous and asynchronous work.

2.3.2 Terminology disparity

The distinction between terms such as computer supported teams (Johansen 1989), cooperative work, collaborative work, collective work and group work, are not well established in the CSCW community. With work having many varied facets, it is no surprise that many synonymous terms abound. Howard (Howard 1987) states that "collaborative work", with its literal meaning being "*to work together, especially in some literary, artistic or scientific undertaking*" is too specific, but "cooperative work" is too general because "*all human activity is in some sense cooperative*". Bannon (Bannon and Schmidt 1991) believes that all these terms have different connotations and designate different types of cooperative work and that we should not abstain from using any of them.

The term "cooperative work" will be used throughout the remainder of this thesis as it is particularly appropriate to the work that has been undertaken. In the domain of software engineering some work is carried out by individuals. The divergent nature of this work, i.e. the software engineers will cooperate when producing their different sections of a document and then collaborate when they bring these together to form the complete deliverable. This is shown in Figure 2-3

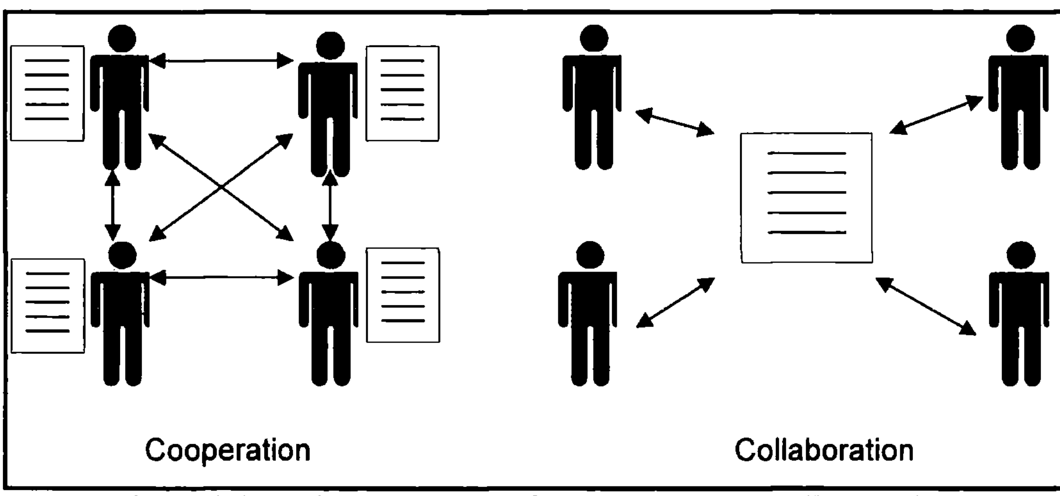

Figure 2-3: A simplistic view of cooperation and collaboration

Figure 2-3 is a good representation of the type of work that has been undertaken within the case studies presented in this research.

2.3.3 Growth of CSCW

Rapid developments in technology have resulted in the isolated PC in a working environment becoming less common due to the increased availability and use of local area networks (LAN). LANs have in general allowed greater connectivity both locally and globally. Many software systems only supported the interaction between a user and the system, i.e. preparing a document, querying a database or in the use of a spreadsheet. Whilst spreadsheets are traditionally seen as a single user system, Nardi and Miller (Nardi and Miller 1990) argue that end user use of spreadsheets is in fact, often a cooperative activity because the users construct their spreadsheets often enlisting the help of other more knowledgeable and experienced co-workers. Therefore, Nardi and Miller see the use of the spreadsheet as a possible medium of group communication.

CSCW has also grown because of the changes in the field of information systems practice, and in peoples' expectations of the technology itself. There has been a shift in emphasis from 'automating' the office to 'supporting' workers with computer systems rather than replacing them (Bannon and Schmidt 1991). Human and social factors show that people want more flexibility, with access to information to be anytime, anywhere, for them to work more effectively. The main problem has been the incompatibility between computer systems and the inability for many of the applications in use to support multiple users effectively. As more and more work is carried out using computer systems, the gaps and inability's of these systems to support groupwork are becoming more apparent.

The consequence of this, is that the area of CSCW is seen as a potentially huge new market by both software developers and network providers, who have become interested in the connectivity and high bandwidth demands of CSCW applications. These applications are known as groupware.

2.4 Groupware

The term 'groupware' was coined in 1978 by Peter and Trudy Johnson-Lenz (Lloyd 1994). They defined groupware as "*a whole system of intentional group processes plus software to support them*" but more recently, they have shown that groupware is a computer-mediated culture. This ties in more closely with Ellis's view (Ellis, Gibbs et al. 1991) where he defines groupware as "*computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment*".

2.4.1 Background

Computer systems and software tools have penetrated most areas of traditional work practice. Widespread availability of personal computers has contributed to this situation. However, most of these systems have been considered in isolation, both from other tools and from other people, or groups using similar tools (Rodden 1993). The availability of networking technology and the need for cooperative activities of projects, has led to the development of systems that aim to support groups.

Groupware has been around since the 1970s but was not widely used simply because of the lack of network infrastructure. It was not until the late 1980s that groupware became more widespread. Text processing programs which were widely used as single user applications became the first target application for groupware. This groupware provided benefits such as allowing multiple users jointly to edit, annotate and revise shared documents thereby increasing the efficiency and quality of group writing (Galegher and Kraut 1990; Ellis, Gibbs et al. 1991; Olson, Olson et al. 1993; Sasse, Handley et al. 1993).

Groupware lies on a network infrastructure and is part of the networked applications environment (Figure 2-4).

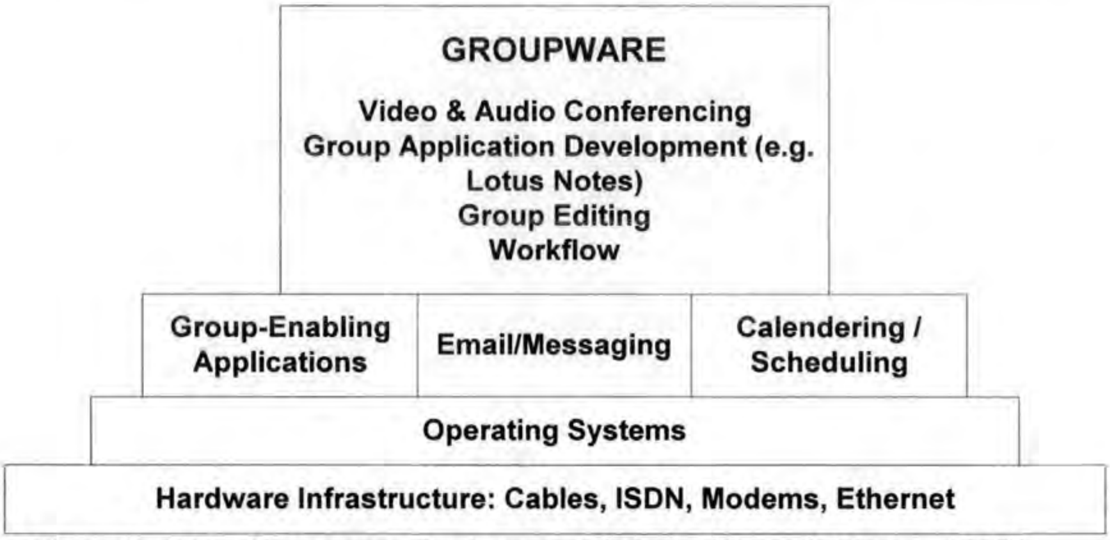

Figure 2-4:The Groupware Environment (adapted from (Coleman 1995))

2.4.2 Classification of groupware

Groupware is a broad term for a group of related software technologies. There have been several classification schemes outlined with the best known classification by Johansen (Johansen 1989), which focuses on the time and place taxonomy of interaction (Table 2-3).

	Same Place	Different Place
Same Time	flip charts, documents, audio/visual aids	IRC[4], MUD's[5], shared editors, DVC[6], POTS[7]
Different Time	email, shared work spaces, post-it notes	email, mail, fax, shared workspace

Table 2-3: Groupware: Time and place interaction

Below is an overview of a selection of groupware tools which have been developed to support group working (Coleman 1995):

[4] Internet Relay Chat
[5] Multi User Dungeon
[6] Desktop Video Conferencing
[7] Plain Old Telephone System

1. **Email** - the most widely used mode of interaction.
2. **Conferencing tools** - include newsgroups, forums and discussion databases. They allow users to carry on a conversation over time or post information that other users can access.
3. **Chat, Shared Whiteboards and Video Conferencing** - commercial packages such as Whitepine CUSeeMe offer synchronous chat. Shared whiteboards allow users to view and edit documents/diagrams simultaneously. Video conferencing allows two or more remote participants to communicate through sound, video, chat and shared whiteboard from the desktop PC.
4. **Collaborative writing tools** - allow people to work together to create and edit documents. The users are provided with a shared screen and document management system.
5. **Shared drawing tools** - allow users to simultaneously draw on a shared WISIWYS (what I see is what you see) drawing pad.
6. **Shared workspaces** - allow the storage and retrieval of documents and other shared information for groups.
7. **Group calendaring and scheduling tools** - support group and resource scheduling through the use of electronic schedulers connected to individual diaries. These tools can be used to organise meetings and place details in members diaries.
8. **Workflow** - provides automation of procedures where documents, task etc., are passed between different people. In addition, these tools provide information necessary for the people to undertake these tasks, notifying users about who has the document, its current status and any over runs in its production plan.

These groupware applications can be divided into two modes of communication. Firstly, there are a number of applications which support synchronous (real-time) interaction between groups of users. Secondly, there are those applications intended to support asynchronous exchange of information between users.

The following section describes the modes of communication that have been used within this research.

2.4.2.1 Communication Modes

The communication modes are as follows:

- synchronous
- asynchronous

2.4.2.1.1 Synchronous

Synchronous interaction requires the presence of all cooperating users. There are many examples of synchronous systems, e.g. shared document editors, shared whiteboards, video conferencing. Advancements in computer technology are making desktop video conferencing (DVC) a viable communication medium over data networks: TCP/IP and ISDN. DVC combines personal computing with audio, video, and communication technologies to provide real-time interaction from a typical personal computer[8]. It is this interaction that facilitates communication amongst the members of geographically distributed groups.

There are two major types of communication channels available to transmit this data: circuit- and packet-switched (Rettinger 1995). Circuit-switched channels, such as ISDN, offer dedicated bandwidth and predictable timing of data delivery, but they do not easily support multipoint communication, which is required for the type of collaborative working that is illustrated in the following chapters. Packet-switched channels, either local (LAN) or wide area (Internet), more easily support multipoint communication, but they do not provide predictable timing of data delivery. DVC systems have requirements for both timely and reliable data delivery. For example, audio and video data require timely delivery while other types of data, such as whiteboard data, require only reliable delivery.

[8] DVC should not be confused with studio based video conferencing.

Real-time audio and video over the high speed academic network SuperJANET in the UK is a low-bandwidth alternative to the more expensive forms of communication (i.e. ISDN). The use of an ISDN line assures high quality audio and video transmission using DVC, and through the use of connecting to a central site, allows multiple users to conference simultaneously.

2.4.2.1.2 Asynchronous - Shared Information Space

In any cooperative work situation, there is a need for some form of communication or information sharing (Bannon and Bodker 1997). A shared information space can come in various forms depending on the work environment e.g. collocated or distributed work groups. All of these forms provide a shared repository for the sharing and editing of files as well as allowing threaded discussion forums: an essential element in any collaboration. Using shared information spaces usually incurs overheads especially if the group is distributed. There is a need to construct and manage the information in such a way that the grouping of the information is meaningful and understandable by the group members. This is relatively easy if the group are collocated, as they have the same work setting and exposure to the same work, without the need for extended descriptions of the work.

Examples of tools that can be used to develop shared information spaces are BSCW (used extensively in this research) and Lotus Notes™. These are described in more detail below.

2.4.2.1.2.1 Basic Support for Cooperative Work (BSCW)

BSCW is an asynchronous shared workspace system and has been used extensively in the case studies within the following chapters. BSCW was conceived as a means of supporting the work of widely dispersed work-groups. It is a document storage and retrieval system extended with features to support collaborative information sharing (GMD-FIT).

The BSCW system consists of a server which maintains an index of all the workspaces it manages. These workspaces are accessible from different platforms using standard WWW clients. Users access the BSCW system using a standard user-name and password and the server responds with a list of the workspaces the user can enter. Each workspace contains a number of shared information objects (each labeled with icons indicating the objects development application). Workspace members can perform actions to retrieve, modify and request more details on these objects. These objects can be documents, links to WWW pages, folders, groups and members. A notification service keep users aware of each others activities.

2.4.2.1.2.2 Lotus Notes

Lotus Notes™ is a group information manager that allows teams to be more effective by allowing them to access, track, share and organise information even if they are only occasionally connected to a network. Lotus has been developing an academic application called LearningSpace which uses the functionality of Notes to manage educational courses. Also the latest release of the Lotus Notes™ server, Domino, allows access via the Internet using a web browser instead of a traditional Notes client, thereby solving the problem of users accessing the server from different machines.

Notes' replication process synchronises database replications over time. It does not provide instant information exchange between geographically distributed sites that depend on separate Notes servers.

Team activities follow established patterns or procedures and therefore there is a need to keep track of these procedures.

2.4.3 Benefits of Groupware

Groupware is an umbrella term for the technologies that support person-to-person collaboration. It can be anything from email to electronic meeting systems to workflow.

The computing infrastructure of organisations is moving from mainframes to distributed computer services based on PC-LANs, rather than centralised computer services provided by mainframes. This is a prerequisite, and incentive for the adoption of groupware. Groupware is basically used, to do the same kinds of things that are already being carried out in offices, but in a better way (Coleman 1995). It provides support for collaboration between people and as such potentially offers many benefits. These benefits include the following:

- improving the quality of access to shared information: groupware can serve as a shared space where collaborators can represent and manipulate their ideas and it can act as a vehicle for co-creation and innovation (Schrage 1996).
- reducing the need for face-to-face meetings: groupware allows for two or more people from remote locations to hold a conversation or meeting with each other through a combination of audio, video, chat and shared whiteboard.
- improving the speed and accuracy of group decision making using teleconferencing, videoconferencing.
- providing better group management: planning and scheduling of group activities, in tracking progress and coordinating the activity of individual members through shared information spaces.
- improving workflow management: workflow tools allow for the automation of the procedure involving documents or tasks being passed amongst group members, and providing each member with information about their input.

2.4.4 Problems of Groupware

Groupware development can be defined as development of IT applications that support groups, this type of definition implies a technical area that is solving the technical problems of providing multi user facilities. But groupware is not just the technology, and failure of groupware can be attributed to a purely technological focus during development with insufficient attention being paid to the specific needs, and requirements, of cooperative work; in other words how it will impact on the way people work.

To understand the workplace, it is not sufficient to refer to an organisations policy and standard practice documentation. The dominant research methodology used to study workplaces within CSCW is ethnography. Studies of group behavior show that inconsistencies, irregularities and unpredictability are widespread (Heath and Luff 1991). Routine practice within an organisation must be observed to gain a clear picture of how a sequence of events actually occurs. Therefore, Ethnographers have focused their attention on existing working practices, and their studies help to build an understanding of group dynamics and the flexibility required, to support existing work, into the computer support systems.

Failure of groupware can be traced to designers naïve assumptions about how and who would be using the technology (Bowers, Button et al. 1995) (Ellis, Gibbs et al. 1991). Every workplace setting is unique, and this is reflected in the interpretations of formal and informal work practices in various studies (Plowman, Rogers et al. 1995). Lack of recognition of the changes needed for employees to adopt these working practices, so that the groupware works as it was intended, is another key factor of failure. Groupware that forces people to abandon familiar tools and methods of working is likely to meet with strong resistance (Grudin 1991).

Many groupware developers have simply hoped to emulate the success of single user software such as word processors or spreadsheets, but "a group of people is not a scaled individual" (Schrage 1996). Often with groupware applications, the effect of the tools on the group is not easy to predict. For example, in the Information Lens system (Mackay, Malone et al. 1989), which manages users email, users exploited a feature in the prototype in a novel and unforeseen way. This exploitation, which was not anticipated, was subsequently supported in the next version.

The well documented groupware failure described in Grudin (Grudin 1988), is the lack of uptake of automatic meeting scheduling systems. To gain collective benefits of any groupware, it has to be accepted by the majority as a common tool; in this particular case, it was not. Online calendar use and group scheduling systems have now become more widely used because they have been integrated with email, have intuitive interfaces and popular individual user features e.g. reminder mechanisms (Grudin 1996).

Most early groupware research and groupware development was done for the business community, and, as such, it was not aimed at or widely adopted by the academic community. This was due to the cost and the client licensing basis of the technology (Young 1998). However, there is an increasing need for universities to embrace this technology in order to give students vital experience, and to compete in the increasingly competitive academic market.

Coleman and Khanna (Coleman and Khanna 1995) suggest that the biggest challenge facing the groupware market is "education". Whereas Coleman is concerned with educating business people about the need to collaborate, the thrust of this research takes one step back and focuses on the education of the potential groupware developers of the future - the software engineering students.

2.5 Software Engineering Education

As technology rapidly evolves, software engineers must deal with new methods, tools, platforms, user expectations and software markets. This changing environment highlights the need for software engineering education that not only teaches current technologies, but also trains engineers to adapt quickly to new technological advancements (Garlan, Glutch et al. 1997).

Gibb (Gibb 1989) defines education as a long term activity designed to build a foundation of knowledge and reasoning abilities; and training, as a short term activity but with specific goals. What educators need to do is to give a good foundation so that the advancements in the technologies can be used effectively after a short training period. Brooks (Brooks 1986) stresses the need to teach students to think like software engineers rather than train them in many different languages, methodologies and tools. He emphasises that it is important for these students to be exposed to some languages, methodologies and tools, but in such a way as to "shape ways of thinking", and to provide experience at using such tools to develop and facilitate the implementation of new tools in the field. Thus, students need to be able to understand concepts and how to apply them to real problems. To use their knowledge in different contexts, they need to be exposed to more than one technology.

Software engineers need to be adaptable, and to be able to deal with constantly changing technologies; therefore, these students need to be able to assimilate technology quickly and effectively. It is important for students to understand the use and value of tools within a software engineering project and software engineering processes (McCracken 1997).

In addition to technical skills and knowledge of software engineering concepts there is a great need for these students to be taught communication skills.

Prospective employers of software engineering graduates emphasise that a critical requirement of these students is the ability to communicate. There is, therefore, a strong emphasis on the need for software engineering students to have team skills and negotiation skills in addition to being able to communicate on different levels and within different disciplines. These graduates must be able to understand both the technological and sociological factors associated in developing software that is to be successfully integrated into organisations for group use.

The remainder of this research looks at the introduction of software engineering students to new technologies and tools. These tools are to support and supplement their existing group work within both distributed and collocated environments. The exposing of the software engineering students to these technologies is an attempt to, as Brooks suggests, "shape ways of thinking" certainly about how they have to work together, and also about the social factors that need to be understood when designing and implementing software to support groups.

2.6 Summary

This chapter has shown and, in particular, reviewed the benefits and problems encountered within CSCW and groupware. What has been shown is the need to understand that the success of the 'field' of CSCW does not simply depend on the success of the groupware systems produced, but is concerned with how people use tools and perform their tasks, and, therefore, the design and implementation of these systems must take this into account. This understanding needs to be brought into the education of software engineers.

This research is not concerned with the design or implementation of groupware, but in determining existing groupware appropriateness to software engineering and in particular software engineering student group working.

The following chapters provide more details of the case studies undertaken with students, both collocated and distributed, using asynchronous and synchronous technologies.

