IC3: Information and Communication integration using VCoIP between 3 collaborating parties

Jian Sun

A dissertation submitted as a partial requirement for the degree of Bachelor of Applied Science at the University of Otago, Dunedin, New Zealand

October 2006
Abstract

This study develops a new communication and collaboration supported tool – IC3. In particular, this tool is an integration of open source video conference over internet protocol (VCoIP) and virtual network computing (VNC) projects. This integrated system supports both virtual communication and collaborated web information sharing. In addition, it aims to facilitate greater eye contact and seating arrangements. The results from a set of heuristic evaluations show that the IC3 system is an effective communication and collaboration tool, and it does improve users’ eye contact and feeling of sitting around a table.
Acknowledgements

First and foremost I would like to sincerely thank my supervisor, Holger Regenbrecht, for providing all the guidance, encouragement and support throughout the year. This dissertation benefited significantly from your vast knowledge.

I wish to express my gratitude to other staff from throughout the Information Science department. I also would like to express thanks to all the honours classmates for the years of friendship. To Leanne Gundry, many thanks for your careful proofreading.

Lastly, I wish to thank my friend Estelle Sun and my parents for your continuous support and encouragement. I appreciate you more than words can express.
# Table of Contents

Abstract II
Acknowledges III
Table of Contents IV
Index of Appendix VI
Index of Figures VII
Index of Tables VIII

## Chapter one: Introduction 1
1.1 Background 2
1.2 Objectives 2
1.3 Motivation and Contributions 3
1.4 Outline of the Study 4

## Chapter Two: System Design and Hypotheses 5
2.1 IC3 System Overview 6
2.2 Hypotheses 8

## Chapter Three: Related Technology and Work Review 10
3.1 Groupware 11
3.2 Communication 12
  3.2.1 VoIP, VColP Software and Protocols 12
  3.2.2 Seating Arrangement 15
  3.2.3 Eye Contact 16
3.3 Collaboration 17
  3.3.1 Shared Workspace 17
  3.2.2 Telepointer 18
  3.3.3 Floor Control 20
3.4 Open Source Software vs Closed Source Software (proprietary software) 21
3.5 Client-server vs. Peer-to-Peer 23

## Chapter Four: IC3 Design and Implementation 24
4.1 Software Design and Implementation 25
  4.1.1 Possible VoIP Solution Evaluation and Implementation 25
  4.1.2 Possible Shared Browser Design and Implementation 31
  4.1.3 The Final System 33
4.2 Hardware Implementation and Physical Setup 34
4.3 System Limitations 35

Chapter Five: IC3 Usability Evaluation 36
  5.1 Testing Environment 37
  5.2 Scenario 37
  5.3 Methods 37
  5.4 Results and Discussion 39

Chapter Six: Conclusions 43
  6.1 Conclusions and Future Research Direction 44

References 45

Appendix 51
Appendix 1: Scenario for IC3 Testing
# Index of Figures

- **Figure 1: Graphic User Interface of IC3**  
  6
- **Figure 2: Cameras are Placed on the Two Sides of the Screen**  
  7
- **Figure 3: IC3 Gaze Awareness Support**  
  7
- **Figure 4: IC3 Video Input and Output Arrangement**  
  8
- **Figure 5: Two Separated Communication Channels for Each Client**  
  30
- **Figure 6: The Final IC3 System (one client view)**  
  34
- **Figure 7: The Proportion of Usability Problems vs Numbers of Evaluators**  
  38
## Index of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1: Comparison of Open Source VoIP Software</td>
<td>28</td>
</tr>
<tr>
<td>Table 2: Results for Scenario Questions 1-8</td>
<td>41</td>
</tr>
<tr>
<td>Table 3: Results for Scenario Questions 9-10</td>
<td>41</td>
</tr>
<tr>
<td>Table 4: Results for Scenario Question 11-12</td>
<td>41</td>
</tr>
</tbody>
</table>
Chapter One