Chapter 3

Background to the Work Groups

This chapter describes the background to the work groups upon which the remainder of this thesis is based. Section 3.2 outlines the objectives of the JTAP project with sub-sections 3.2.1 Phase 1, and 3.2.2 Phase 2 describing the short term distributed software engineering student tasks, and distributed final year projects respectively. In section 3.3 Software Engineering Group (SEG) work based locally at Durham is described. In section 3.5 the results of JTAP Phase 1 are used to refine a set of requirements for software engineering group working activities. Section 3.4 briefly summarises each student workgroup.

3.1 Introduction

Software Engineering group projects have been widely adopted in many undergraduate and postgraduate Computer Science courses (Horning and Wortman 1977), (Robillard 1996),(Drummond, Boldyreff et al. 1997). As well as reinforcing theoretical concepts, they provide students with experience of the type of team work found in industry. Over the next 20 years, software will increasingly be developed by geographically distributed teams; therefore, introducing new aspects of cooperative working into software engineering practice. It is important that students gain experience of this mode of working.

Whilst it is important to select the most appropriate technology to best support groupwork, the success of distributed and collocated software engineering projects also rely heavily on understanding and addressing a number of non-technical issues, primarily these are group and work organisation.

The specific focus of research over the past two years within the JISC/JTAP Project - "Developing a Virtual Community for Student Groupwork" (JTAP 1996-98), has been on collaborative student groupwork in software engineering. This research has focused on both organisational issues as well as the technology that is available to support groupwork.

The following sections describe the work undertaken within the JTAP project, this is then followed by a description of the software engineering group (SEG) work carried out at Durham.

3.2 JTAP Project

In the period 1996 to 1998, the Computer Science Department at Durham has been participating in a project with the Department of Computation at UMIST and the department of Computer Science at Keele University on "Developing a Virtual Community for Student Groupwork"[9]. The primary objectives of this JTAP project were (Bennett, Munro et al. 1996):

- to give students experience of working collaboratively in geographically distributed teams, using modern group working technologies such as video conferencing (video, audio, shared whiteboard, chat tool).
- to develop staff experience of operating this type of distributed project and to produce deliverables which will enable such projects to be successfully implemented in other institutions with maximum benefit at minimum risk or cost.

[9] Work described in this thesis relating to JTAP has been undertaken as a joint venture of the three universities involved, and specifically by the Research Associates (RA's) assigned to this project. One of the RA's is the author of this thesis.

- to assess rigorously the results of the project so that others have confidence in the validity and repeatability of the results.
- to provide requirements for future development of CSCW tools

The project's goal has been to make a significant contribution to realising the potential of collaborative technology within higher education with same-time/different place taxonomy (see Table 2-2), as well as exploring the role of asynchronous communication mechanisms such as email and shared document storage i.e. BSCW (Basic Support for Cooperative Work) and Lotus Notes™.

The JTAP project has been focused on the needs of distributed software engineering and computer science students undertaking groupwork. The project initially undertook short investigative pilot studies and as a result of these, further prolonged case studies were made. These studies were undertaken using distributed groups formed with final year students from the three universities. The main distributed group projects were carried out as part of these students' final year projects.

Both synchronous and asynchronous communication were required to support the group working. For the distributed JTAP students, synchronous interaction in the form of DVC was necessary. For both JTAP and SEG (see section 3.3) students, there was also the need to provide a common asynchronous repository - a shared workspace.

Whilst these technologies were required, in the main, by the student work groups, this technology was also required for the staff who were organising and running the projects; therefore, evaluation of suitable commercially available technologies had to be considered from both students and staff perspectives. To do this, initial evaluation pilot exercises were undertaken to determine the feasibility of distributed students undertaking simple software engineering tasks using DVC. In addition to these exercises, the JTAP project officers also used the shared workspace BSCW, as a central repository for all project deliverables.

The following section describes the pilot exercises and discusses results obtained.

3.2.1 JTAP Phase 1: Pilot exercises to evaluate the usability of adopted groupware technology

The definition of usability used in this work is as follows:

> "Usability is the effectiveness, efficiency and satisfaction to which specified users can achieve specified goals in particular environments".
>
> ISO CD 9241-11.3, version 8.8, May 1993

During the adoption phase in the JTAP project, different controlled exercises were carried out to evaluate DVC usability with respect to various software engineering tasks. Typical software engineering activities undertaken, many of which are common to team working, e.g. exchange of ideas, work allocation, develop and edit graphical designs, develop shared documents etc., can be supported by groupware, although no one technology alone may support all tasks. Studies undertaken by Whittaker and Geelhoed (Whittaker, Geelhoed et al. 1993) identified sets of general workspace tasks i.e. textual, graphical, for which particular technologies were most or least effective although they did not focus on tasks specific to a particular area of work and their concept of a workspace was limited to an electronic whiteboard.

The findings from our pilot exercises were used in refining a set of requirements for the subsequent and prolonged group working activities.

3.2.1.1 Objectives of Pilot Exercises

The objectives of these pilot studies which were essentially exploratory in nature, were to establish the usability of the distributed group working environment. The studies involved determining the feasibility of typical software engineering tasks being carried out collaboratively using synchronous modes of communication, and to give students limited, low-risk experience of distributed collaborative working.

3.2.1.2 Pilot Exercises

In the pilot exercises, various software engineering tasks were identified. The students were required to carry out these tasks in a networked computer environment using a variety of groupware support tools and technologies. The pilot exercises related to the software engineering activities of:

- discussion and negotiation
- diagramming
- review and evaluation

These tasks was carried out using synchronous communication only i.e. audio, video, chat and shared whiteboard (Table 3-1).

Feature	
Video	allows the participants to see each other simultaneously
Audio	for questions, comments, discussion
Text Chat	allows all participants to type messages and see the messages of others.
Whiteboard	on screen workspace and is similar to the whiteboard found in a conference room. Images can be loaded e.g. presentations or drawings from other applications which can subsequently be annotated by all conference participants.

Table 3-1: Features of DVC

3.2.1.3 Subjects

These pilot exercises involved four groups of three, third year Computer Science students from each university taking part.

3.2.1.4 Environment

At each site, a small collaborative working laboratory was set up. These laboratories each consisted of three multimedia PCs with the Connectix QuickCam camera and either headsets or standalone microphones. Each site had the same configuration for both hardware and software.

3.2.1.5 Modes of Communication

Synchronous modes of communication i.e. DVC were used which included: shared whiteboard, chat tool, video and audio. To support the groups' requirements, multipoint conferencing is necessary therefore, a local reflector at Durham as well as a JTAP project reflector at UMIST were used. A reflector is a UNIX or NT based application with the ability to hold multiple independent conference simultaneously. It accepts multiple client connections and reflects the video, audio, and additional data to all participants concurrently.

3.2.1.6 Evaluation techniques

Questionnaires were used to assess the usability of the groupware tools with respect to the specific tasks of requirements analysis, preliminary design and, requirements assessment carried out during the pilot exercises. Three questionnaires were designed. The first taking the personal details of the user, the second, an anonymous questionnaire, to find out the level of computing experience and group working background for each user. These two questionnaires were completed by each user before the task was started. The final questionnaire was a more comprehensive evaluation of the usability of the tools used during the tasks.

The analysis process was based in the main, on the completion of these questionnaires. Whilst questionnaires are a low cost, low disturbance mechanism for data capture, this data alone can not always be trusted to produce meaningful results when used in isolation (Jorgensen 1995). The completion of questionnaires can, in some cases, have low priority with the respondents. Answers can therefore be based on best guesses and relying on the memory and objectivity of the respondents. This best guess can manifest itself in the scale given for answering i.e. 1,2,3,4,5 where 1 is poor and 5 is very good. Many users often choose the neutral ground e.g. 3. In addition, ambiguity can occur when respondents have different understanding of terms presented. Therefore, to help achieve a higher quality of results, observation-based approaches and post focus group meetings have been used to augment this process.

3.2.1.7 Pilot Exercise Results

All tasks were completed with some degree of success by all groups although it was necessary to modify the group constituents on two occasions (a student failed to turn up, and the equipment at Durham failed). A brief summary of questionnaire responses are outline below (Brereton, Lees et al. 1998):

- Most students enjoyed using the tools and were successful in their use.
- Audio quality was not high although in many cases it was quite useable.
- The shared whiteboard was universally popular although many participants pointed out its limitations and opportunities for improvement
- The Chat facility was generally considered useful and enjoyable to use but was somewhat overused by those participants who failed to communicate successfully using audio.
- The video did not rate as very useful other than putting "a face to a name" because of the low refresh rate. It did however, provide limited cues as to what other members were doing i.e. typing, talking, reading.

3.2.1.8 Problems encountered

From observations, questionnaire responses and task products, it was clear that many participants would have benefited from more training. This lack of training manifested itself mainly in the lack of confidence found in the majority of students when using the technology. In particular problems with audio resulted in the students immediately refusing to use it, or, being very hesitant because they were never sure if the others in the group could hear them. The pilot exercises were designed to allow for 20 minutes familiarisation with the technology followed by a 20 minute period of "ice breaking", but these 40 minutes was generally spent by the students familiarising themselves with the technology only.

There was clearly considerable scope for improvement in the products used; for example, an unexpected problem was that the students expected telephone quality audio and were therefore disappointed. Rather than persisting with its use (and making adjustments to settings, microphone position, etc.) the students quickly reverted to using the Chat facilities (which they were generally quite proficient with). This problem relates back to the lack of training. Those students who undertook more than one exercise were more effective and enthusiastic users of audio the second time around. The video was not considered very useful, but it did give some valuable cues. The video window is relatively small, therefore, head and shoulders can be seen, but nothing of the background environment which is visible in a face to face meeting. The frame refresh rate was at times slow and "blocky"; therefore, the quality of the image was poor.

The lack of any formal protocols, e.g. which reflector to connect to, who creates a new workbook on the whiteboard, who should begin talking, drawing etc., resulted on many occasions of no productive work. Each student waited for the others to begin working. On occasions, a student would take control and this eased the problem, but this assumption of control was dependent on the student's personality.

3.2.2 JTAP Phase 2: Sub-projects

The second stage of the JTAP project looked at tasks running for longer periods using the infrastructure already in place at each site from the earlier pilot exercises. Three collaborative sub-projects were drawn up that would be the focus of third year final projects in CSCW.

These students were divided into six groups of three students who worked collaboratively on the sub-project. The sub-project's focus was to experience and evaluate CSCW technologies. Each site supervised two groups undertaking their sub-project. This was consistent with the main objectives of the JTAP project proposal, and this phase followed on from the self contained phase 1 tasks. This work is described in more detail in Chapter 5, with results presented and discussed in chapter 6.

The results from the first year of the JTAP project - Phase 1 pilot exercises (Layzell, Macaulay et al. 1997) were the basis for the local adoption and diffusion of both synchronous and asynchronous technology within Durham's Computer Science department's Software Engineering Groups (SEG) project work.

3.3 Software Engineering Group (SEG)

Software engineering group projects have run successfully since 1984 within the Department of Computer Science at the University of Durham (Drummond, Boldyreff et al. 1997). This type of project presents the first opportunity for the student to work as part of a group, to divide up work among several team members and make technical decisions as a group - a not uncommon real-life parallel. The deliverables, which take the form of reports associated with various phases of the SEG project are prepared as electronic documents, and since 1996, these have been written in HTML and submitted via the World-Wide Web (WWW). The SEG project lifecycle follows the classic waterfall lifecycle model with some modifications (Figure 3-1).

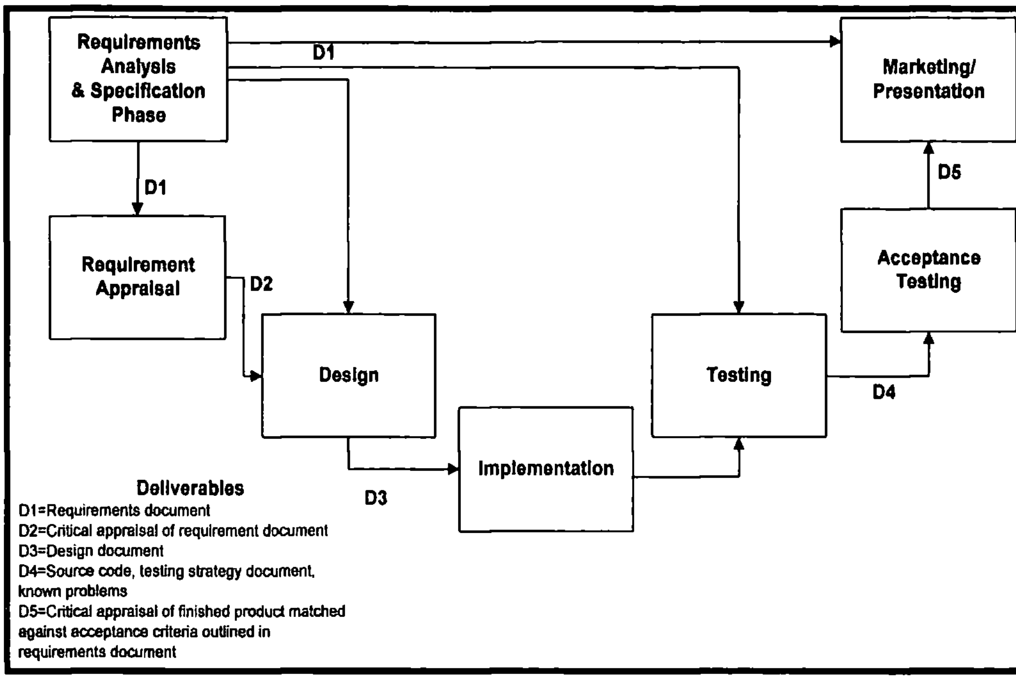

Figure 3-1: SEG project phases

Use of the WWW offers major advantages such as providing a visible and easily accessible record to the groups work and also allowing the group to explicitly link their earlier work to their later development and testing work.

The project itself is well structured into phases, with a series of strict deadlines which must be met. Each phase has a resulting deliverable which has a clearly defined structure, i.e. specific section headings and a prescribed number of pages. As shown in Figure 3-1, these deliverables feed into the subsequent phases. The phases are:

- phase 1 - requirements capture with a resulting requirements specification
- phase 2 - appraisal of the requirement specification by another group.
- phase 3 - design phase
- phase 4 - implementation and testing
- phase 5 - acceptance phase
- phase 6 - group presentation of their final system to members of the academic staff.

Having successfully established and run SEG projects at Durham for a number of years, the department had begun to consider how these could be improved to more realistically mirror industrial practice of distributed software engineering. In particular, these considerations have given rise to studies to identify appropriate groupware technology for supporting SEG projects, and the development of a virtual software engineering environment to specifically support the software engineering tasks performed by students in groups.

In 1996-97, all SEG work by students was developed and delivered as WWW-based documents. Experiences with using the WWW in this way have been very positive and the quality of student work improved dramatically in its presentation. With the addition of a suitable Web-based shared workspace, it was anticipated that students would be able to coordinate their project work more effectively.

By its very nature the SEG project is an exercise in cooperation among the students working in a group[10]. The newly introduced practical exercises i.e. Introduction to DVC, shared workspace tutorial, specifically focus on introducing the students to computer-support, both asynchronous and synchronous.

3.4 Workgroups Summary

For both JTAP and SEG, there was a need to provide both technological and social infrastructures to support cooperative work. There is also a need to understand and address both group organisation issues, i.e. how to organise and manage the group, and work organisation.

Three workgroups have been described: distributed students undertaking small collaborative pilot exercises (JTAP1), distributed students undertaking a more

[10] SEG is part of the Software Engineering I module. Students are expected to spend 14 hours per week on this module. There are 2 hours of lectures and 4 hours of timetabled practicals.

prolonged collaborative exercise (JTAP2), and finally collocated SEG students based at Durham.

Preliminary results from the pilot exercises have initially shown there is a lack of functionality in the groupware tools used, for software engineering work. This deficiency has resulted in a set of requirements that need to be addressed in future groupware use.

3.5 Requirements of Software Engineering groups

There is the need to create virtual team environments that replace face-to-face relationships which are not possible due to the geographical locations of team members. This environment can be achieved, in part, by the use of synchronous communications such as DVC described. Whilst the pilot studies only used synchronous technologies, it is generally the case that team members often work asynchronously (off line), therefore needing additional support in the form of an asynchronous shared workspace. This workspace must provide a means for fostering team activities by providing a platform of services that supports interpersonal relationships and maintains group awareness of the project status.

Based on the initial results of the pilot exercises, and the use made of BSCW by the JTAP project members, it was possible to determine a set of requirements for future collaborative software engineering and software engineering student work. These requirements whilst not comprehensive acted as the basis for continuing work detailed in the following chapters.

The requirements are as follows:

- *Configuration*: A common set of collaborative technology both hardware and software is required to support both synchronous and asynchronous working in the form of DVC and shared workspaces.

- *Technical Support*: A 'technical' groupware facilitator is needed who can set up the hardware and software infrastructure, organise and manage its use and monitor usage. This support should be ongoing.

- *Familiarisation*: A period of familiarisation with the technology and group working concept is important for all users. Social protocols for group behavior need to be established. Group familiarisation at the onset of the task is important as a group at this early stage will lack cohesiveness and therefore find interaction difficult. An 'ice breaking' session is necessary.

- *Training*: Training is an essential factor for the effective use of the technologies and groupware chosen. Groups need to learn how to cooperate using the technology. Onsite training and training documentation must be available to provide this.

- *Motivation*: It is necessary for users to be motivated to use the technology. These users need to understand that their contribution is necessary to the ongoing advances in distributed working for software engineers.

3.6 Summary

In this chapter, the background and objectives for both the JTAP project and SEG work have been described. To support student group working, a basic toolset was provided. This toolset allowed initial pilot case studies using synchronous technologies to be undertaken. Results obtained from the pilot exercises and the use of BSCW for the management of the JTAP project, have provided the experience and infrastructure for this work to be carried forward for both Phase 2 of the JTAP work, and the introduction of these technologies into the SEG work. Results from the pilot exercises have also been the basis for developing a set of user requirements for software engineering groupwork. The next chapter discusses the integration of BSCW into the SEG work at Durham.

Chapter 4

Developing a Virtual Environment

This chapter first describes the rationale for the change in the computer support for SEG work based at Durham. This change involves the introduction of a virtual environment in the form of a shared workspace - *SEGWorld*. Section 4.2 discusses the development of *SEGWorld* and the design issues and decisions made. Section 4.3 details the design of *SEGWorld*. Section 4.4 briefly outlines the use made of BSCW for Phase2 of the JTAP project.

4.1 Rationale for change

The structure of SEG (Software Engineering Group) work has been well documented. The stability of the SEG work over the past six years, the careful management, continued monitoring and the experience gained by its developers have led it readily into the next phase of change. This change has been to provide more flexible support for group working for both students and their supervisors. This has been achieved by utilising the resources of the WWW and introducing new technologies such as DVC, and, more importantly, because of the scheduled prolonged nature of SEG projects, the asynchronous tool BSCW.

4.2 *SEGWorld*

The *SEGWorld* project is complementary to the JTAP project and seeks to apply JTAP results suitably adapted for locally based SEG work through the development of a WWW based system. The virtual environment *SEGWorld*[11] has

[11] *SEGWorld* refers to the shared workspace environment created and adapted using the BSCW system.

been developed by tailoring the existing and freely available Basic Support for Cooperative Work (BSCW) system.

To further develop SEG work, *SEGWorld* based on BSCW has been developed. *SEGWorld* essentially provides a repository for all the relevant teaching materials associated with SEG projects as well as facilities for posting notices to students, and providing access to software tools relevant to student project work and their associated software engineering practical sessions. *SEGWorld* is a public area with access granted to all involved in SEG. In addition to this public workspace, it was essential to provide each group with its own private group workspace where their work could be stored securely with access granted to themselves, their tutor and the *SEGWorld* Administrator (Figure 4-1).

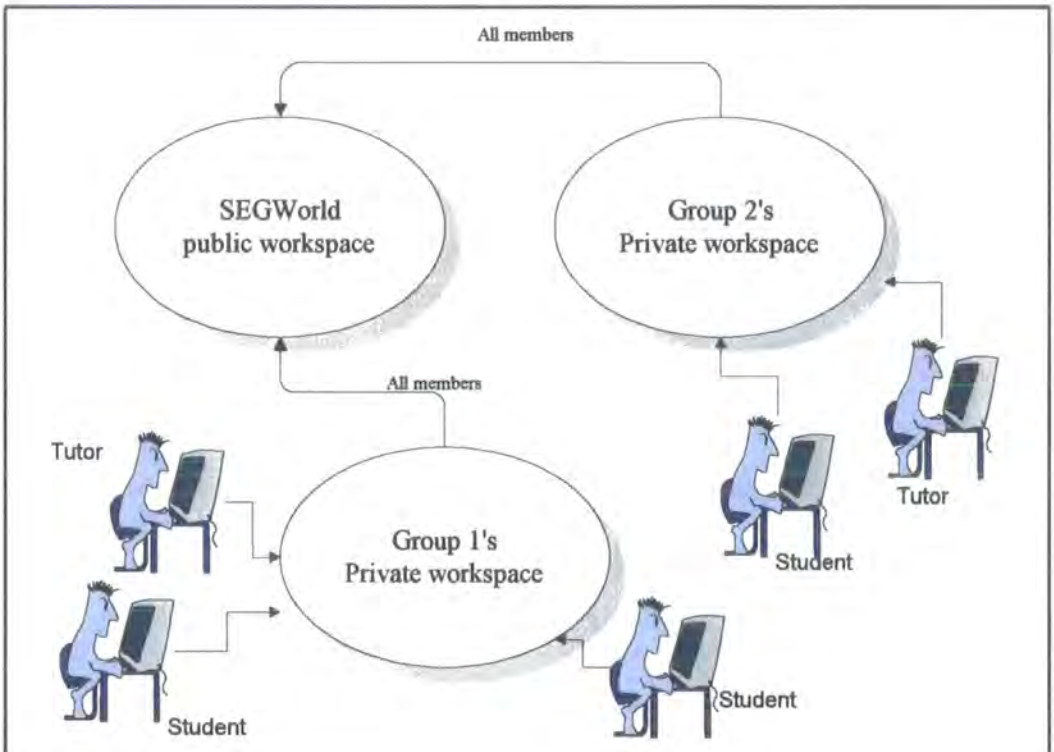

Figure 4-1: Overview of the private and public workspace

4.2.1 Design Issues and Decisions

In the design of *SEGWorld*[12] various questions and the issues involved were considered, these being:

1. What should be automated?
2. Should all phases of the lifecycle be included in the structure of *SEGWorld*?
3. Students should see *SEGWorld* as being supportive of their work.
4. *SEGWorld* needs to be adaptable for growth.
5. *SEGWorld* should be easy to use.
6. Visibility of student activities is important for team members and academic staff.
7. Training in the use of the system is important

In response to the first two questions, it was considered necessary to first automate the entry of a large number of 2nd year undergraduate students and tutor details into the BSCW database. In addition to this, each student requires a *SEGWorld* user ID and password. Manual entry of this data and generation of userid's and passwords would be error-prone and time-consuming.

Secondly, a design decision was made not to use *SEGWorld* for the development and subsequent versioning of the Modula-2 code which is used within the implementation phase of the SEG work. It was decided that student use of the shared workspace should be confined to SEG documentation, not source code, as it was felt that the students would otherwise be exposed to, too many changes. It is anticipated that in the coming academic year 1999/2000 the implementation phase of the lifecycle will be incorporated into the workspace.

It was recognised as being significant, that the students should see the shared workspace as being supportive of their work. Therefore it is important factor that the system should be easy to use and not hinder their normal progress. This could

[12] The design details of SEGWorld include the design of the public and private workspaces.

be achieved by ensuring that the students had a full understanding of the concept and functionality of the workspace and as a consequence the development of training materials, i.e. a *SEGWorld* tutorial, was vital.

4.3 Design of SEGWorld

Figure 4-2 below, proposes a process development and usage scenario depicting the sequence of events in developing *SEGWorld*. This process development is adapted from work undertaken by Christie (Christie 1995). This process lifecycle model is analogous to the software development lifecycle, for example step 1 is associated with requirements definition, step 2 and 3 are associated with design and testing phases, step 4 can be associated with coding while step 5 and 6 support deployment of the process.

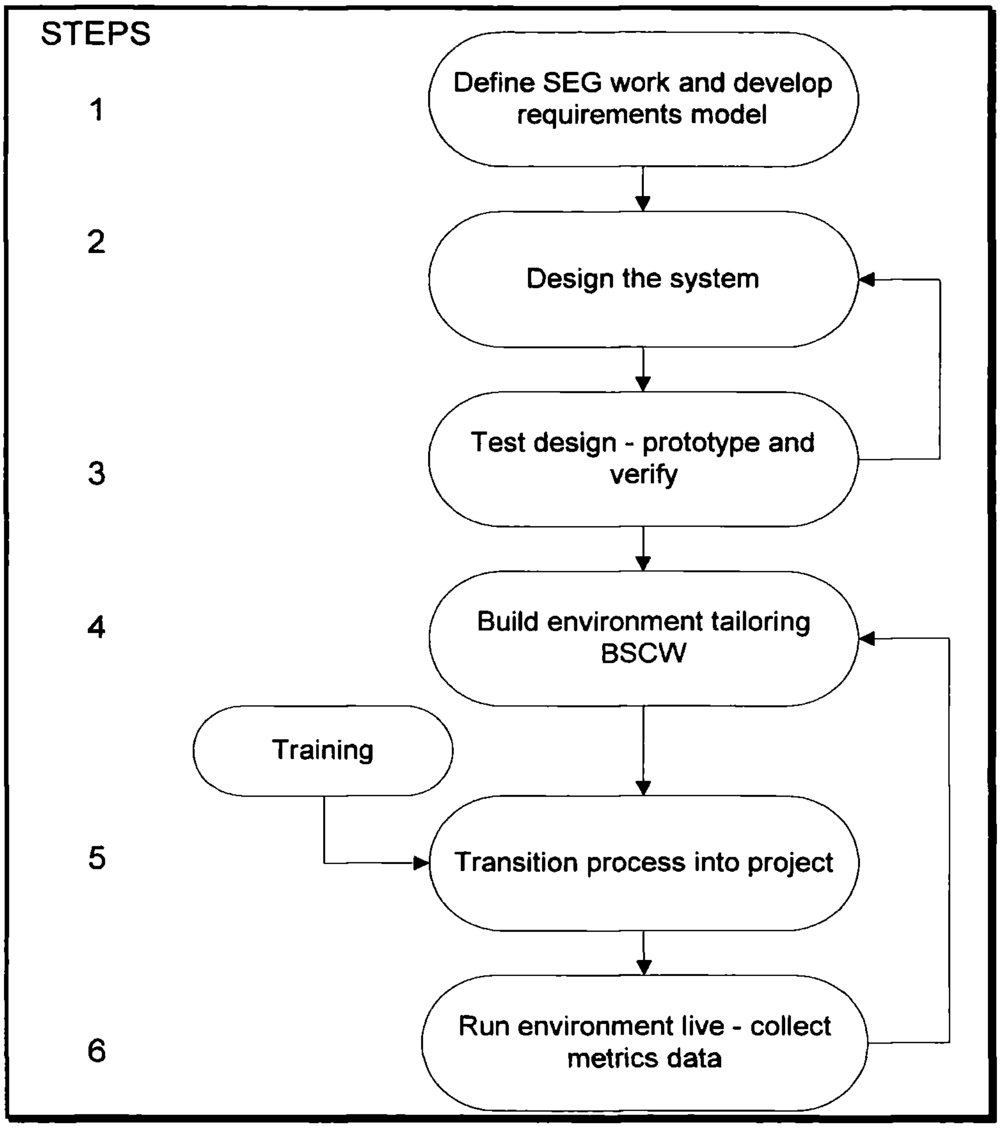

Figure 4-2: Process development and usage scenario of SEGWorld

In many cases requirements capture and analysis, in a maintenance environment, is carried out because there is a problem with an existing system. This, in itself, can be problematic as it is very difficult to be definitive about a problem specification. For example, in a diverse user community there may be differing requirements and priorities (Sommerville 1995). In the design of *SEGWorld,* the student user group structure and aims of the project remain the same each year (the project subject changes annually but the structure remains the same), and as such, the student priorities rarely change. Therefore, the requirements were already stable and well documented (Drummond, Boldyreff et al. 1997)

The process development scenario (Figure 4-2) provided an overview of the development of *SEGWorld* showing the steps involved. In Table 4-1 below, the objectives and activities involved within each step of the development scenario are defined.

For example, within step 1, the objectives were to define the current practices and structure of SEG, and, in addition, to determine what functionality should be provided by BSCW. The activities column defines how these objectives have been met.

STEP	Objectives	Activities
1	Define SEG work	• Present format well defined
	Develop requirements model	• Central repository providing document management for group work and electronic hand-in point • Security for groups' work • Secure recording of student deliverables for marking • WWW access • Platform independence • Version control • Speed of response • Monitoring of student activity • Student awareness of other group members' activities
2	Design the system	• Determine public and private workspace contents (see Figure 4-3) • Resolve access permissions to these areas
3	Test design - prototype and verify	• Tailor BSCW to match design using test data • Test functionality i.e. uploading, copying, version control etc. • Test access permissions
4	Build environment tailoring BSCW	• Develop and run Perl scripts which automate the addition of users into the BSCW system • Email all members of BSCW system with automatically generated random user passwords and system username • Manually allocate users to correct workspace
5	Transition process	• Develop training material/tutorial

	into project	• Undertake student training sessions • Alter existing SEG phase documentation to support activities in environment
6	Run environment live - collect metrics data	• Monitor use of system by each group, using questionnaire, focus groups, and automatic daily activity log generated by BSCW. • Monitor how often workspace is accessed during the different software lifecycle phases • Determine the extent of contributions made by members of the group • Determine the use made of functions provided by BSCW for further administration

Table 4-1: Activities in the development of SEGWorld

Step 2 of the development process determined the contents of the public and private workspaces. The contents are shown in Figure 4-3. This hierarchical structure allows for the addition of other work, for example, the storage of the group's software engineering practical exercise deliverables.

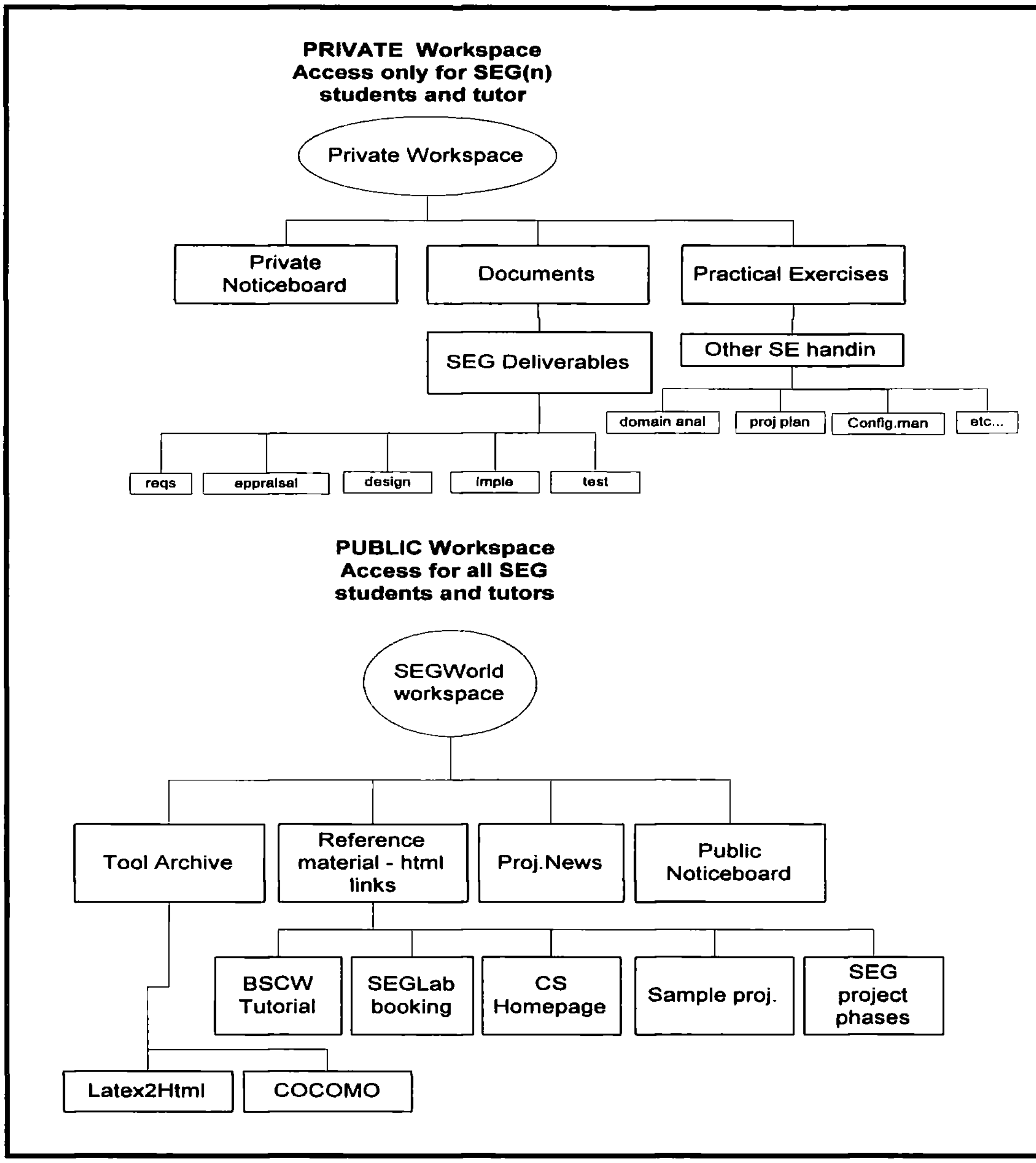

Figure 4-3: Public and private views of the SEGWorld environment

The design of the *SEGWorld* environment, in part, mirrored the hierarchical file system structure that had been in existence on the Novell system used by the students of previous years. BSCW offers a hierarchical file system structure similar to this with folders and sub-folders replacing directories and sub-directories (Figure 4-4). These similarities were an important factor for the integration of the system into the SEG work.

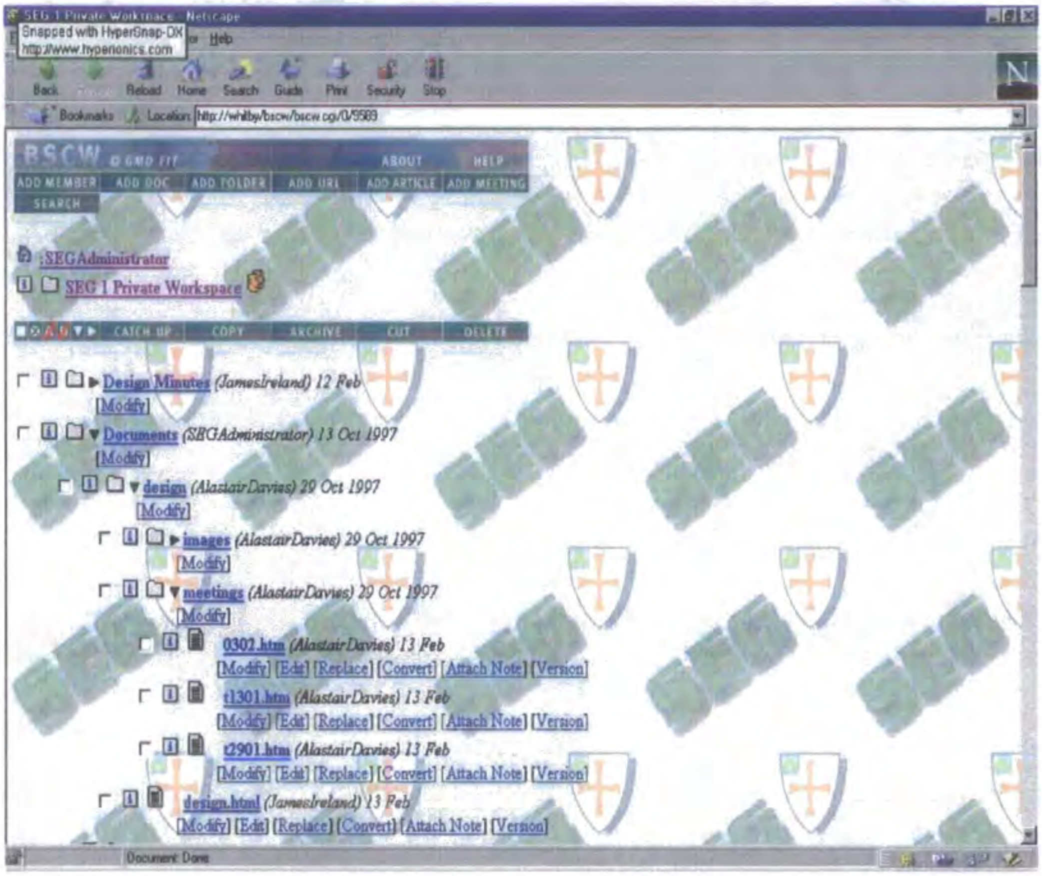

Figure 4-1: Example of *SEGWorld* folder structure

Development of a virtual environment for JTAP Phase 2

The work that has been described in the previous sections, relates to the collocated work within the SEG projects at Durham, and the network support provided. In addition to SEGs having the shared workspace provided for their group work, it was also necessary to provide this same support for the JTAP2 students (briefly described in chapter 3 section 3.2.2).

These students carried out cooperative work across sites and therefore required both synchronous and asynchronous communication. Synchronous communication was provided by DVC, and the asynchronous communication was via a shared workspace - BSCW. It was important to provide a central repository for the distributed students as they needed to have some shared area where work in progress was visible and easily accessible. In addition to this, the provision for monitoring each other's activities, e.g. determining who has read the most recent version of a document, was an important factor.

The structure of the JTAP2 students workspace was similar to that of the SEGs. Each JTAP2 group had their own workspace, and within these areas, these students were allowed to structure their work to suit their method of working.

4.5 Summary

This chapter has described the development of a virtual environment in the form of a shared workspace for use by both the SEGs - *SEGWorld,* and JTAP2 students, to support their group working. The next chapter will describe the structure of the case studies used which involved both collocated and distributed student groups, using both shared workspaces and DVC.

Chapter 5

Case Studies

This chapter describes the case studies undertaken within this research. An initial overview is provided in section 5.2 with section 5.3 describing the data collection methods used. Section 5.4 describes the laboratory facilities available to the student groups. Sections 5.5. and 5.6 describe the JTAP and SEG case studies respectively. For each case study, the objectives, task, subjects, setting, procedure and methods of data collection are stated.

5.1 Introduction

The objective of undertaking a series of case studies was primarily to determine the usability of the technologies and associated groupware tools applied to software engineering groupwork.

These case studies involved evaluating the shared workspace *SEGWorld,* specifically developed for the SEGs, and the use of a shared workspace in general for distributed students. In addition to the student use of asynchronous technology, it was equally important to expose all students involved to synchronous technologies - DVC. The data collected from these case studies will be used to prove or disprove the hypotheses outlined in Chapter 1.

5.2 Overview

Table 5-1 presents a summary of the case studies and shows the mode of communication, student group location, timescales and method of data collection, used in the case studies. As shown, there are many variables associated with the case studies, such as, the collocated students being 2nd years whilst the distributed students are 3rd years. Some case studies are short whilst others are carried out over the academic year.