Introduction

This chapter describes the background and the purpose of this dissertation. The importance and contributions are also outlined. Finally, the structure of the remainder of the dissertation is provided.
1.1 Background

Internet electronic mail (email), instant messaging and Voice over Internet Protocol (VoIP) are commonly used today as communication tools among geographically distributed people with computers and the internet. Email is a relatively mature and reliable tool that provides easy, private and costless communication between people. However, it is a lightweight and asynchronous approach to communication. With the rapid improvement of the capability of current internet infrastructure, more synchronous and data intensive communication services such as Video Conference over Internet Protocol (VCoIP) and remote control services such as the Virtual Network Computing (VNC) system are becoming more and more popular. In particular, VCoIP can provide more effective and content-rich communications within a small group of people. Moreover, they represent a rapidly growing area with great market potential, and push the development and improvement boundary of several key areas of visual communication such as data transport protocol, infrastructure design and efficient video/audio codec algorithm.

1.2 Objectives

The purpose of this study is to use standard personal computers (PCs) and low-tech equipments to develop a system that enables communication and information exchange between two or three parties using the internet. This system is named as IC3, which is an Integrated Communication and Collaboration tool for three parties. In order to achieve this objective, three steps are carried out in this study. The first is to find a VCoIP solution which allows two or three partners to communicate. The second step is to find a synchronized internet browser solution with multiple mouse pointers display (telepointers). Thirdly, these two components are modified and integrated to achieve an IC3 environment.
1.3 Motivation and Contributions

This study is motivated by the increasing demand for groupware that facilitates both effective communication and collaboration. This motivation is best illustrated by the following scenario. Adam, living in New Zealand, is looking for a second-hand car from an online trading website. However, he has little knowledge about cars. Adam has two friends, Bob and Chris, who have more knowledge about cars and would be able to provide good suggestions to Adam regarding various brands and models. The most effective way to seek their advices would be to physically meet together and talk to each other in person. Unfortunately, Bob is on a vacation in Hong Kong and Chris is currently working in the United States. Therefore, the three of them are geographically separated. When it is not economical or possible to meet face to face, such as in this scenario, different communication technologies often come into peoples’ mind.

In this particular case, a traditional communication tool (i.e. the telephone) would not work well. The main reason for this is that the telephone cannot provide a single communication channel among the three parties. It is also not optimal to use some asynchronous or light-weight communication approaches such as email, instant message or chat rooms. These approaches either do not facilitate real time communication or lack visual contact, which are essential for effective communication. Among various computer and internet driven communication choices, VCoIP is the only one that is capable of providing virtual present - the closest simulation to the physical present. VCoIP also eliminates the difficulties existing in a VoIP system such as the difficulty to follow who is speaking and lost status of the person you are talking to. Besides virtual communication, Adam also expects that the three of them can share the content and collaborate on a single browser so that they can look at the same information at the same time. To summarise, the motivation for this study is to integrate VCoIP and a shared web browser solution to facilitate both communication and collaboration among a small group.
This study contributes to groupware development by investigating the possibility of achieving an IC3 environment by integrating open source software. Also, this study intends to develop and examine a solution which provides better eye contact and seating arrangements for more effective communication.

1.4 Outline of the Study

This paper is organised as follows. Chapter 2 will give an overview of the IC3 system and introduce some of the basic ideas. In chapter 3, the related area and work regarding IC3 system will be reviewed. Chapter 4 will describe the software usefulness evaluation and system implementation. Chapter 5 will describe the usability test carried out for the IC3 system, and the relevant results and discussion. Finally, in chapter 6, conclusions will be drawn and future work will be addressed.
This chapter gives an overview of the proposed IC3 system. The system design decisions and objectives are also explained. Finally, the hypotheses of this dissertation are stated.
2.1 IC3 System Overview

The purpose of this dissertation is to develop a low-tech setup IC3 system that enables
communication and information exchange between two or three parties using the
internet.