Mode of Communication	Student Group Location	Timescale	Data Collection
Experiment Set 1			
a) Synchronous (DVC) Asynchronous (BSCW)	Distributed - JTAP phase 2 (3 sites)	7 weeks	• Questionnaire • Focus Group
Experiment Set 2			
b) Synchronous (DVC)	Collocated - SEG (Durham site)	2 x 2 hr sessions	• Questionnaire • Observation
c) Asynchronous (BSCW)	Collocated - SEG (Durham site)	15 weeks	• Questionnaire • Activity Log • Focus Group

Table 5-1: Summary of case studies

5.3 Data collection methods

There were four main methods of data collection used to varying degrees for each case study. These methods are as follows:

1. Questionnaire — Three questionnaires relating to DVC: the first for personal details, the second for ascertaining the level of computer skills and the third to determine the usability of the technologies from the view point of suitability to task, suitability to environment, i.e. what software engineering activity was suited to the use of audio (Appendix A).

   A further on-line questionnaire relating to the use of BSCW, was developed specifically for SEG students (Appendix B).

2. Activity log — A daily workspace activity report provided an opportunity to keep track of what was happening within the SEG workspaces. The report was compiled nightly and sent to the *SEGWorld* Administrator (the author). The report lists the previous day's activities sorted by workspace. In the case of the JTAP2 students where the BSCW server was maintained at Keele, a manual count of activities was undertaken.

3. Focus groups — At the end of both the SEG and JTAP2 projects, group meetings were held. These meetings took the form of informal group discussions with individual groups. The aim of these sessions was to provide the students with the opportunity to discuss openly the successes or problems encountered within their work, and to discuss how improvements could be made. The discussion topics included BSCW, DVC, social aspects of this type of working, and general hardware and software problems encountered.

4. Observations — Observations were made of both JTAP1 pilot exercises and SEGs undertaking the DVC tasks. This exercise was primarily to observe the students' use of the technologies and how they overcome problems if any occurred.

5.4 Laboratory Facilities

As previously mentioned in chapter 3, section 3.2.1.4, a small collaborative laboratory housing three multimedia PC's was set up at each university to support the distributed collaborative work. At Durham an additional larger laboratory was setup specifically for SEG use. This SEGLab contains twelve multimedia PCs similar in specification to the smaller laboratory. Prior to the SEGLab, the SEGs had access to campus PCs in open access areas. It was seen as important that SEGs should have an area specifically designated for them where they could work and meet as a group.

The following sections detail the case studies undertaken for both collocated and distributed student groups.

5.5 JTAP Case Study - Phase 2

5.5.1 Experiment Set 1-a: DVC and BSCW

In the first year of the JTAP project, single session on-line conferencing tasks were run for group of three students (described in chapter 3 section 3.2.1). In the second year, the project progressed to looking at longer term examples of distributed group working.

University of Durham regulations do not currently permit students to undertake projects having a project supervisor from another university. These longer term projects were therefore designed so that the students did a main project in CSCW at their own university - focusing on an area of interest, e.g. evaluation of groupware. During the course of their main project, these students also took part in a collaborative "sub-project" with set requirements and deliverables.

5.5.1.1 Objectives

The objectives of this second phase were to run live parallel tasks to promote the concept of distributed group projects and, therefore:

- to give students experience of working collaboratively in a geographically distributed team, using group working technology, and
- to allow students to evaluate these technologies against software engineering needs for group working

5.5.1.2 Task

There were three customers who specified tasks to be undertaken. The tasks were Comparative Shopping Monitor (Durham), Departmental Room Booking System (Keele), Student Accommodation Management system (UMIST).

These collaborative sub-projects concerned the specification, design, implementation and presentation of a database. The projects followed through the full software development life cycle in order to provide experience of the different types of collaborative working.

5.5.1.3 Subjects

Each university offered three projects to third year computer science students. The projects were all run twice, and were designed for groups of three collaborating students - 1 from each university, giving 18 students in total.

5.5.1.4 Setting

For the sub-project, groupware, e.g. CU-SeeMe, Netmeeting, Lotus Notes™, BSCW were made available for student use. Members of the groups choose which applications they wished to use. The only stipulation was that the systems were to be built using MS Access, documentation in Word 95 and deliverables were to be maintained and stored in the BSCW workspace.

Each site took responsibility for one of the specified tasks and supervised the two groups undertaking that particular task as well as providing technical support on site for their local students undertaking the other two tasks.

5.5.1.5 Procedure

The sub-projects were to be undertaken over a period of seven weeks. This time span was comprised of a two week period for requirements gathering followed by a five week period for design and implementation.

It was decided by the universities involved that before actual sub-project work began, it would be advantageous for the students to meet their peers and the "customer" face-to-face and elicit information from their "customer". The benefits of this would be to allow the groups to begin their requirements gathering being aware of each others' strengths and weakness and already knowing each other on a social basis. In a study undertaken by Olson and Teasley (Olson and Teasley 1996) it was found that even though their study groups were provided with technologies which allowed them to meet on-line, the group members felt that for collaborative activity to be successful it was important to understand what each other knew, and to build a basis for trust and commitment. Therefore meeting in person and sharing more than work tasks was crucial.

This meeting took place at a single location, and the students were briefed about the aims, objectives and their role in the sub-projects.

The two week requirements phase consisted of the groups developing a requirements specification and where possible checking this with the customer. This phase was undertaken in the latter part of the Christmas term.

The five week design and implementation phase consisted of the design and development of a database based on customer requirements. This phase was undertaken mid-way through the Easter term. Because of the length of time between the end of the requirements phase and the beginning of the design phase, it was felt that once again a face-to-face session should be held to reinforce group bonding.

5.5.1.6 Data collection methods

The following data collection methods were employed in this case study:

- questionnaire, and
- focus group meeting (with locally based students only).

5.6 SEG Case Studies

5.6.1 Experiment Set 2-b: SEG - DVC

This case study was a re-run of one of the pilot exercises that has been described in chapter 3. The main aims of this case study were to allow the students experience of using DVC technology and to elicit their reactions and thoughts on this method of working for software engineering activities.

5.6.1.1 Objectives

1. To introduce to the students the concept and practical experience of synchronous collaborative working
2. To determine student satisfaction with the process
3. To determine the usability of the technology

5.6.1.2 Task

The students had to establish and prioritise features of a system to automate the accounts of a small bookshop. The members of a group were given different paper based descriptions of the manual accounts system and were asked to discuss the case study information and to produce, as a group, a simple diagrammatic representation of the manual system. Subsequently, the group were required to identify possible features of an automated system and to prioritise these feature into three releases (with the most important features in the first release).

5.6.1.3 Subjects

In 1997-98, the subjects were nine SEGs of six students and one SEG of seven students. These were the whole of the 2nd year computer science undergraduate intake. All 2nd year students must take the Software Engineering I module which includes the SEG project. In all the groups the students had not had any previous groupworking experience.

5.6.1.4 Setting

At each video conferencing session, two groups of students were accommodated. The work was carried out in the SEGLab in which desktop video conferencing equipment (PC based) was available to each student. The room was arranged in four rows, each row housing three PC's. The rows were divided off from each other by the use of moveable screens. The students were allocated to the PCs with the intention of splitting members within each group.

5.6.1.5 Procedure

Each group undertook 2 x 2 hours video conferencing session over a two week period.

The first session was primarily for the students to familiarise themselves with the technology and its capabilities, and for the students to work as a group for the first time. This session began with a short introduction by the author who outlined the aims and objectives of collaborative work, both collocated and distributed, and explained how it was to be facilitated by the video conferencing tools available to the students. Each group then experimented at will with the equipment. A background questionnaire was completed by every member to determine technical competence and past experience (if any) in any form of collaborative working which may have influenced their performance.

The second session comprised of a structured collaborative software engineering task that was to be undertaken. The session was broken down into four 20 minute slots followed by a slot to complete an evaluation questionnaire.

These slots are further described below:

1. Introduction, briefing: how the session was expected to progress
2. Information sharing: this involved a discussion of the information in the case study and the construction of a simple model representing information of the firm's existing manual system.
3. Brainstorming description: this involved using the case study and the model to discuss and list all possible features that the new automated system would have.
4. Brainstorming prioritisation: this involved the prioritisation of the main features from the previous state into three lists, these lists to correspond to the three release versions of the final automated system
5. Evaluation questionnaire: this involved the completion of a questionnaire by every member of the group.

The total time allocated for the task execution and for the subsequent completion of the evaluation form was 1 hour 40 minutes.

These sub tasks within the slots, are undertaken using the technology provided i.e. DVC. Students within each group were discouraged from talking to each other face to face.

5.6.1.6 Data collection methods

The data collection methods employed in this case study are:

- observation,
- questionnaire, and
- focus group meeting.

5.6.2 Experiment Set 2-c: SEG - BSCW

The aim of this case study was to investigate appropriate support for collaboration and coordination between SEG members by monitoring practical usage of long term asynchronous collaborative working project support.

5.6.2.1 Objectives

- to introduce the students to asynchronous group working technology
- to determine the feasibility of using such technologies within software engineering

5.6.2.2 Task

Each SEG had to implement an electronic voting system. The main function of the system would be to allow election of student representatives to the Staff and Student Committee. As well as handling the election of student representatives, the voting system must also support voting on specific issues, i.e. referenda. It must be possible to restrict voting within the system to specific periods of time, e.g. a specific day or between specified dates.

The system must also be secure so that votes cannot be tampered with once they are cast. It must not allow multiple voting by the same individual, and it must record an audit of voters (but not their actual votes) for inspection by a super user.

The development of the voting system was undertaken over the period of the academic year 1997-98 and followed the phases discussed in Chapter 3.

5.6.2.3 Subjects

The subjects were nine SEGs of six students and one SEG of seven students (as described in section 5.6.1.3). In addition to these groups there were two other groups; one group of four and another of five, who were Natural Science and Joint Honours students. In the case of the Natural Science/Joint Honour students, the system requirements were narrowed.

In all of the groups, none of the students had worked together in class or on work experience projects.

5.6.2.4 Settings

The SEGLab as described in section 5.4 was made available to each group. An automated web based room booking system was used. As BSCW is accessible from any platform running a WWW browser, the students could access *SEGWorld* from any system connected to the university network.

5.6.2.5 Procedure

The SEG project is well structured into phases of the software lifecycle. This is documented in chapter 3 section 3.3. At the end of each phase there is a deliverable e.g. requirements specification, design document. Each group had to develop this deliverable using the shared workspace and ensure that at the deadline the document was correctly named in the correct folder e.g. in the case of the design deliverable, a folder called *Design* had to contain the *design.html* file and any associated *images*.

5.6.2.6 Data collection methods

The data collection methods employed for this case study are:

- questionnaire,
- daily activity log, and
- focus group meeting.

5.7 Summary

This chapter has described the different case studies which have been undertaken within this research. The first case study involved a small number of distributed students undertaking a project encompassing the phases of the software lifecycle. This seven week project allowed these students to experience and evaluate a variety of technologies specifically developed to support groupworking.

The second case study looked at local SEG work involving a short DVC session. This session provided local students with the opportunity to gain experience of synchronous technologies. Thirdly, a longitudinal study was undertaken, once again involving the SEGs, with the emphasis being placed on the use of an asynchronous shared workspace - *SEGWorld*.

The following chapter presents the results obtained from these case studies.

Chapter 6

Results

This chapter is concerned with presenting the results of the case studies (described in Chapter 5) and associated discussions. Section 6.1 re-introduces the hypotheses upon which the discussion of results is based. Sections 6.2 to 6.5 describes each hypothesis in turn, and presents the analysis of results from empirical data and student comments collected via questionnaires, observations and focus group meetings. For each hypothesis, the student workgroups, methods of data collection used, results and short conclusion are provided. Section 6.6 concludes this chapter with an overview of all results reported in this work.

6.1 Introduction

The results for each hypothesis are based on a combination of the responses of the student groups: SEG, JTAP1[13] and JTAP2[14]. Data collected for analysis has been via questionnaires, observations and focus group meetings. These data collection methods have been described in detail in chapter 5, section 5.3.

The hypotheses presented and discussed in the following sections are:

Hypothesis 1. The introduction of an asynchronous shared workspace into software engineering groupworking will aid group members in organising and coordinating their work.

[13] JTAP1 refers to students who undertook distributed pilot exercises which are described in Chapter 3 section 3.2.1.

[14] JTAP2 refers to students who undertook a prolonged distributed project as part of their final year dissertation which is described in Chapter 5 section 5.6.

Hypothesis 2. Greater use of shared workspace functionality will be made as the project progresses.

Hypothesis 3. Students undertake more collaboration in the earlier stages of the software lifecycle.

Hypothesis 4. Synchronous communication has an important role to play in both collocated and distributed software engineering groupwork.

6.2 Hypothesis 1

The introduction of a asynchronous shared workspace into software engineering groupworking will aid group members to organise and coordinate their work.

6.2.1 Subjects and data collection methods

Student workgroups	*Method of data collection*
JTAP 2 (3rd year distributed students)	• Questionnaire • Focus group meeting
SEG	• *SEGWorld* daily activity log • Questionnaire • Focus group meeting

In order to determine if this hypothesis was true, specific questions relating to coordination and organisation of work, via a questionnaire, were asked. In addition to this, observations were recorded and automated BSCW daily activity logs (for SEG only)[15] were analysed. These data are reported upon in the following sections.

[15] The BSCW server used for the JTAP distributed project was located and maintained at Keele University.

6.2.2 Use of Organisation and Coordination functions

From a high level perspective Figure 6-1 represents the responses from both SEG and JTAP2 students, related to the question on whether the workspace did enable better organisation and coordination of their work. From this figure, it is clearly shown that the JTAP2 students valued the use of the workspace for this purpose more than the SEG. With the JTAP2 students being distributed, the need for a central storage facility to structure their work was an important requirement; BSCW provided this. As SEGs have a group UNIX account in addition to their private workspace, five of the groups used both, with UNIX generally being the preferred choice because of quicker response times. However, it was noted that those SEG groups who used UNIX, mirrored the hierarchical structure of the workspace. Therefore, although the workspace was not their preferred choice, they choose to reflect its organisation in the UNIX filespace.

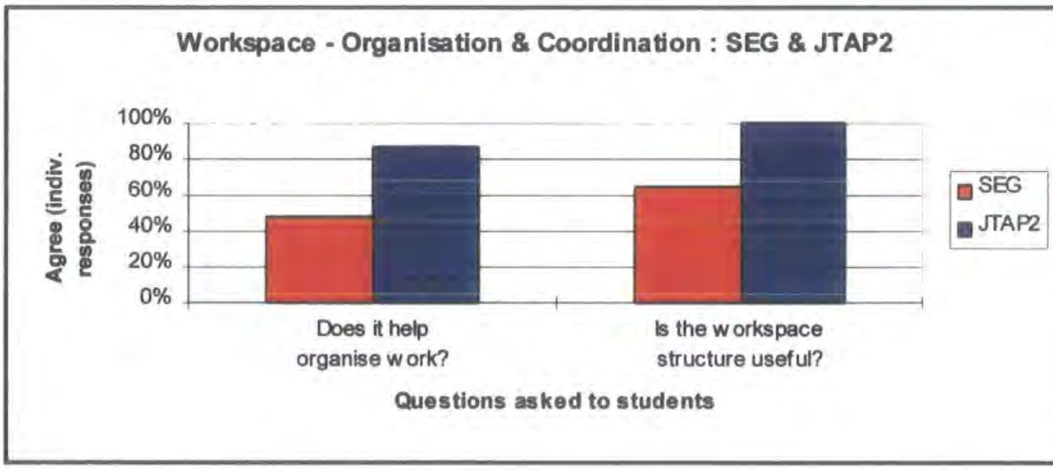

Figure 6-1: Organisation & Coordination

In general both sets of students felt that the hierarchical structure of the workspace was intuitive and graphically illustrated how their work was being structured. But, as the level of decomposition of folders (directories) into sub-folders (sub-directories) increased, navigation became slow. Students commented on the lack of shortcuts to the various documents. In fact, students were simply unaware that shortcuts are possible.

From a lower level of granularity, Figure 6-2 highlights functions provided by the system that were used during the SEG project. These functions have been chosen because they are associated with organisation and coordination. They are as follows:

- *Meeting* - schedules a new meeting showing various details and those invited to participate. An email is automatically generated to inform members of these details.
- *Versioning* - versions a document. A new version is created which becomes the current version, whilst old versions are still readily available.
- *Attached Note* - attaches a note to a specified object that is displayed to other users when they attempt to access the object. There is no formal locking of the objects, but notes may be used informally to achieve locking.
- *Catchup* - deletes event icons for the selected objects e.g. a new document has a "NEW" icon displayed when it is first uploaded into the workspace. Unless catchup is used, the "NEW" icon remains regardless of how old the object is.

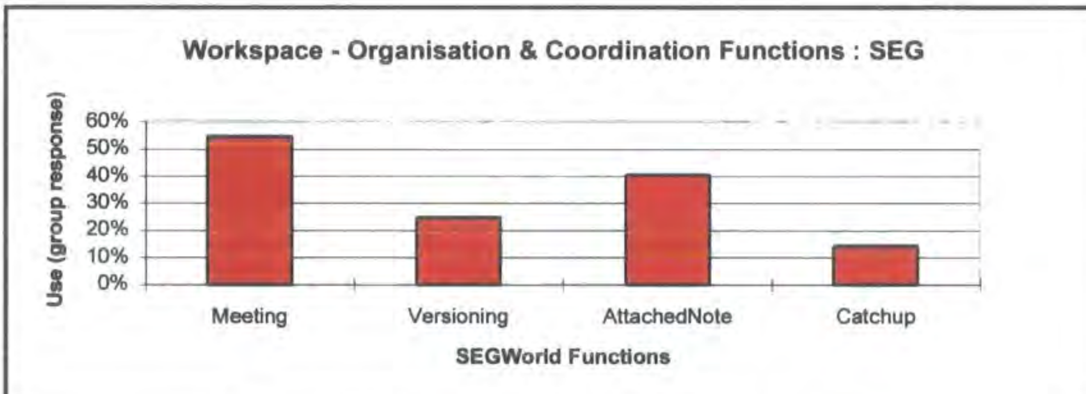

Figure 6-2: SEG Organisation & Coordination

6.2.2.1 Meeting Function

The meeting facility was used by 54% of SEGs and thought by some individual students, to be a useful function. The remainder of the students did not use it because they met face-to-face on a daily basis. In addition to this, some SEGs felt that it was simpler to use a standard email system rather than to have to load a browser, and then *SEGWorld* in order to use the meeting facility. In contrast, JTAP2 students found the meeting function extremely useful to organise future meetings for video conferencing sessions.

6.2.2.2 Versioning Function

The versioning mechanism provided was easy to apply, but only 25% of the SEGs used it. In particular within three of the groups only one member from each used this function. These individuals were, in general, the appointed group secretary whose task it was to undertake such work. In contrast to this, the JTAP2 students all used versioning to varying degrees. Initially versioning was under used, but, as the JTAP2 project progressed, the distributed groups increasingly versioned work and, in particular did so during collaborative implementation[16]. JTAP2 students commented that versioning served two main purposes: prevention of the proliferation of redundant copies of the database, and enabling other users to quickly locate the latest version of the document in the workspace.

An interesting point noted in the SEG results was that within two of the groups all members stated they had used the versioning function, but when these results were checked against the automated daily activity logs, it was found that only two members from each group were shown to have actually used the function. This anomaly may be due to members within each group being aware of the activities of each other and therefore feeling that they had also contributed to this process, or alternatively the problem could lie in inaccurate completion of questionnaire. The difficulties encountered using questionnaire data have been discussed in Chapter 3 section 3.2.1.6.

[16] This data was collected from the distributed students via the questionnaire and focus group meeting. Automatic monitoring of the BSCW server at Keele was not possible.

6.2.2.3 Attach Note Function

Whilst BSCW does not provide a locking system for documents, i.e. when someone is currently editing a document, it is possible for someone else to have access to the same document, the *Attach Note* offers the primary editor the opportunity to inform anyone else wanting this document, what the current status is. Whilst 40% of SEGs said they used this function, there was no workspace activity to support this. In the main, this function was not widely used by SEGs or JTAP2 students, simply because they did not fully understand its purpose.

6.2.2.4 Catchup Function

The *Catchup* function, which provides an up-to-date view of the activities which have occurred within the workspace was used only by 14% of SEGs. On further questioning most students admitted to not being aware of what this function actually did.

6.2.3 Hypothesis 1 Conclusion

There is strong support for this hypothesis based on data from both subject groups. The hypothesis made, was that use of a shared workspace would help in the organisation and coordination of the students groupwork. BSCW provides a central repository for all project documentation for the student groups, and allowed members to be aware of other group members activities. This awareness was seen as being important by the students as it provided them with information on the current state of a particular document. In effect what they were provided with was a simple workflow mechanism.

6.3 Hypothesis 2

Greater use of shared workspace functionality will be made as the project progresses.

6.3.1 Subjects and data collection methods

Student workgroups	*Method of data collection*
JTAP2 - 3rd year distributed students	• Questionnaire
SEG	• SEGWorld daily activity log • Questionnaire

The following graphs show the use made by SEGs, of the various functions provided by *SEGWorld*. These functions are a subset of those available and were chosen as they represented the most common events that would occur in the process of producing a document. The objective of logging the daily usage of these functions was to determine if the use of *SEGWorld* increased as the project progressed. This anticipated increase would indicate that the students had overcome any initial problems and were becoming more confident in using the workspace.

The graphs represent the average number of times each function was used by each group throughout the different phases.

In Figure 6-3 most activity is centered around creating documents and reading. The negligible amount of activity by most SEGs for the editing and versioning functions would indicate that they did not fully understand these functions. Rather than editing or versioning an existing document, it would appear that they have deleted and then re-created the document. At this early stage in the use of the workspace, this was not unexpected.

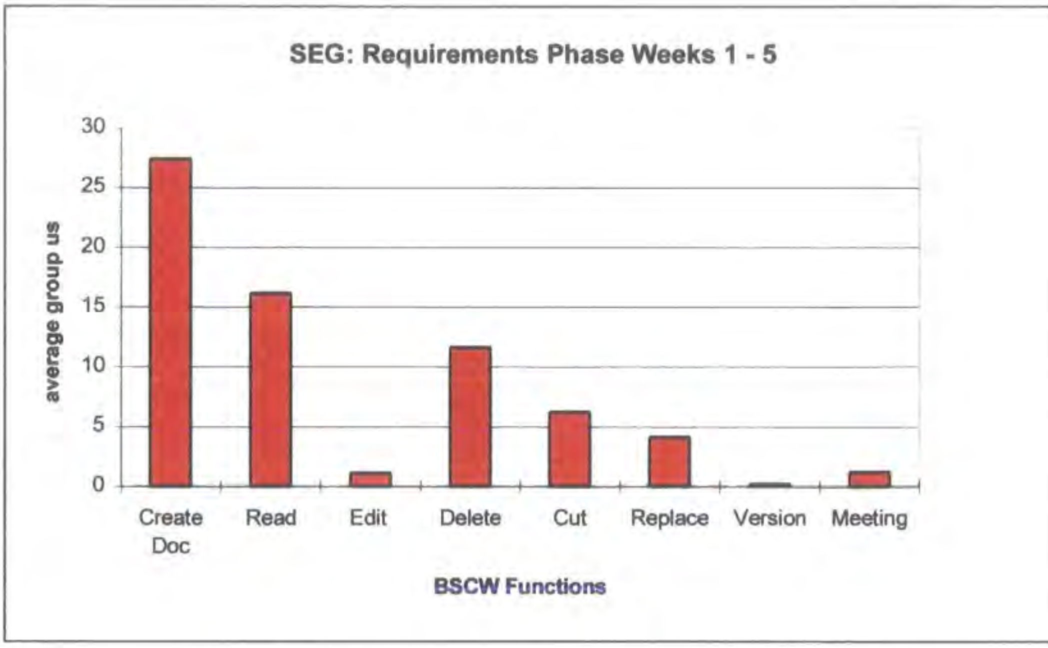

Figure 6-3: BSCW Activities - Requirements Phase

In Figure 6-4 the read activity is used to the extent that it shows on average each group has read the document eight times. It can, therefore, be inferred that most members of each group have read the appraisal document. This phase is for one week only and the deliverable is a relatively short document.

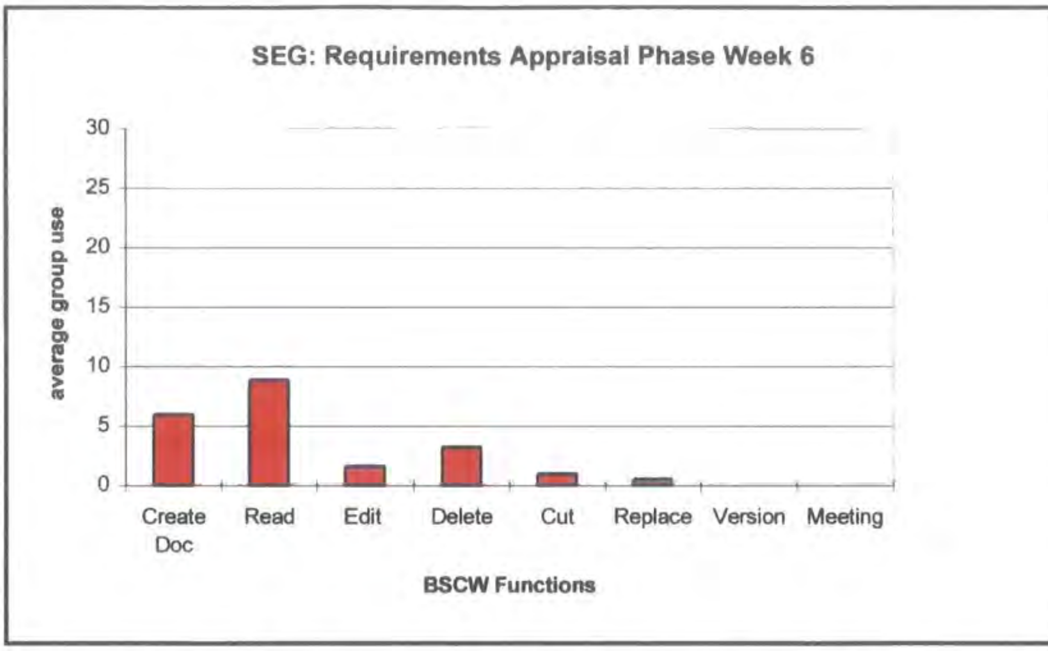

Figure 6-4: BSCW Activities - Requirements Appraisal Phase

Data from the design phase of SEG, is shown in Figure 6-5. This phase is a work intensive phase. Within this phase, it can be seen that there is a marked increase in the use of the edit function and a decrease in document creation indicating better student understanding of these functions. Whilst versioning has been used by most groups, its usage was still disappointingly low.

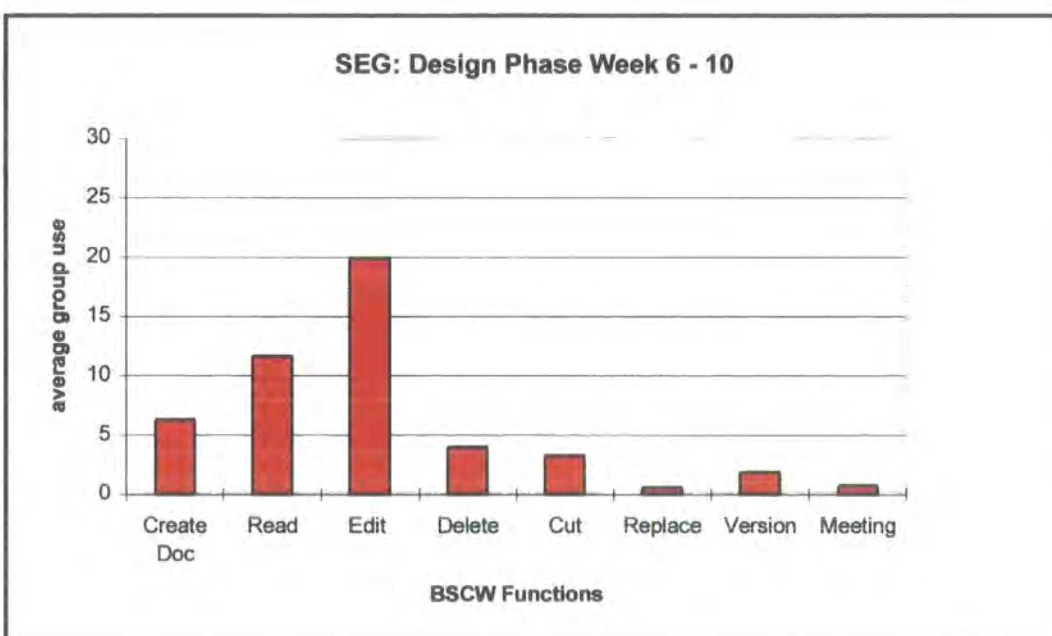

Figure 6-5: BSCW Activities - Design Phase

The implementation phase, shown in Figure 6-6, consists of developing the product software and writing a report detailing the implementation and testing strategy, and testing any know problems with the system. A decision was made at the onset of the SEG project that the use of *SEGWorld* for developing code would be inefficient. As the programming language used was Modula-2, problems were foreseen in that not only the module that the student was working on would have to be downloaded, but all the associated modules; it was felt that this imposed too high an overhead on the SEGs. The implementation was carried out on a Novell system. BSCW and other shared workspaces e.g. Lotus Notes™ are basically generic tools and as such offered SEG no specific support for software code development. In the future, the department is moving to Java as its first teaching language, it is not anticipate this will pose such a problem.

Figure 6-6 shows the development of the implementation document.

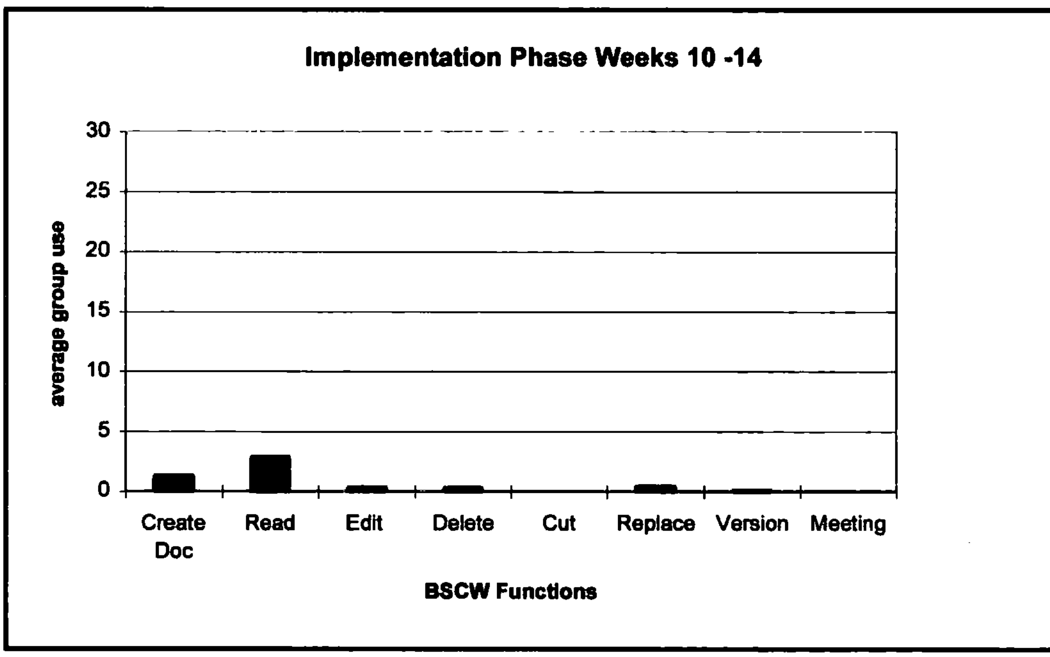

Figure 6-6: BSCW Activities - Implementation Phase

Overall, utilization of some of the more useful functions, e.g. versioning, was poor. This has been attributed to the following factors:

- students were aware of many functions but were insufficiently motivated to gain an understanding of how to use them,
- at times usage of the workspace was hampered by poor response times of the network, and
- there was a mismatch between the work in the implementation phase and the support provided by the workspace.

In addition to the factors presented above, the use of data collected from questionnaires poses a problem. An anomaly occurred within the data collected regarding the meeting function. Within hypothesis 1 (Figure 6-2) it is shown that 54% of students stated that they used the meeting function[17] but this fact is not mirrored in the phase graphs above, which represent the average number of specific activities undertaken within the workspace[18]. The assumption made to explain this anomaly is that, within each group workspace, a meeting folder was created to hold agendas and minutes of meetings. Students may have thought that this was the "meeting" function. This highlights the problem of different perceptions of a term.

Of these factors described above, the main problem that needs to be overcome is the students' understanding of the concept and function of the shared workspace. Initially, the students were introduced to *SEGWorld* via an online tutorial. This introduction was not supported by experienced demonstrators; therefore, simple problems that arose at this stage were left unsolved and many students formed a poor image of the system. This manifested itself in the lack of SEG understanding of many of the functions provided (shown in the SEG graphs previously discussed in this section). In addition to this, the poor response times of the system were a major contributing factor to the slow uptake of the continued use of *SEGWorld*.

[17] This data was from the online questionnaire completed at the end of SEG.
[18] This data was from the automatically generated daily activity log for each group workspace.

In contrast, the JTAP2 students did not have this tutorial to introduce them to BSCW, but their motivation to learn the system was greater as it was their main point of contact with the other distributed members of their group. This motivation is shown in Figure 6-7 where the time invested in learning the system is markedly different to that of the SEGs.

The problems that SEGs encountered have resulted in only 28% of SEGs who would look forward to using the system again. The JTAP2 students in contrast gave a very positive response.

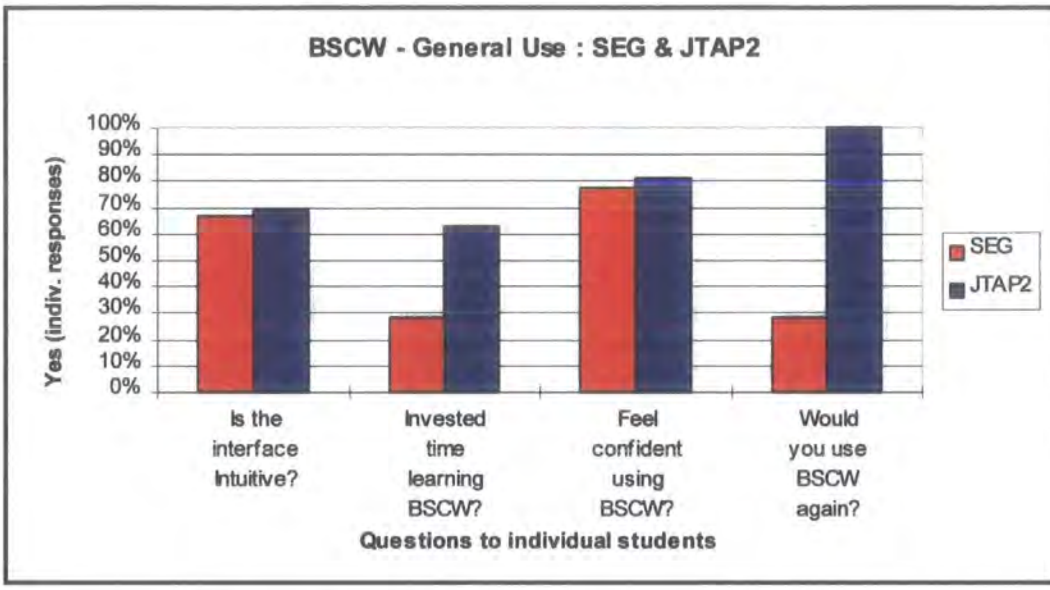

Figure 6-7: BSCW - Student General Responses

6.3.2 Hypothesis 2 Conclusion

From the SEG phase graphs, it appears that there is no significant increase in the use of functions as the project progressed. However, comparing the activities in Figure 6-5 (design) with Figure 6-3 (requirements), there is less excessive document creation and better use of the versioning and editing functions. This would indicate better acceptance of some of the more useful workspace functions, by the groups. The nature of the SEG project with two short phases (requirements appraisal and acceptance testing), and the fact that *SEGWorld* inadequately supports implementation resulted in the decrease of use in the later stages of the project.

The problem of introducing a new concept such as a shared workspace highlights that it is not just a question of giving students a new application with an associated tutorial. The shared workspace is more than just an application; it also provides awareness of other members activities, and it requires the group members to organize and coordinate their work differently. Therefore, the introduction of *SEGWorld* to SEGs in the coming academic year will be preceded by an initial lecture. The on-line tutorial has been revised to be more specific, highlighting functions that will be of particular use to the students. In addition to this, the demonstrators have become more experienced and can take a more active role in ensuring students gain a clearer understanding of the system.

6.4 Hypothesis 3

Students undertake more collaboration in the earlier stages of the software lifecycle.

6.4.1 Subjects and data collection methods

Student workgroups	***Method of data collection***
JTAP2 - 3rd year distributed students	• Questionnaire • Group workspaces • Focus group meeting • Face-to-face
SEG	• SEGWorld daily activity log • Questionnaire • Focus group meeting

In order to determine if this hypothesis was correct it was necessary to use data collected from the methods above. The main method used was the monitoring of the number of activities each group undertook within their group workspaces.

Whilst this monitoring helps to indicate where in the software lifecycle most activity has occurred, it is only able, in part, to help prove the hypothesis. The reason for this was that the workspaces used by SEGs supplemented their face-to-face meetings, whereas, in the case of the distributed students, the workspace was their main point of contact and was supplemented by the video conferencing sessions.

The following graphs represent the workspace usage during the different phases of the software lifecycle by distributed JTAP2 and SEG student groups. These results take account of the differing time scales for JTAP2 and SEG.

Time average techniques are used to remove the effect of the different time/phase allocations. Each SEG and JTAP2 phase is shown proportionally of the overall project lifecycle:

TotalActivitesPerPhase / #WeeksOfPhase

Figure 6-8 shows the activities within the JTAP2 workspaces during the different phases of the software lifecycle. These phases comprise of:

2 weeks - Requirements

2 weeks - Design

3 weeks - Implementation

Figure 6-8 indicates that 58% of activities occurred within the requirements phase. These JTAP2 results are felt to be more realistic than SEGs' result of 69% (see Figure 9) because JTAP2 groups had less opportunity for face-to-face interaction.

Face-to-face meetings were provided on two occasions for the JTAP2 students. The first meeting was at the beginning of the requirements phase, with a second meeting at the onset of the design phase. This subsequent meeting was felt necessary as there was a gap of approximately eight weeks between the end of the requirements phase and beginning of design phase. There was no face-to-face meeting at the beginning of the implementation phase[19].

The JTAP2 students supplemented the workspace with video conferencing sessions and email. Neither of these modes of communication were observed as monitoring email was not possible and the meeting times chosen by the students were random.

Although this face-to-face, video conferencing and email collaboration is not represented in Figure 6-8, the results are still a reasonable reflection of the amount of collaboration undertaken throughout the project, because the workspace was the central point of contact for all the JTAP2 groups.