The IC3 environment proposed is illustrated as follows. The graphical user interface
(GUI) elements of IC3 are designed at the initial stage as shown in Figure 1. IC3
contains one shared browser in the middle of the screen, two video output windows
on the left and right edges of the screen and optional self display at the bottom of the
screen. Users’ mouse pointers are linked to their video windows. This simply
designed system is expected to integrate video conferencing and collaborative web
browsing, and support multiple mouse pointers. Moreover, by using two video
cameras and placing them as close as possible to the remote parties’ video output
windows (Figure 2), IC3 is also expected to provide greater video communication
experiences as participants have greater eye contact (Figure 3), and the feeling of
sitting next to each other (around a table) is enhanced through this special camera
arrangement.

Figure 1: Graphic User Interface of IC3
Figure 2: Cameras are Placed on the Two Sides of the Screen (close to video output)

Figure 3: IC3 Eye contact Support
Figure 4: IC3 Video Input and Output Arrangement
2.2 Hypotheses

To achieve the proposed IC3 environment, a system that integrates video conference software, allowing participants to communicate, and a system which can provide a synchronised view of a web browser are required.

Since VoIP is one of the area that has been strongly supported by the open source community and the research project scope is constrained by limited time and resources, it is decided that instead of building a new fully working system from scratch or in a closed source manner, this study is focusing on the possibility of providing such an integrated system by using open source code software. A usability test on the system will be carried out to examine whether such a system will actually improve the user experience in communication and collaboration.

Therefore, the hypotheses of this study are:

**H1:** Is it possible to achieve the integrated communication and collaboration IC3 environment by integrating open source code projects?

**H2:** Does this IC3 system improve the users’ eye contact and increase the feeling of sitting around a table?

**H3:** Is this IC3 system an effective communication and collaboration tool?
Chapter Three

Related Technology and Work Review

In this chapter, the previous literature that relates to the development of the IC3 system is reviewed. It starts with an overview of groupware. Section 2 and section 3 look at communication and collaboration related issues respectively. Concerns associated with open source software are discussed in section 4. Finally, section 5 briefly addresses architecture related issues.
3.1 Groupware

Groupware is defined by Eills, Gibbs and Rein (1991) as a computer-based system that supports a group of people engaged in a common task and that provides an interface to a shared environment. With recent advances in network infrastructure and computing power, desktop video conferencing and groupware systems are rapidly evolving into technologically viable solutions for remote communication, collaboration and coordination. Groupware not only reflects a change in emphasis from using the computer to solve problems to using the computer to facilitate human interaction, it is also regarded as a highly demanding tool since a significant portion of a person’s activities occurs in a group, rather than an individual context (Eills et al, 1991). Conventional communication systems such as email, newsgroups, document management systems and defect tracking systems are asynchronous groupware. They involve neither immediate user interaction nor object sharing. This contrasts with a synchronous groupware system, which allows physically separated users to interact with one another with shared computational objects in real time (Phillips, 1999). Therefore, it is argued that successful synchronous groupware should be able to provide effective communication. The effectiveness of these communications can be further improved by proper seating arrangements and better eye contact support. Similar to communication, collaboration is a cornerstone of group activity. Effective collaboration demands that people share information. For that reason, a shared workspace that offers a group context and explicit notification of each other’s actions (gestures) becomes very necessary. The effectiveness of communication and collaboration can be enhanced if a group’s activities are coordinated. Without coordination, a team will easily engage in conflicting actions (Eills et al, 1991).