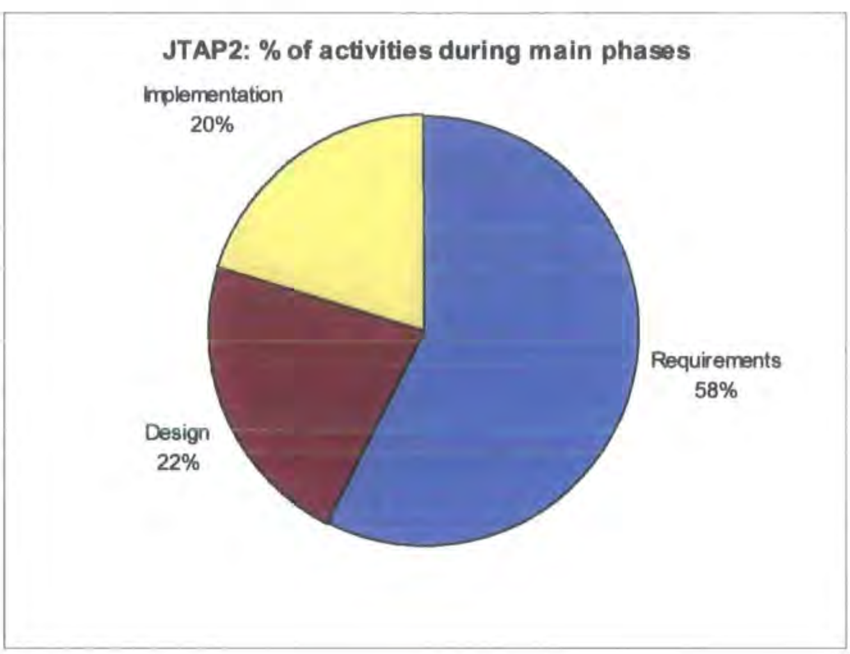

Figure 6-8: JTAP2 workspace usage during the software lifecycle

Figure 6-9 shows the SEG activities undertaken within the group workspaces during each phase of the project. SEG phases comprise of:

5 weeks - Requirements

[19] One JTAP2 group did not undertake any implementation. This is reflected in Figure 6-8.

1 week - Requirements Appraisal

4 weeks - Design

4 weeks - Implementation

1 week - Acceptance Testing

Figure 6-9 indicates that the largest proportion of workspace activities can be attributed to the early stage of the project. Sixty-nine percent of all workspace usage by SEGs occurs during the requirements phase, which combines requirements specification and requirements appraisal. Implementation activities are negligible due to the workspace being used for the development of the testing strategies report only, hence causing some distortion in the resultant graph.

What the graph does not show is the face-to-face collaboration that occurs with SEGs. This interaction is considerable as the students spent most of their academic day together and hold numerous informal meetings.

The graph shows much activity in the requirements and requirements appraisal phase which would indicate early collaboration. However, from general observations, at the beginning of the project, SEGs lack group cohesion. This lack of cohesion results in considerable activity as the groups need to get to know one another, determine individual members strengths and weaknesses, and in understanding the project domain. As the project matures the groups begin to work more steadily, each member gains a clearer understanding of the project and what their individual contribution is. Thus, the group members are able to work more independently as the project progresses.

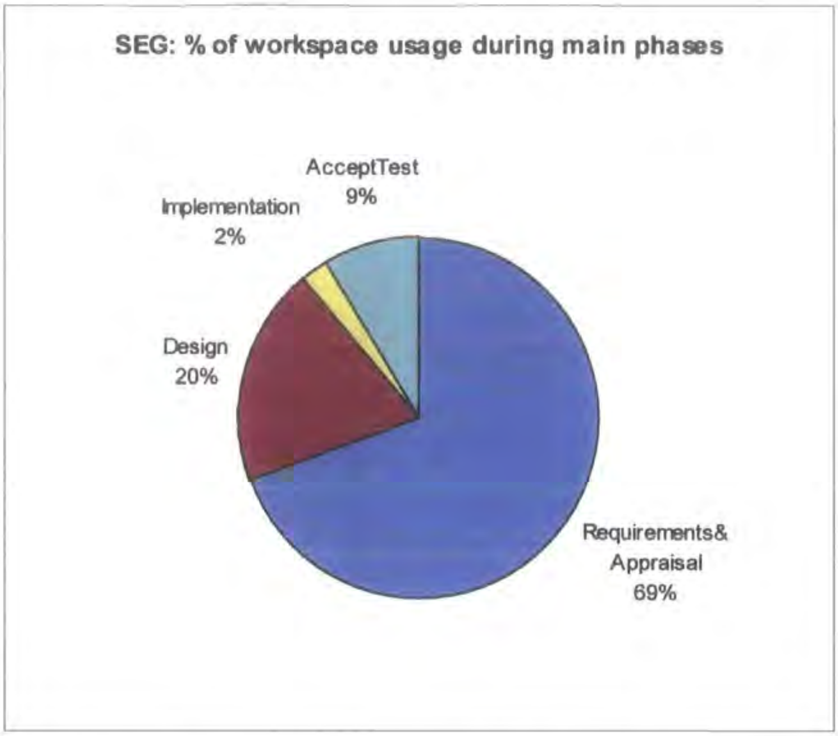

Figure 6-9: SEG workspace usage during the software lifecycle

6.4.2 Hypothesis 3 Conclusion

The results show that over 50% of the total workspace usage occurs within the initial lifecycle phases. However, the results presented for this hypothesis are primarily taken from activities undertaken in the workspaces and therefore it is not possible to prove conclusively that more collaboration occurs early in the software lifecycle. These results do not take into account face-to-face, video conferencing and email activities which all supplemented the workspaces. Each of these modes of communication still play an important role, to varying degrees, within the projects (although the JTAP 2 students felt that most video conferencing sessions were unproductive as the time was generally spent trying to get the audio to work properly).

Whilst the results do not prove the hypothesis, general observations, especially with SEGs, show that there is intensive activity undertaken in the requirements phase. This activity takes the form of ad-hoc and informal meetings, discussions during practical sessions, lunchtimes etc. This activity can be attributed to new groups beginning to form and to familiarise themselves with each other and the project. As the project progresses, group roles and organisation becomes more stable.

6.5 Hypothesis 4

Synchronous communication has an important role to play in both collocated and distributed software engineering groupwork.

6.5.1 Subjects and data collection methods

Student workgroups	*Method of data collection*
JTAP 1 - 3rd year distributed students	• Questionnaire • Observation
JTAP 2 - 3rd year distributed students	• Questionnaire • Focus group meeting
SEG	• Questionnaire • Observation • Focus group meeting

For this hypothesis results are graphically presented and generally represent two comparisons:

1. distributed and collocated student groups undertaking the same task within the same time scale,
2. distributed student groups undertaking different tasks within different time scales.

The following results show a comparison between the JTAP1 pilot exercises (chapter 3 section 3.2.1) and the SEG video conferencing short session (chapter 5 section 5.5.1). The SEG session mirrored that of the JTAP1 exercise. It is difficult to make a strict comparison between these results even though the same exercise was undertaken, as the work groups and the locations were different. For example, the SEG students were all 2nd year computer science students, and they were collocated on the same campus whilst the JTAP1 students were all 3rd year students in geographically distributed locations. What can be shown, and is discussed in the following sections, are the differences in responses to the use of each of the video conferencing tools: Video, Chat, Shared Whiteboard, Audio.

6.5.2 Video

The video results in Figure 6-10, show a comparison between distributed and collocated students who undertook the same task. Findings from the questionnaire results indicates that students felt that video was beneficial from the point of view of providing a sense of presence of the other group members. This resulted in other members of the group now having a stronger identity because of the ability to "put a face to a name". From a work point of view, because of the slow refresh rate of the video image giving a "blocky" picture, only 21% of JTAP1 students agreed that it was useful in the completion of the actual software engineering task.

In contrast only 6% of SEGs thought the tool useful. In comparison to this, work undertaken by Olson et. al., (Olson, Olson et al. 1997) found that distributed groups using high quality video, produced work with indistinguishable quality when compared to those groups using the same technology but in a face-to-face situation. The video image used by students in the research reported here, is relatively small and the quality of the image is not sharp, but given the findings of Olson and Olson et al, improvement to the video quality could lead to different results.

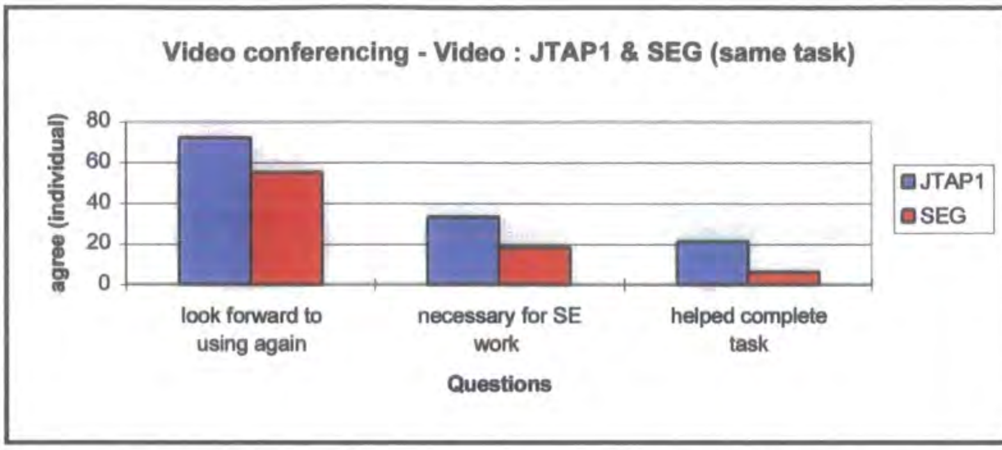

Figure 6-10: Video use for collocated and distributed groups

In contrast 6-11 looks at the two distributed groups who undertook very different tasks over different periods of time. The results show a closer correlation between the groups responses, but there was still only an approximate 20% response to the fact that video had played a significant role in the completion of either task. Once again student comments were similar to those described in the previous paragraph.

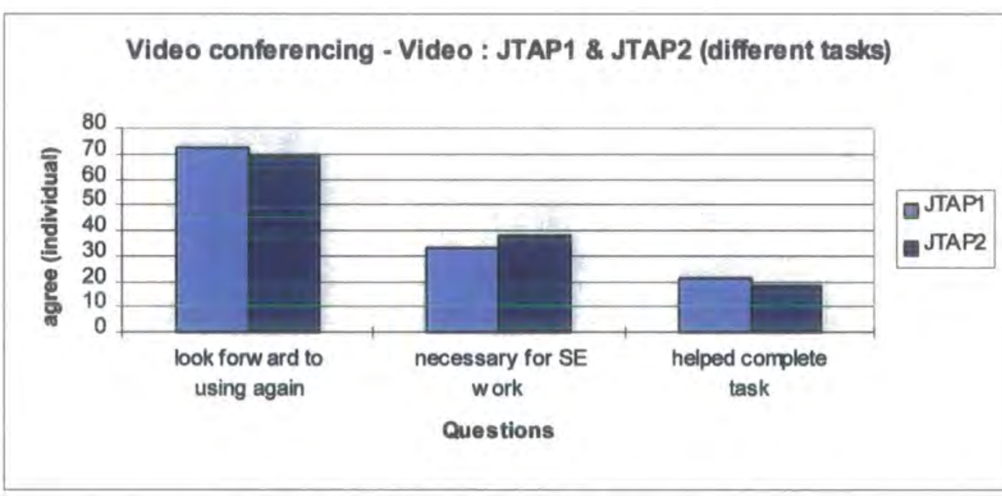

Figure 6-11: Video use for distributed groups

6.5.2.1 Video Conclusion

Video was generally thought to be useful, however, communication problems did occur over the video channel. These, in the main, can be attributed to network congestion. Confusion also occurred due to the scarcity of ways to direct questions to individuals. In normal face-to-face communication, eye contact and other gestures can be very important. This was often lost because of the low quality of the video image. Similarly the video and audio do not often afford lip synchronisation, which can lead to more confusion as it can be difficult to determine who it is who is actually speaking. This is similar to results from work undertaken by Tang and Isaacs (Tang and Isaacs 1993) who found that because of a time delay between video and audio, participants could not see verbal openings and therefore turn taking became a problem.

Whilst many students both collocated and distributed felt that video had not contributed to the successful completion of the task in hand, over 55% of them said they would look forward to using video again. As many comments from the students regarding video were negative, this interest in using video again may therefore be attributed to the "novelty" element of using this technology.

6.5.3 Audio

The audio results in Figure 6-12, shows that there is a significant difference between JTAP1 and SEG responses. Whilst the SEG students recorded very similar likes/dislikes relating to audio to those of the JTAP1 students, this marked difference in responses has been attributed to the different environments in which the students were working. SEG students undertook the exercise in the SEGLab and were therefore in close proximity of each other. The JTAP1 students were distributed across three sites; at each site each student worked in a separate room.

The audio quality in the SEGLab was poor with interference caused by the increased network traffic. With audio it is necessary to encourage the students to speak into the microphones more clearly, slowly and louder than they would normally. In particular, this increase in volume within the SEGLab was unacceptable; and most students could often hear their group members from the other side of the room. In many instances, students were lacking in self confidence to speak out loud or to continue a conversation as they were never sure if anyone could hear them.

In Figure 6-12 it is clearly shown that the 43% of SEGs felt that whilst audio was necessary for software engineering, only 3% used it to complete the task in hand and only 5% would look forward to using it again. These figures are significantly different to those of the JTAP1 students. Observations made within the SEG environment, showed that on more than one occasion the groups resorted to communicating face-to-face rather than continue with audio that had intermittent quality. In the case of the JTAP1 students, they had no option but to persevere with audio, and once they became adjusted to the varying quality i.e. time delays and clipped sentences, they worked reasonably well.

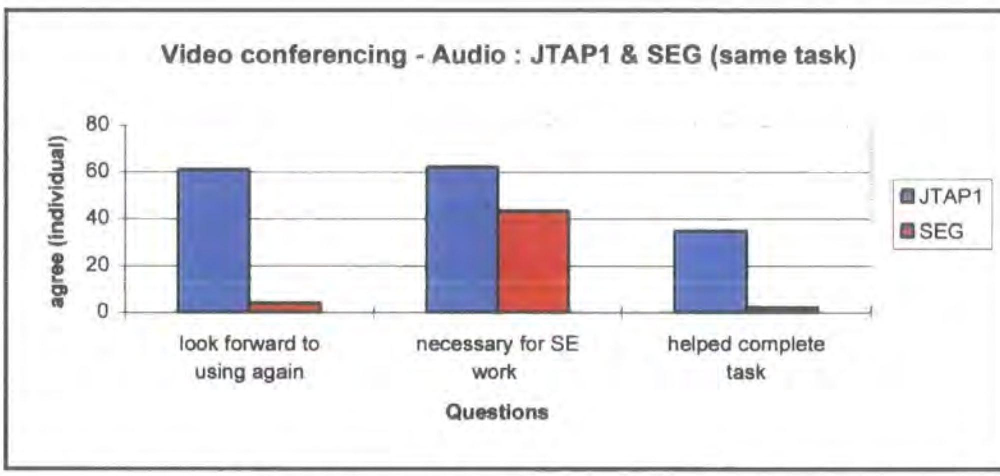

Figure 6-12: Audio use for distributed and collocated groups

In Figure 6-13, the student groups were both geographically distributed. What is shown are similar results even though the tasks undertaken had different content and were of different duration.

These students had greater motivation to continue with the audio function, and in some cases adapted to the intermittent quality.

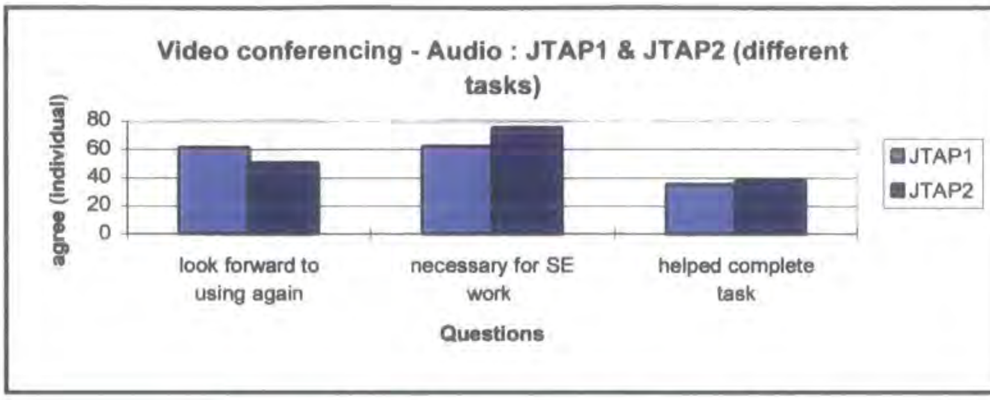

Figure 6-13: Audio use for distributed groups

6.5.3.1 Audio Conclusion

Audio was perceived as having useful functionality within software engineering working. Thus, its potential importance was recognised. The audio was found to be unreliable from time to time, but when functional, it was of an acceptable quality. Many students found audio failure to be more disruptive to communications than video failure, as communication is primarily based on the audio channel. One of the main problems with the audio was students' inability to use it properly. More training would improve the situation, and, therefore, the acceptance of audio. Students were, in general, disappointed with audio quality and often gave up using the audio very quickly.

6.5.4 Shared whiteboard

The shared whiteboard enabled students to make synchronous annotations to shared documents etc. Students found this application easy to use, but noted that there was only an average user interface and that the functionality offered was limited. Each group needed to develop a protocol determining who would create a workbook[20]. In some cases, a work protocol was not used, which resulted in

[20] A workbook is essentially a set of shared whiteboard pages that the group can work with and move between. It is necessary for one member of the group to initially open a workbook and then the remainder of the group can join it.

each group member creating separate workbooks, and as such, other group members could not see each other's work.

A positive response (shown in Figure 6-14 and Figure 6-15), from both collocated and distributed students, for the use of a shared whiteboard, was given. Over 70% of students found the whiteboard helped them complete the task, with over 80% stating that the whiteboard was necessary for software engineering work. All students felt that synchronous collaborative diagramming was an important feature for software engineering work, and would have found it extremely difficult to complete the task without using it. This finding is similar to that of Whittaker and Geelheod (Whittaker, Geelhoed et al. 1993) whose study determined that a shared whiteboard provided a permanent record of group activity, and concluded that a shared whiteboard was most useful applied to tasks possessing a strong graphical component.