In a real working situation, people are kept aware of what others are doing by seeing what others are working on or by notification from other people. This helps to organise people’s own work and co-operation with others. This important aspect of groupware is referred to as workspace awareness (Gutwin, Greenberg & Roseman,
Information available in and throughout the physical workspace allows people to maintain awareness of others’ locations, activities and intentions. According to Gutwin et al (1996), when shared activity moves from face-to-face to distributed groupware, many things change that impair people’s ability to maintain workspace awareness. For instance, the perceivable environment shrinks, the communication means are weakened, physical interactions need to be replaced by a computerised way and so on. They further point out that the groupware designer must try to recreate the conditions and cues, that allow users to keep up a sense of workspace awareness.

3.2 Communication

3.2.1 VoIP, VCoIP Software and Protocols

In recent years, VoIP and VCoIP software have become more and more popular as a communication tool among geographically distributed people. VoIP (also called IP Telephony, Internet telephony, and Digital Phone) is the routing of voice as data packages over the Internet Protocol or any other IP based network. VCoIP is used for routing both audio and video data over the Internet or IP based network. The term VCoIP comes after the video function is added to the existing VoIP infrastructure and is being widely used in video conferencing tools, thus VoIP is quite often used to refer to both terms. This creates a lot of ambiguity and difficulty when working on this topic, which will be discussed in detail later.

In comparison to the Public-Switched Telephone Network (PSTN), a number of factors can influence the adoption of VoIP. First and foremost, the cost of using the VoIP network is as much as half that of the traditional PSTN network (Weiss & Hwang, 1998). This cost saving is due to the efficient use of bandwidth - for example, voice compression is employed in VoIP. As a result, during a lot of today’s cheap international telephone calls, the voice is actually carried by the VoIP network. Secondly, VoIP provides the possibility to make a video call (VCoIP) or to be combined with other computer based services such as messaging, white board and file
sharing in more collaborative and multimedia environment, whereas the PSTN does not have such capability. Thirdly, VoIP can include features like multiple endpoint meeting, call forwarding and caller ID with little cost due to the flexibility of the overall infrastructure (Varshney, Snow, McGivern & Howard, 2002).

However, there are many challenges faced by VoIP. Most critically, VoIP relies heavily on the internet. Consequently, this leads to issues of latency (network conjunction and loss of packet), network reliability, security and emergency calls (as not all VoIP networks are connected to emergency services and are unable to identify the location of the calls due to the inherent mobility of IP end points). The quality of service also relies on the hardware environment such as the computer, webcam and microphone. Finally, as computers require electricity in order to work, not like PSTN which uses a separate power source, the VoIP service is vulnerable to any sort of power outage. However, some of these issues can be resolved. For example, the security problem can be eliminated by better system design: the data can be encrypted at the endpoints and throughout the network, and an additional level of identity authorization features can be introduced through the use of passwords (Kolbehdari, Lizotte, Shires & Trevor, 2006). Because VoIP is part of the proposed system, the limitations VoIP is currently facing also apply to the proposed IC3 system.

Just like people have to understand a universal language in order to communicate with each other, a universal VoIP protocol is required for different VoIP clients. The two major competing standards for VoIP are the Internet Engineering Task Force’s (IETF) Session Initiation Protocol (SIP) standard and the International Telecommunication Union’s (ITU) H.323 standard. Initially, H.323 has been the most popular protocol, although its popularity has decreased due to its poor traversal of NAT (network address translation) and firewalls. ITU approved the first version of the H.323 standard in 1996. The current version (6) was officially approved in June 2006. Many video conference applications deploy the H.323 protocol. Examples are Microsoft NetMeeting which utilises H.323 for video conferencing and shared collaboration and
Ekiga, known as GnomeMeeting, which uses an openH.323 implementation.

The other common standard - SIP - is a signaling protocol for the establishment of communication sessions between SIP endpoints. SIP differs from other communication protocols by its strong industry support, multi-vendor integration at the application layer, modularity, and common standards (Kolbehdari et al, 2006).

Both H.323 and SIP are protocols used to define and manage VoIP connections. The underlining protocols in charge of multimedia data transfer are the Real-time Transport Protocol (RTP) and the Real-time Transport Control Protocol (RTCP) protocol.

RTP defines a standardised packet format for delivering audio and video over the Internet. The only standard it obeys is that User Datagram Protocol (UDP) communications are done via an even port and the next higher odd port is used for RTCP. RTCP is a sister protocol of RTP, and is used to provide control information of RTP. Although there are no standards assigned, RTP is generally configured to use ports 16384-32767 (Capilla & Dueñas, 2002). It can carry any data with real-time characteristics, such as interactive audio and video. RTP is used in conjunction with both H.323 and SIP to form the technical foundation of VoIP industry.

Another common but not strongly supported VoIP protocol is IAX (Inter-Asterisk eXchange). Now it is most commonly referred to as IAX2, the second version of the IAX protocol. After Mark Spencer created Asterisk, the open source PBX (Private Branch Exchange), he started to see how this lack of simplicity could be a huge barrier to the VoIP market. In response, he created a new protocol- IAX. The goals for this new protocol are to minimise the bandwidth for signaling and media, and provide internal support for network address translation (NAT) transparency. Instead of using RTP, IAX uses UDP over a single Internet port (Port 4569 for IAX2) to transmit and receive signaling and media. Therefore, IAX provides an easier way for firewall
traverse and uses much less overhead than RTP. IAX can be used with any types of streaming media, including video, but is targeted primarily at the control of IP voice calls. However, the limitation of IAX2 is that it is only used by Asterisk and is only supported by software such as Yate, SofaSwitch, OPAL and FreeSwith. Thus, IAX based VOIP software is not a major target of the VoIP software investigation.

3.2.2 Seating Arrangement

Parsons (1982) summarises several key aspects affecting effective communication. The sending and receiving skills are two important ones. The ability to express yourself clearly and efficiently, and the ability to understand other people have always been associated with success, whether in business or life in general. He also identifies that the setting and environment in which the communication is taking place must be given serious consideration. For instance, the room you are in may be too big, too small, too cold, too hot, too noisy, or for various reasons, uncomfortable. If you are giving an information session, then theatre style seating is appropriate. If you want participation and collaboration, then a round table seating arrangement is more suitable; as everyone can be seen and heard more readily. Further, group size and atmosphere are also two important considerations identified by Parsons (1982). Many of these factors cannot be controlled or improved by a communication support system. What can be controlled for however are group size and seating arrangement. As the proposed IC3 system is aiming to provide a communication and collaboration environment for 2 or 3 people, it is also suggested that a round table seating arrangement could be simulated through special camera and video stream placement.

Barbour and Barbour (2001) suggest that teams can use seating arrangement to establish a sense of equity among meeting. Dabkowski (2004) supports this view and further points out that using a round conference table eliminates a sense of hierarchy among participants that may otherwise exist when particular team members are seated at the head of a rectangular table. Wilson (2004) investigates the practices of
discussions with students via videoconferencing, and concludes that the specific seating arrangement is important for improving eye contact and interaction.

Research on the social use of space suggests that spatial arrangements exert an important influence on the course of interactions. Batchelor and Goethals (1972) examine the relationship between task instruction and seating arrangement. Their results show that the type of work has a significant effect on the spatial arrangement. When the instruction given to participants is for a collaborative work, a circular seating arrangement is more preferred by participants. Hendrick, Giesen and Coy (1974) and Giesen and McClaren (1976) also show a clear preference for circular arrangements to facilitate interaction.

### 3.2.3 Eye Contact

Many studies address the importance of eye contact or gaze awareness in communication and group interaction. It is estimated that 60 percent of conversation involves gaze, and 30 percent involves mutual gaze (Sellen, 1992). Garau, Slater, Bee and Sasse (2001) summarise that eye contact serves at least five distinct communicative functions: regulating conversation flow, providing feedback, communicating emotional information, communicating the nature of interpersonal relationships and avoiding distraction by restricting visual input. Takeuchi and Naito (1995) further demonstrate the importance of eye contact in social interaction by arguing that eye contact can indicate whom a message is directed to.