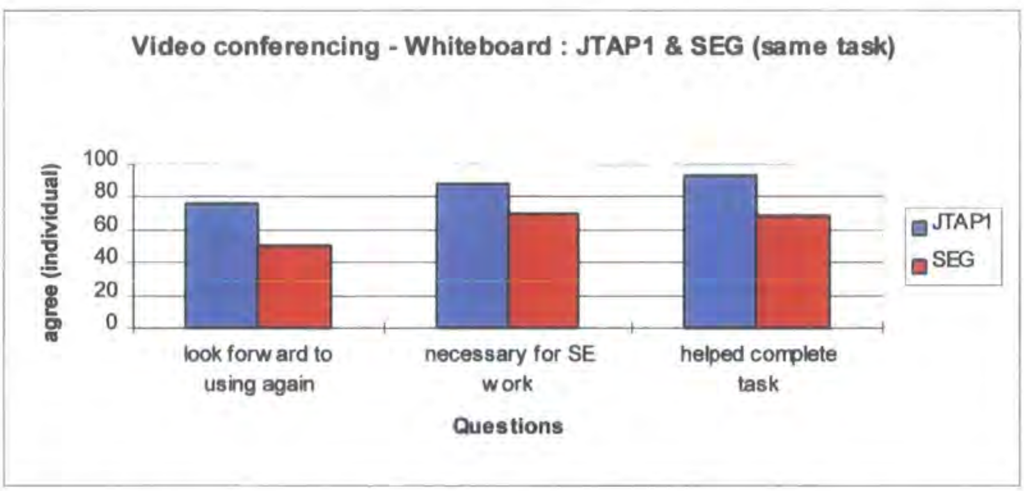

Figure 6-14: Whiteboard use for distributed and collocated groups

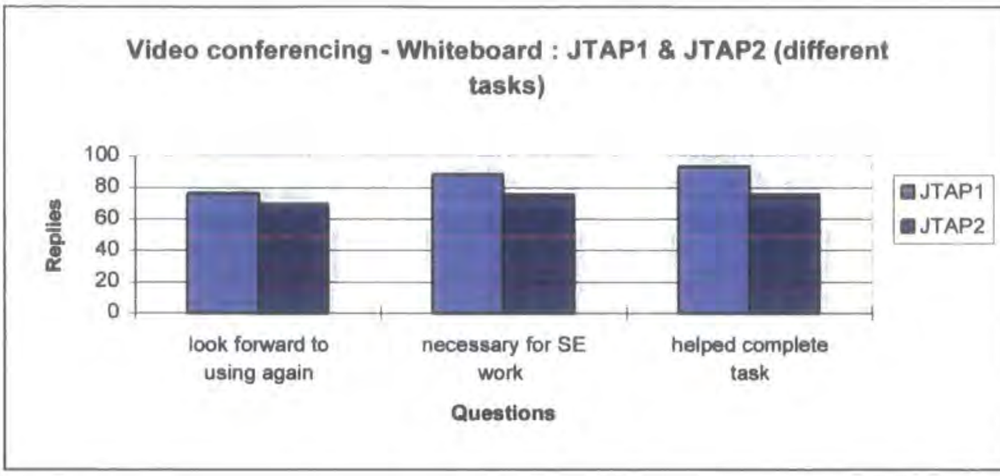

Figure 6-15: Whiteboard use for distributed groups

6.5.4.1 Whiteboard Conclusion

Both distributed and collocated students enjoyed using the whiteboard even though each of these groups used a different whiteboard during their exercises. The initial whiteboard that the JTAP1 students used lacked basic functionality and was relatively unintuitive to use. This whiteboard was replaced for subsequent exercises and the later whiteboard, although still lacking in software engineering functionality (i.e. connections between entities on an ERD moving when the entity was repositioned), was deemed a very necessary component of DVC.

6.5.5 Chat

Chat received positive comments. Many students had previously used chat facilities in newsgroups and the UNIX "talk" tool and were therefore familiar with the concept and use of a chat tool. In Figure 6-16 and Figure 6-17 it can be seen that the use of Chat was well received by all groups, with over 60% of students being happy to use it in the future. Similarly, 60% of the students felt that Chat was necessary for undertaking and completing software engineering work. Generally Chat was seen as being useful for discussion and negotiation, and shared document creation.

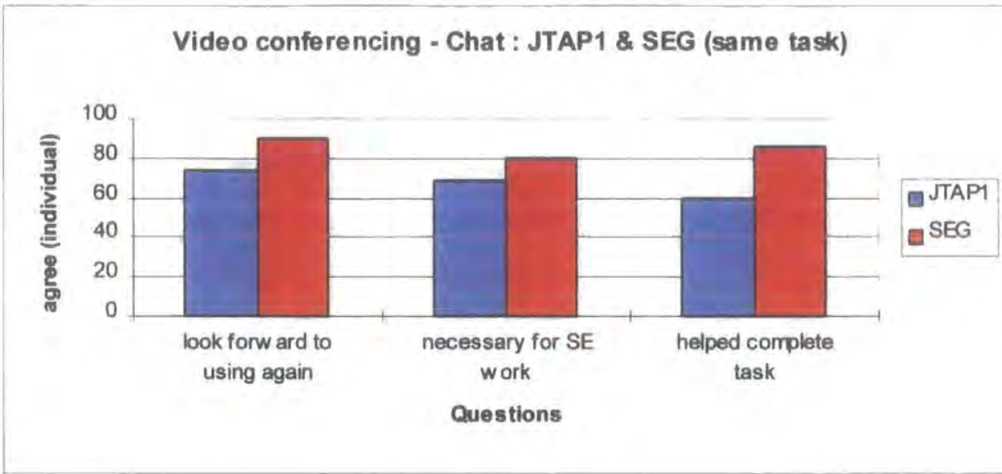

Figure 6-16: Chat use for distributed and collocated groups

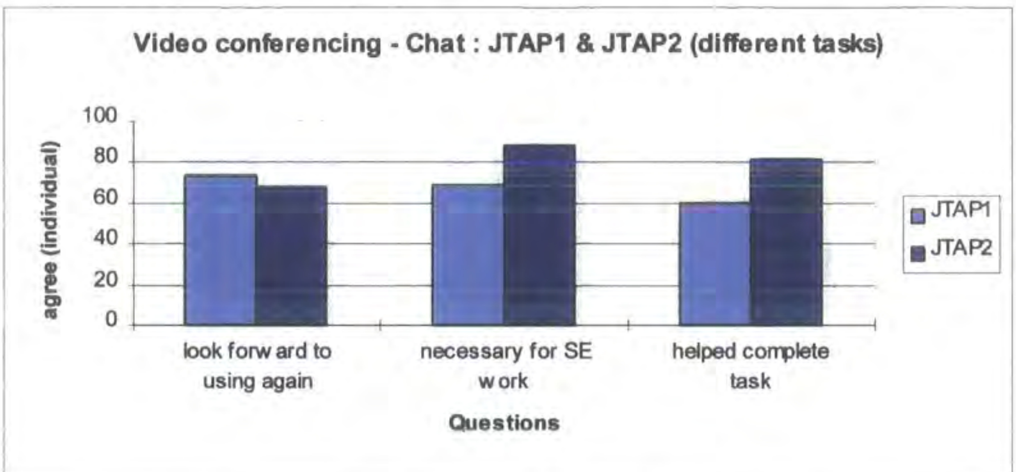

Figure 6-17: Chat use for distributed groups

6.5.5.1 Chat Conclusion

Chat was seen as a necessary component for software engineering work. Typical comments received from students were "quick and easy", "good in its simplicity". As audio problems were frequently experienced, Chat provided an invaluable backup to the audio provision. It was noted that most students used the chat window as back up to failing audio.

6.5.6 Completion of Task

When students were asked had the DVC aided them in successfully completing the task, most were dissatisfied with their result (Figure 6-18). All students were asked if they would have successfully completed the task if it had been undertaken in a face-to-face situation; over 80% said that the exercise would have been successfully completed. Based on these figures, it is assumed that the problem lies in the DVC technology's lack of support for software engineering rather than the tasks set being too difficult.

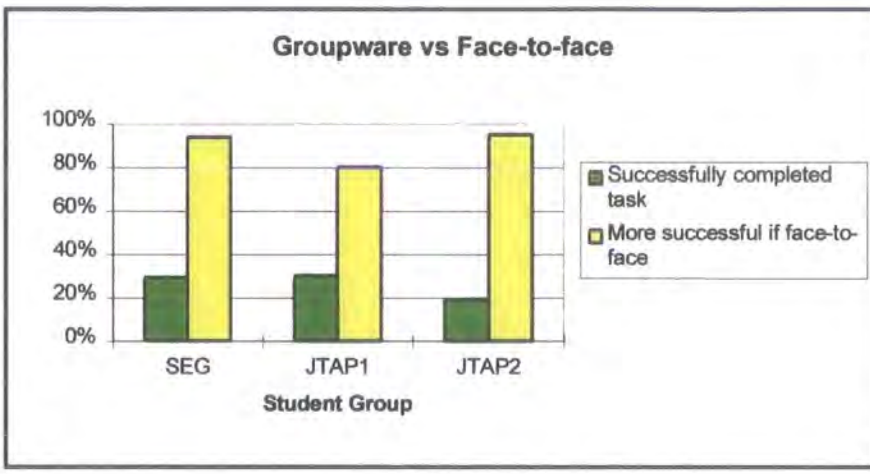

Figure 6-18 : Groupware vs. Face-to-face

6.5.7 Hypothesis 4 Conclusion

Based on the results presented, synchronous communication has an important part to play in software engineering work. However, the importance has been recognised more, within distributed work than collocated work. The whiteboard and Chat tool provided the most reliable and useful communication between group members. Audio is seen as being an important component, but, at present, performance levels over the Internet does not support the level of speech quality and reliability that is required. Video is seen to have limited use because of the image size and poor quality.

For JTAP1 and SEG students only a brief introduction of the technologies was given because of the nature of the task i.e. a short exercise. The lack of familiarity with the software caused many students concern as they felt they spent most of their time attempting to master the technology rather than the task in hand. This would indicate the necessity for a more prolonged introduction to the technologies.

Over 50% of students agreed that they would like to use DVC again (excluding SEGs future use of audio), even though using DVC resulted in problems occurring (previously mentioned). Therefore, the students' desire to use DVC in the future may be attributed to their interest in future developments and the "novelty" factor of this technology.

6.6 Summary

The table below summarises the hypotheses concerned and the conclusion drawn.

Hypothesis	Conclusion
1. The introduction of an asynchronous shared workspace into software engineering groupworking will aid group members in organising and coordinating their work.	This hypothesis has proved to be correct. A graphical, web-based, multi-platform environment which provides awareness of group member activities aids in the organisation and coordination of group work.
2. Greater use of shared workspace functionality will be made as the project progresses	This hypothesis has in part been proved to be incorrect. The use of the workspace functions for SEG showed no significant increases, but the functions used were used more appropriately. This improved use was for a limited number of

	functions, and failure to make improved use of others may be attributed to poor motivation by students to learn "extra" functions. As *SEGWorld* did not well support the implementation phase of the project, which is a major phase, minimal data was available for collection.
3. Students undertake more collaboration in the earlier stages of the software lifecycle	It has not been possible to prove or disprove this hypothesis because of insufficient data, but from general observations it was felt that there is more activity at the beginning of the software lifecycle when groups are forming and understanding the task in hand, and the potential of each member.
4. Synchronous communication has an important role to play in both collocated and distributed software engineering groupwork	This hypothesis has been proved to be correct. The individual components of synchronous communication have been evaluated in the context of software engineering with the results showing that the benefits of video are marginal and hard to justify. Audio is seen as being vital, but, at present, performance levels over the Internet are inadequate and unstable. Chat and whiteboard applications were both found to be useful and reliable. Current commercially available DVC applications are in the main generic and not specific to software engineering. Synchronous communication has an important role to play in software engineering work, especially when having to work in geographically distributed locations. For collocated students, this role is less significant because face-to-face interaction is possible.

The final chapter considers the research outcomes and contributions, and presents some topics for further work.

Chapter 7

Conclusions and Further Work

7.1 Introduction

In this research, an investigation has been carried out into computer support for cooperative work within software engineering and, in particular, groupwork in the context of software engineering education. This investigation has involved the development of collaborative working environments both synchronous and asynchronous. The synchronous environment provided DVC (audio, video, chat and shared whiteboard) for cooperative working. The asynchronous environment has been developed by tailoring the existing shared workspace system - BSCW, for distributed student working, and *SEGWorld* for SEG students based at Durham. A major objective of providing both environments was to help determine the usefulness of such technologies to software engineering students. To determine the usability of these environments, cases studies involving different technologies and groups of students have been undertaken.

To help assess the value of the work reported here, the work is evaluated against a set of success criteria given in Chapter 1. In the following section, the degree to which the work satisfies these criteria is discussed.

7.2 Evaluation of the Criteria for Success

This section discusses each of the success criteria defined in Chapter 1 with respect to the work performed. Each of the success criteria are show in italics.

a) *Investigation of the areas of CSCW and groupware to determine current commercially available software specific to software engineering.*
An overview is presented in Chapter 2 describing the areas of CSCW, groupware and software engineering. Within this chapter typical software engineering processes are described and the use of groupworking technologies to support these processes is considered. This research has attempted to identify groupworking applications specific to software engineering. Whilst tools and software engineering platforms have been developed, they are still research based and are not commercially available. Groupware that is available is, in the main, generic, e.g. BSCW and Lotus Notes™, and whilst offering some tailorability, these generic groupware do not fully support software engineering processes such as, code development.

b) *An identification of a set of requirements for collaborative working support for software engineering students, both distributed and collocated, and a formulation of hypotheses regarding software engineering students' use of groupware.*
A set of requirements for collaborative work have been identified (Chapter 3, 3.5) and although not comprehensive, form a basis for future student groupwork activities. These requirements have placed emphasis on the need for student understanding, motivation, training and technical support. These areas must be addressed for successful adoption of new technologies and working methods into the curriculum.

Software engineering students must have good technical knowledge and skills, but it is equally important that they understand and experience the groupworking process. These students need to be aware of the problems, both sociological and technical, that can be encountered whilst undertaking groupwork. Providing this awareness is the forerunner to improving student motivation in adopting the new technologies and methods of working. To facilitate this awareness, students were introduced to DVC and the shared workspace concept.

The introduction of these technologies to the groupworking students, formed the basis for formulating a number of hypotheses. These hypotheses were based on the investigation of particular areas of groupworking. These areas were organisation and coordination of tasks, determining the level of cooperation that occurs within the phases of the software lifecycle, the levels of usage of functions within a shared workspace and what importance is placed on the role of synchronous communication within software engineering student groupwork.

c) *An undertaking of case studies to prove or disprove the hypotheses, using groupware with software engineering students.*
The aim of the case studies has been to allow the students to undertake software engineering tasks using the available technologies and applications. In doing this, it was possible to collect the necessary data from the different groups to determine if the hypotheses formulated regarding the use of the technologies (DVC and shared workspace), could be proved or disproved.

1. With respect to organisation and coordination, the introduction of a shared workspace provided a formal setting for the practical side of the software engineering course undertaken at Durham. The students found this both helpful and useful. In general, the students felt that the hierarchical structure of their workspace, the configuration management provided, and the awareness of other groups members' activities assisted them in organising and coordinating their work.

2. Collocated student use of the workspace functions although limited was reasonably comprehensive, but, not consistent throughout the year. This has been attributed mainly to poor response times of the server upon which *SEGWorld* was running, and *SEGWorld* not supporting code development. In addition to this, the introduction of the *SEGWorld* was not well supported by demonstrators, and, therefore, the students formed a poor opinion of the system. This highlights the points raised in Success Criteria

b, regarding providing groupwork understanding and technical support. It is these factors which play an important role in motivating students.

3. Attempting to determine if more collaboration is undertaken in the earlier stages of the software lifecycle proved to be difficult. Whilst it was possible to monitor workspace usage, the information provided was only part of the overall picture of collaboration. For SEGs, the main mode of communication used was face-to-face and email. As these students spend much of their working days together, it was impossible to monitor and collect quantitative data from this type of collaboration. Similarly, for the JTAP students whose communication channels were limited (their face-to-face opportunities were restricted) to video conferencing and email; again it was not possible to monitor either of these effectively. Observations have, however, shown that at the beginning of the projects, there was intense activity within the student groups. Therefore, whilst not being able to prove this hypothesis because of insufficient data, it could not be disproved either.

4. DVC not only provided the students with the opportunity to evaluate new technologies in a practical manner, but also it allowed them to gain an understanding of the problems encountered when using groupware to carry out software engineering tasks. These problems were not just technical, but were also social in so much that the students were forced to develop a group working protocol, e.g. who was tasked to open a shared workbook, or to ensure that an agenda was written and made available before meetings. The technology could be unreliable, therefore, there was a need to be precise in words and actions during a DVC session. A side effect of this was better group cohesion, the students had to work harder at coordinating their efforts. Although synchronous communication was popular with the students, it had no significant effect on the successful completion of the tasks during SEG and JTAP1 trial sessions. In contrast, the JTAP2 students felt it was a very necessary form of communication, and would have found it extremely difficult to complete their sub-project without it.

From a student perspective, many regarded the experience gained through the use of *SEGWorld* and DVC as very useful. Negative comments received were, in general, regarding the slow response times of the *SEGWorld* server, which made the system less attractive to use and resulted in the relatively low number of SEG students being happy to use the system again. In contrast, the distributed students enjoyed using the shared workspace and found it invaluable for working across sites. For all students, the use of DVC was a new experience, and, in some cases, disappointing due, in particular, to their expectations of high quality audio which was not often possible. What was valuable for the students was that they were exposed to technology from both a technological and end user perspective, therefore, they gained an insight into the benefits that these technologies can provide and the problems that can arise when using these technologies.

7.3 Further work

The findings described within this research are mainly based on anecdotal evidence, and, as such, the results presented give an insight into software engineering student groupworking, both collocated and distributed. Problems have occurred within some of the results, due to, the questionnaire results containing subjective views. This type of anomaly can be common to any questionnaire based results, due, to misinterpretation or misunderstanding of the question, or simply, that little thought is given to the answer supplied.

To build on the work reported in this research, it would be interesting to undertake a similar investigation on a small number of SEGs. This further investigation would help determine if the use of groupware does add value to the project work, in so far, as leading to a measurable improvement of its quality.

To undertake this work, would involve having a more controlled work environment, where the SEGs are "shadowed", as far as possible, in their formal and informal interactions. This "shadowing" would allow for the monitoring and subsequent collection of results from observations, face-to-face interaction, email and, again, questionnaires. Collection of these types of data would enable other hypotheses, in addition to those presented here, to be categorically proved or disproved. Thus, determining the feasibility of groupware within software engineering education. This type of monitoring however, would present problems, such as, it could limit the creative aspects of some SEGs if such conditions were imposed. But, by carrying out a more controlled investigation, the correlation of results would provide a more objective view.

Whilst further investigation of the work, reported here in this research, would be useful, further work should be carried out, to firstly, improve *SEGWorld* from an administrative point of view, and secondly, to introduce the shared workspace concept into other modules within the Computer Science department.

This anticipated increase in the number of students using BSCW, places a greater emphasis on the need for the development of support tools to automate processes. From an administrative perspective, tools should be developed to make the administration of *SEGWorld* more efficient. This toolset should automate processes such as, the collection and depositing of SEG deliverables into a secure workspace area, and, in particular, the development of a fully integrated marking system. This marking scheme would involve student deliverables within the BSCW system being marked and commented on by the appropriate tutor, using web-based online marking forms. These marks and comments, contained within an automatically generated email, would provide valuable feedback to the students.

The use of BSCW could also be extended to other modules within the Computer Science department and, in particular, the Programming Design Structure (PDS) module. What BSCW would provide is a focal point for the PDS students to work from. Within the first year of university, some students may often feel isolated, which can result in these students falling behind in their studies. The introduction of a shared workspace which provides: threaded discussion forums, frequently asked questions, an awareness of other student activities, module information, and simply being part of a smaller group, would go some way in providing this focal point. It would be hoped that this would provide the student with another point of contact, in addition to the module tutor.

7.4 Final Words

Students found that the experience they gained in undertaking groupwork to be enjoyable, enlightening and an attractive addition to their CVs. The introduction of groupware, whilst being problematic, has given them added experience. As one SEG student commented on the questionnaire *"looking back at the SEGWorld system and the structure and design of the SEG project I think they work well together as SEGWorld supports the SEG and likewise the SEG project provides the field to test the full use of SEGWorld".*

References

Bannon, L. (1993). "CSCW: An Initial Exploration." Scandinavian Journal of Information Systems **5**: 3-24.

Bannon, L. and S. Bodker (1997). Constructing Common Information Spaces. Proceedings of the Fifth European Conference on Computer Supported Cooperative Work, Lancaster University, UK, Kluwer Academic.

Bannon, L. and K. Schmidt, Eds. (1991). CSCW: Four Characters in Search of a Context. Studies in Computer Supported Work, Elsevier Science.

Bennett, K., M. Munro, et al. (1996). A proposal for the JISC Technology Application Programme : Developing a Virtual Community for Student Groupwork, Department of Computation, UMIST.