Lack of eye contact seems to be one of the most difficult problems to overcome in a video conferencing system. In a traditional VCoIP system, the camera is usually placed on top of the monitor. Thus, when a conferee looks at the displayed video image of the interlocutor, an angular deviation between the camera shooting axis and the line of sight of the interlocutor exists. This is recognised as eye contact parallax. This deviation impedes interlocutors in identifying out true viewing directions, and
eye contact between conferees will never be experienced (Lui, Beldie & Wopking, 1995).

Many studies show that an absence of individual eye contact deteriorates the communicative presence (Bocker & Muhlbach, 1993; Hopf, Runde & Bocker, 1994). To eliminate the angular deviation, various systems have been developed and tested. One example is the “Multi-Attendant Joint Interface for Collaboration” (MAJIC) system developed by Okada, Maeda, Ichikawa and Matsushita (1994). MAJIC achieves gaze awareness by specially arranging video cameras and projectors to display life-size video images of participants onto a large curved screen without boundaries between them. Another video conferencing example aiming to provide eye contact is documented by Yang and Zhang (2002). They describe an eye-gaze correction system using a pair of calibrated stereo cameras and a personalised face model to track the head pose in 3D. Both of these approaches are expensive to set up. A cheaper pure software approach was introduced by Gemmell, Toyama, Zitnick, Kang and Seitz (2000). Their approach is similar to Yang and Zhang’s (2002); however the drawback of this approach is that it uses a synthetic view to simulate a natural gaze awareness video conversation which still seems unnatural.

### 3.3 Collaboration

#### 3.3.1 Shared Workspace

Many groupware systems provide an environment for collaboration called a shared workspace. This is a bounded space where people can see and manipulate artefacts related to their activities. In the real world, a shared workspace is simply a physical space where people can undertake joint activities.

In a real-time synchronised situation, according to Lauwers and Lantz (1990), generally there are two categories of approaches to developing synchronous collaborative applications. In the first category, collaboration awareness, applications
are specifically designed for simultaneous use by multiple users. The shared workspace and multiple user collaboration mechanisms are designed at the very beginning and built into the applications. In the second category, collaboration transparency, single-user applications are shared by collaboration-aware mechanisms that are unknown, or transparent to the shared applications and their developers. This is a more generic way of application sharing such as screen sharing and application view sharing. The screen sharing approach replicates one user’s desktop onto multiple workstations so that all users see and interact with the same view. This allows singular user’s applications to be shared to multiple users simultaneously without modification. On the other hand, collaboration awareness tools provide a richer user interface that support collaboration more effectively. For example, these tools might support different views based on the roles of the individual user, or allow users to work on separate portions of the shared artefact while keeping its state consistent (Ellis & Gibbs, 1989).

The first screen sharing was developed by Engelbart and English (1968) as part of the oNLine System (NLS). Since then, screen sharing has been used to allow multiple users at separate locations to view a common desktop or terminal, or to allow one user to remotely control a remote computer. Commercial products that support screen sharing include Microsoft’s NetMeeting, Symantec’s PCAnywhere and most recently the VNC system.

3.3.2 Telepointer

Studies of small face-to-face groups working together over a shared work surface reveal that gesturing comprises about 35% of the group’s activities (Tang, 1991). A gesture is a rich communication mechanism. Through gesturing, participants are able to indicate relationships between the artifacts, draw attention to particular artifacts, show intentions about what they are about to do, and suggest emotional reactions (Greenberg & Roseman, 1999). In a real-time desktop based video conference, the
ability to point at shared objects on the display can also greatly enhance communication between conference participants (Lauwers and Lantz, 1990). Many groupware systems now use telepointers (also known as multiple cursors) to provide a simple but reasonably effective mechanism for communicating gestures (Hayne, Pendergast & Greenberg, 1994).

According to Lauwers and Lantz (1990), telepointing and annotation facilities are most easily implemented if the underlying window system supports transparent windows. In this case, the annotation and telepointing tools can create transparent windows on top of existing windows, and use these transparent windows as their drawing canvases. Unfortunately, no existing window system supports transparent windows yet.

A number of prior studies in this area focus on the design, implementation and usefulness discussion of multiple mouse cursors working on a single display device. This kind of application is termed as Single Displayed Groupware (SDG). Early efforts to implement multiple input devices include Bier’s MMM (1991), Hourcade’s MID (1999), Tse’s SDGTookit (2004), and open source project CPNMouse (CPNMouse, 2006). The former three implementations face the same problem, that the Windows operating system (OS) elements only provide listener for a single mouse pointer, thus these SDG implementations involve the creation of own applications and elements that facilitate listening to multiple mice. CPNMouse uses a common mouse driver to cover original mice drivers and register multiple mice to a small application. Although ordinary Windows OS elements such as scroll bars and buttons are able to respond to each of the registered mouse, the performance is very poor and unstable. Most importantly, these implementations focus on supporting multiple devices on a single display or a single system rather than displaying and managing multiple mouse cursors from clients over the network. Nevertheless, researchers in this area struggle with the potential value of their work as they face the difficulty of compelling a multiple input devices support application. And for those who do have this kind of
application, they often come up with positive results on the tasks performed among the collaborative parties (Bier & Freeman, 1991; Hourcade, Iyer & Bederson, 1999).

### 3.3.3 Floor Control

Floor control is often recognized as one of the most important attributes or requirements for groupware. Dommel and Garcia-Luna-Aceves (1997) define floor control as a technology that deals with conflicts within shared workspaces. It helps coordinate and mediate access to a shared workspace, by regulating turn taking in conversations or access of controls over shared objects. They further summarise that:

Floor control encompasses the following tasks: …, granting or denying floor requests according to the enacted service policy, tracking the status of shared resources at all connected sites, relaying floor requests between sites, managing temporary access rights to data, authorizing or denying usage according to session control information, and broadcasting or multicasting changes of floor control states to collaborating members. (Dommel & Garcia-Luna-Aceves, 1997, p. 27)

The benefits of floor control from a user and system perspective are identified by Dommel and Garcia-Luna-Aceves (1999a). They state that, from user level, misbehaviors are avoided “by shifting the risk of collisions from data to signaling” (p. 20). From system level, floor control can “orchestrate intelligent allocation of scarce shared resources based on user input” (p. 20). They also point out that social and cultural issues of communication may be involved in floor control deployment.

Dommel and Garcia-Luna-Aceves (1999b) particularly address the significance of floor control for “tightly coupled” sessions, which often involve ample interactions and collaborations. They further point out that floor control can either be deployed by human moderators or by the system through using prediction, filtering and reservation.
3.4 Open Source Software vs Closed Source Software (proprietary software)

The term open source software refers to the basic level of open source which simply means software for which source code is open and available. However it is necessary to have a closer look at different categories of open source software.

The “open source software” definition generated from open source movement is specified as computer software whose source code is available under a copyright license (which defines the privileges and restrictions a licensor must follow) that permits users to study, change, and improve the software, and to redistribute it in modified or unmodified form. Examples of open source licenses include Apache License, BSD license, GNU General Public License (GPL), GNU Lesser General Public License (LGPL), MIT License, Eclipse Public License and Mozilla Public License. The open source software that are useful for this study either fall under the GPL or LGPL category.

The GNU General Public License has been the most prominent. The author retains copyright and permits redistribution and modification under terms to ensure that all modified versions remain free. The GPL also requires works derived from GPL licensed work to be licensed under the GPL.