Bowers, J., G. Button, et al. (1995). Workflow From Within and Without: Technology and Cooperative Work on the Print Industry Shopfloor. Fourth European Conference on Computer-Supported Cooperative Work 1995, Stockholm, Sweden, Kluwer Academic.

Brereton, P., S. Lees, et al. (1998). "Distributed Group Working in Software Engineering Education." Information and Software Technology **40**: 221 -227.

Brooks, F. P. (1986). Keynote Speech: People Are Our Most Important Product. Software Engineering Education: The Educational Needs of the Software Community. N. E. Gibb and R. E. Fairley, Springer-Verlag.

Brothers, L., V. Sembugamoorthy, et al. (1990). ICICLE: Groupware For Code Inspection. CSCW'90, Los Angeles, CA, ACM.

Checkland, P. (1997). CSCW in the context of IS: can the reality match the rhetoric? Key Note Address to ECSCW '97, Lancaster University, UK.

Christie, A. M. (1995). Software Process Automation, Springer-Verlag.

Coleman, D. (1995). Groupware Technology and Applications: An Overview of Groupware. Groupware Technology and Applications. D. Coleman and R. Khanna, Prentice-Hall PTR: 3 - 41.

Coleman, D. and R. Khanna, Eds. (1995). Preface - Groupware: Technology and Application, Prentice Hall.

Curtis, B., H. Krasner, et al. (1988). "A Field Study of the Software Design Process for Large Systems." Communications of the ACM **31**(11): 1268 - 1287.

DeMarco, T. and T. Lister (1987). Peopleware. New York, Dorset House.

Dewan, P. and J. Riedl (1993). "Toward Computer-Supported Concurrent Software Engineering." Computer(January 1993): 17 - 27.

Drummond, S., C. Boldyreff, et al. (1997). Software Engineering Group Project work: Past, Present and Future. Presented at Special Interest Group in the Teaching of Software Engineering (SIGToSE), London., http://www.dur.ac.uk/~dcs1sad/papers.

Ellis, C. A., S. J. Gibbs, et al. (1991). "Groupware : Some Issues and Experiences." Communication of the ACM **34**(1): 39-58.

Englebart, D. and H.Lehtman (1988). Working Together. Byte: 245 - 252.

Forte, G. and R. J. Norman (1992). "A Self-Assessment by the Software Engineering Community." Communication of the ACM **35**(4): 28 -32.

Galegher, J. and R. E. Kraut (1990). Computer-Mediated Communication for Intellectual Teamwork: A Field Experiment in Group Writing. CSCW 90, ACM.

Garlan, D., D. P. Glutch, et al. (1997). "Agents of Change: Educating Software Engineering Leaders." IEEE Computer **November**(1997): 59-65.

Gibb, N. E. (1989). "The SEI Education Program: The Challenge of Teaching Future Software Engineers." Communications of the ACM **32**(3): 594 -605.

GMD-FIT Basic Support for Cooperative Work (BSCW), http://www.gmd.de/.

Grudin, J. (1988). Why CSCW Applications Fail: Problems in the Design and Evaluation of Organisation Interfaces. Conference on Computer -Supported Cooperative Work, Portland, Oregan.

Grudin, J. (1991). "CSCW Introduction." Communication of the ACM **34**(12): 30-34.

Grudin, J. (1996). "Regrouping for Groupware." American Programmer **9**(8): 7 - 10.

Harrison, W. H., H. Osser, et al. (1990). "Coordination Concurrent Development." CSCW '90: 157 - 167.

Hawryskiewicz, I. and I. Gorton (1996). Platforms for Cooperative Software Development. American Programmer. **August**.

Heath, C. and P. Luff (1991). Collaborative Activity and Technological Design: Task Coordination in London Underground Control Rooms. Second European Conference on Computer-Supported Cooperative Work, Amsterdam, Kluwer.

Horning, J. J. and D. B. Wortman (1977). "Software Hut: A computer program engineering project in the form of a game." IEEE Trans. Software Engineering **SE-3**(July): 325-330.

Howard, R. (1987). "Systems Design and Social Responsibility: The Political Implications of "Computer Supported Cooperative Work"." Office: Technology and People: 175-187.

Howard, R. (1988). Panel Remarks: "CSCW: What does it mean?". Proceedings of CSCW '88, Portland, Oregon, ACM.

Johansen, R. (1989). User Approaches to Computer-Supported Teams. Technological Support for Work Group Collaboration. M. H. Olson, Lawrence Erlbaum Associates: 1 - 33.

Jones, T. C. (1986). Programming Productivity. New York, McGraw-Hill.

Jorgensen, M. (1995). "The Quality of Questionnaire Based Software Maintenance Process." Software Engineering Notes **20**: 71 - 73.

JTAP (1996-98). JTAP-2/140: Developing a Virtual Community for Student Groupwork, http://cssec.co.umist.ac.uk/.

Kling, R. (1991). "Cooperation, Coordination and Control in Computer Supported Work." Communications of the ACM **34**(12): 83-88.

Kraut, R. E. and L. A. Streeter (1995). "Coordination in Software Development." Communication of the ACM **38**(3): 69 - 81.

Layzell, P., L. Macaulay, et al. (1997). Developing a Virtual Community for Student Groupwork: End of Year JTAP report, Department of Computation, UMIST.

Lloyd, P., Ed. (1994). Groupware in the 21st Century: Computer Supported Co-operative Working Towards the Millenium, Adamantine Press Ltd.

Mackay, W. E., T. Malone, et al. (1989). How do experienced Information Lens users use rules? CHI '89 Proceedings, ACM. **Austin, Texas, USA:** 211 - 216.

McCracken, W. M. (1997). "SE Education: What Academia Can Do." IEEE Software **14**(6): 27 -29.

Nardi, B. A. and J. R. Miller (1990). An Ethnographic Study of Distributed Problem Solving in Spreadsheet Development. CSCW '90, Los Angeles, CA, ACM.

Olson, J. S., G. M. Olson, et al. (1997). Face-to-Face Group Work Compared to Remote Group Work With and Without Video. Video-Mediated Communication. K. E. Finn, A. J. Sellen and S. B. Wilbur, Lawrence Erlbaum Associates.

Olson, J. S., G. M. Olson, et al. (1993). "Groupwork close up: a comparison of the group design process with and without a simple group editor." ACM Transactions on Information Systems **11**(4): 321 -348.

Olson, J. S. and S. Teasley (1996). Groupware in the Wild: Lessons Learned from a Year of Virtual Collocation. Computer Supported Cooperative Work, Boston, Massachusetts, USA, ACM Press.

Plattner, B. (1994). Forward comments. Towards a CSCW Framework for Scientific Cooperation in Europe. H. B. Lubich, Springer-Verlag. **Lecture Notes in Computer Science 889**.

Plowman, L., Y. Rogers, et al. (1995). What are workplace studies for? Forth European Conference on Computer-Supported Cooperative Work, Stockholm, Sweden, Kluwer Academic.

Rettinger, L. A. (1995). Desktop Videoconferencing: Technology and use for remote seminar delivery. Computer Engineering. Raleigh, North Carolina, North Carolina State University.

Robillard, P. N. (1996). Teaching Software Engineering through a Project-Oriented Course. Ninth Conference on Software Engineering Education, Daytona Beach Florida, IEEE Computer Society.

Rodden, T. (1993). Technological Support for Cooperation. CSCW in Practice: An Introduction and Case Studies. D. Diaper and C. Sanger, Springer-Verlag: 1-21.

Rodden, T. and I. Sommerville (1994). Supporting Cooperative Software Engineering. Computer -Supported Cooperative Work. S. A. R. Scrivener, Avebury Technical: 207 - 225.

Salim, S. S. (1998). An Investigation into CSCW and IS. Department of Computation, University of Manchester Institute of Science and Technology.

Sasse, M. A., M. J. Handley, et al. (1993). Support for Collaborative Authoring via Email: The MESSIE Environment. Third European Conference on Computer-Supported Cooperative Work, Milan, Italy, Kluwer Academic.

Schrage, M. (1996). "The Future of Groupware." American Programmer 9(8): 3 - 6.

Sommerville, I. (1995). Software Engineering, Addison-Wesley.

Tang, J. C. and E. Isaacs (1993). "Why Do Users Like Video? Studies of Multimedia-Supported Collaboration." Computer Supported Cooperative Work (CSCW) 1(3): 163-196.

Vessey, I. and A. P. Sravanapudi (1995). "CASE Tools as Collaborative Support Technologies." Communications of the ACM **38**(1): 83 - 95.

Whittaker, S., E. Geelhoed, et al. (1993). "Shared Workspaces: how do they work and when are they useful?" Int. Journal Man-Machine Studies(39): 813 -841.

Williams, N., G. S. Blair, et al. (1994). The Impact of Distributed Multimedia Systems on Computer Support for Cooperative Work. Computer Support for Co-operative Work. K. Spurr, P. Layzell, L. Jennison and N. Richards, John Wiley & Sons: 147 -169.

Wilson, P. (1994). Introducing CSCW - What Is It and Why We Need It. Computer Supported Cooperative Work. S. A. R. Scrivener, Avebury Technical: 1 - 18.

Young, R. (1998). Are Universities ready for Groupware? JTAP -2/140 Department of Computation, UMIST, UK.

Appendix A

JISC Technology Applications Programme
UMIST, University of Durham and Keele University

Registration Sheet

Personal Details

1. Name ________________________________
2. Date __________
3. University ________________________________
4. Course ________________________________
5. Year __________
6. Age __________
7. Sex Male / Female
8. Email Address ________________________________

Unique CSCW ID __________

Developing a Virtual Community for Student Groupwork, JTAP-2/140

JISC Technology Applications Programme
UMIST, University of Durham and Keele University

Background Questionnaire

Personal Details

1. Date ____________
2. CSCW ID ____________
3. University ____
4. Course ____
5. Year ____________
6. Age ____________
7. Sex Male / Female

Work Experience - Full Time & Industrial Placements

1. Number of years worked before starting course ____________
2. Number of years relevant Information Technology / Software Engineering work ____________
3. Have you done an industrial placement as part of your course Yes /No

Group Working Experience - In Employment

1. Have you had training in group working Yes / No

   If yes give details of the longest / most recent / most effective / best

   ____

2. What was the duration of the group working project ____________
3. How many people were in the group ____________
4. Did the group have an official leader Yes / No

   If yes were you the leader Yes / No

Group Working Experience - In Course

1. Have you had training in group working Yes / No

   If yes give details of the longest / most recent / most effective / best

   ____

2. What was the duration of the group working project ________
3. How many people were in the group ________
4. Did the group have an official leader Yes / No

   If yes were you the leader Yes / No

Computer Experience

1. What windows operating systems have you used
   - Windows 95 / Windows NT Yes / No
   - MacOS Yes / No
   - Windows 3.x Yes / No
   - UNIX X-Windows Yes / No
2. Which is the operating system you predominantly use ________
3. Have you used any Video Conferencing Yes / No
4. If yes, what and how often ______________________________

   ____

5. Have you used any chat tools (eg. UNIX talk, Netscape chat) Yes / No
6. Have you used any drawing tools (eg. Paintbrush, Corel Draw) Yes / No
7. Have you used any shared workspace tools (eg. BSCW, Lotus Notes) Yes / No
8. If yes, what and how often ______________________________

   ____

Developing a Virtual Community for Student Groupwork, JTAP-2/140

JISC Technology Applications Programme
UMIST, University of Durham and Keele University

Evaluation Questionnaire

To help us evaluate this exercise, would you please answer the following questions? All responses will remain confidential. Many thanks for your help.

1: Task Details

1. CSCW ID __________
2. Date __________
3. Title of task ______________________________
4. What were the CSCW ID's of the other two members of your group? ______________________________
5. What was the time allowed for the task? __________
6. Was there any previous preparation time? Yes / No

   If yes, how did you spend it? ______________________________

____

____

____

____

2: Environmental issues

Please tick JUST ONE of the options for each statements.

Audio	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagre e
I did not enjoy using the Audio component.					
I would look forward to using Audio in this way in the future.					
Audio is not necessary for Software Engineering students who need to work in this way.					
Audio greatly helped our group complete the task.					
I found Audio very easy to use.					

Whiteboard	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagre e
I did not enjoy using the Whiteboard.					
I would look forward to using the Whiteboard in this way in the future.					
The Whiteboard is not necessary for Software Engineering students who need to work in this way.					
The Whiteboard greatly helped our group complete the task.					
I found the Whiteboard easy to use.					

Chat	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
I did not enjoy using the Chat component.					
I would look forward to using Chat in this way in the future.					
Chat is not necessary for Software Engineering students who need to work in this way.					
Chat greatly helped our group complete the task.					
I found Chat easy to use.					

Video	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
I did not enjoy using the Video component.					
I would look forward to using Video in this way in the future.					
Video is not necessary for Software Engineering students who need to work in this way.					
Video greatly helped our group complete the task.					
I found Video easy to use.					

3. Environment Suitability

Please give brief answers to the following questions in your own words.

1. If you had to work in this way in the future, for which one of the following software engineering tasks would Audio be most suitable? (Please circle one option)
   Discussion/negotiation *Diagramming* *Presenting information*

Shared document preparation
Programming/testing/debugging
Why do you think so?

....
.
....
.
....
.

2. If you had to work in this way in the future, for which one of the following software engineering tasks would Audio be least suitable? (Please circle one option)
Discussion/negotiation *Diagramming* *Presenting information*
Shared document preparation
Programming/testing/debugging
Why do you think so?

....
.
....
.
....
.

3. What did you like best about Audio? Please give your main reason.

....
.
....
.
....
.

4. What did you like least about Audio? Please give your main reason.

....
.
....
.
....
.

5. If you had to work in this way in the future, for which one of the following software engineering tasks would Chat be most suitable? (Please circle one option)
Discussion/negotiation *Diagramming* *Presenting information*
Shared document preparation
Programming/testing/debugging
Why do you think so?

....
.
....
.
....
.

6. If you had to work in this way in the future, for which one of the following software engineering tasks would Chat be least suitable? (Please circle one option)
   Discussion/negotiation *Diagramming* *Presenting information*
   Shared document preparation
   Programming/testing/debugging
   Why do you think so?
   ....
   .
   ....
   .
   ....
   .

7. What did you like best about Chat? Please give your main reason.
   ....
   .
   ....
   .
   ....
   .

8. What did you like least about Chat? Please give your main reason.
   ....
   .
   ....
   .
   ....
   .

9. If you had to work in this way in the future, for which one of the following software engineering tasks would the Whiteboard be most suitable? (Please circle one option)
   Discussion/negotiation *Diagramming* *Presenting information*
   Shared document preparation
   Programming/testing/debugging
   Why do you think so?
   ....
   .
   ....
   .

....
.

10.If you had to work in this way in the future, for which one of the following software engineering tasks would The Whiteboard be least suitable? (Please circle one option)

Discussion/negotiation *Diagramming* *Presenting information*
Shared document preparation
Programming/testing/debugging

Why do you think so?

....
.
....
.
....
.

11.What did you like best about The Whiteboard? Please give your main reason.

....
.
....
.
....
.

12.What did you like least about The Whiteboard? Please give your main reason.

....
.
....
.
....
.

13.If you had to work in this way in the future, for which one of the following software engineering tasks would Video be most suitable? (Please circle one option)

Discussion/negotiation *Diagramming* *Presenting information*
Shared document preparation
Programming/testing/debugging

Why do you think so?

....
.
....
.
....
.

14. If you had to work in this way in the future, for which one of the following software engineering tasks would Video be least suitable? (Please circle one option)

Discussion/negotiation *Diagramming* *Presenting information* *Shared document preparation* *Programming/testing/debugging*

Why do you think so?

....

....

....

15. What did you like best about Video? Please give your main reason.

....

....

....

16. What did you like least about Video? Please give your main reason.

....

....

....

4. Overall Task Assessment

In this section, please choose JUST ONE of the answers given by circling it.

1. Did you understand the task? Not at all A little A lot
2. How successfully did your group complete the task?

1	2	3	4	5	6	7	8	9	10
Not at all successfully									Very successfully

3. How successfully do you think your group would have completed the task under face-to-face conditions?

1	2	3	4	5	6	7	8	9	10
Not at all successfully									Very successfully

4. How much more experience with the tools you used would be necessary for you to be able to use them optimally with this sort of task?

   0 hours 1 hour 2 hours 3 hours 4 hours 5+ hours

5. How much would you welcome the opportunity to gain this experience on your course?

   1 2 3 4 5 6 7 8 9 10
   Not at all ... Very much

6. In the first few minutes, what did you concentrate on?

   Mastering the technology Completing the task Both equally

7. During the final few minutes, what were you concentrating on?

   Mastering the technology Completing the task Both equally

8. Was the training useful? Not at all A Little A lot
9. Was the documentation useful? Not at all A Little A lot
10. Was the time sufficient? Too little Satisfactory Too much
11. What was the hardest part of the task?

    ____

    ____

12. Who was the most active in your group?

    ____

    ____

13. Did you agree with your group's conclusion? Yes / No
14. How satisfied were you with the session? Not at all A Little A lot
15. What improvements would you recommend for future users

    ____

    ____

    ____

Appendix B

SEGWorld Questionnaire

The questionnaire below was completed by each SEG student at the end of the project. The questionnaire is web-based and the results from each questionnaire were contained within an automatically generated email. The contents of this email were deposited into a spreadsheet.

A Survey of the usage of Basic Support for Cooperative Work (BSCW) within SEG

Information Survey

As part of a larger research effort on application of information and communication technology in higher education I am evaluating BSCW's contribution to the work you have undertaken in SEG work. Please answer as honestly as possible, we welcome this feedback whether critical or complimentary. The information you provide will be used for improving on current practices for future SEG's and in no way reflects your contribution to the group.
This survey is divided into the following sections:

- Group Information
- Functionality of BSCW
- Human Aspect
- Form Submission

Please fill in as many sections as possible. In the case of multiple choice, only one of the options may be selected. At the end of the form, click on the button "submit the form"!

Group Information

Please supply information about your group. (If your role within the group has alternated please indicate your last position)

SEG number

Role in group

EMail

Back to Top

Feature Checklist

In the following section, please select either Yes or No from columns 1 and 2 for each function used. In columns 3 and 4 please indicate Yes or No if you were AWARE of this function.

Have you used or were aware of the following functions provided by BSCW?:	Yes	No	Yes	N
Create	○	○	○	○
Read - (opens a document for you to read)	○	○	○	○
Edit	○	○	○	○
Delete	○	○	○	○
Cut	○	○	○	○
Replace	○	○	○	○
Version	○	○	○	○
Rename	○	○	○	○
Drop	○	○	○	○
Revised	○	○	○	○
Meeting	○	○	○	○
Attached Note	○	○	○	○

Software Information

Functionality

For what specific tasks did you mainly use BSCW?

Which functions of BSCW did you find most useful for these tasks?

Usability

At the beginning of the academic year you had to work through the BSCW tutorial. Was this exercise: (click only one)

- Very Easy
- Quite Easy
- Average
- Quite Difficult
- Very Difficult

Please enter any comments you may have regarding the tutorial

How would you rate general usability of BSCW? (please click only one)

- Very Easy
- Quite Easy
- Average
- Quite Difficult
- Very Difficult

Please enter any comments you may have regarding the usability of BSCW

Were there any other issues or difficulties in the functioning of the BSCW?

Back to Top

Human Aspect

In the following section, please select either yes or no for each question.

Do you:	Yes
require more mental effort to complete tasks using the software?	
need to invest a lot of time and effort learning the system?	
understand all (or most) of the system's functions?	
feel satisfied with the system?	
feel you have control when dealing with the system?	
find the system attractive and exciting to use?	
find the software compatible with your work environment?	
use the software frequently?	

Back to Top

FORM SUBMISSION

Thank you for taking the time to answer these questions.

Submit the form | Empty all of the boxes

Back to Top

Written by: *Sarah.Drummond@durham.ac.uk*, last updated 16/3/1998 .