The GNU Lesser General Public License is designed as a compromise between the strong copyleft GPL and simple licenses such as the BSD license and the MIT license. The LGPL places restrictions on the program itself but does not apply these restrictions to other software that merely link with the program. A shared library mechanism is the most common method that satisfies LGPL. It is noticed that LGPL is “GPL-compatible” which means LGPL licensed software can be converted into GPL licensed software.
Another category of open source software is free software. The “free” refers to “freedom” not “zero price”. The definitions of open source software and free software are almost identical. Also, they share an almost identical set of licenses. In practice, there are very few cases of software that are open source software but not free software, and vice versa. Both groups, Free Software Foundation (FSF) and Open Source Initiative (OSI), claim the other’s definition is ambiguous. However, their common enemy is proprietary software and commercial software which normally come with some or none of the freedom open-source software or free software would provide.

Public domain software and copyleft software are the two kinds of software where authors have abandoned the copyright. Since public domain software lacks copyright protection, it may be freely incorporated into any work and redistributed under any license, whether proprietary or free. However, the source code of public domain software and copyleft software may or may not be available.

For the purpose of constructing the proposed IC3 system with minimum effort, either open source software, free software, public domain software or copyleft software is acceptable. This also aligns with one of the most important objectives of the open source software movement which is for code reusing and knowledge sharing.

In comparison to proprietary software or closed source software development style, open source software or open source software development style has many advantages, such as faster system growth, more creativity, more modular, fewer defects and successes because of their simplicity (Paulson, Succi & Eberlein, 2004). Advocates of open source also argue that its projects provide a good starting point for commercial software. It is especially important for small or medium companies with limited human, technology, and capital resources, as for these companies, starting with open source may be their only choice. However, an awareness of the problems and development difficulties with open source software is needed. Since no one is
responsible for open source software, there is little incentive to continuously work on it, thus there is no guarantee the software will work or the bugs in the software will be fixed. Further, as nobody takes the responsibility for negligence, there is no guarantee for maintenance and support. Therefore besides a few well known and well developed open source projects such as Mozzilla firefox, GIMP, and eMule, the majority of open source projects are facing problems such as low development efficiency, code redundancy, lack of commenting and documenting, low readability and vital system errors. In addition, a problem is that people think differently, and this is also true for software developers. Therefore, when working with open source, software developers have to cope with other people’s design and coding styles which could be very different and difficult.

3.5 Client-server vs. Peer-to-Peer

In client-server architecture, all data must be propagated to a server machine that is further propagated to all clients. The advantages of a server-based solution include group, security, scalability, reliability, and better support and management in maintaining the corporate environments (e.g. bandwidth control, firewalls, NATs, etc). However, these features and functions, as well as data propagated from clients, are largely subject to the availability of the server. By contrast, peer-to-peer systems allow data to be propagated directly from one peer to another peer. Thus, it reduces data transfer pressure on the server side, and data under peer-to-peer system architecture can transfer faster than data under client-server architecture. The multicast architecture is best used in a large-scale, multi-end points and sub network environment.

For the proposed IC3 system, because of the data intensive nature of VoIP and shared browser view, the peer-to-peer architecture is preferable. However, the final choice of architecture largely depends on the solution’s architecture itself.
Chapter Four
IC3 Design and Implementation

The chapter describes the detailed system component choices, problems encountered during the evaluation of various open source software, and the final implementation of the IC3 system. At the end, hardware implementations are described and current system limitations are identified.
4.1 Software Design and Implementation

As described in Chapter 2, the IC3 has two major parts: the VCoIP part for communication and the shared web browser part for information sharing and collaboration. This section describes the choices of software solutions for both parts and the integration of the final system.

4.1.1 Possible VoIP Solution Evaluation and Implementation

The VCoIP part of this study focuses on finding a solution that satisfies the following requirements. Firstly, the solution has to be developed on an open standard such as ITU’s H.323 or IETF’s SIP standards. Secondly, it has to be a Microsoft Windows operating system based solution. As Microsoft still has the largest market share in PC’s operating systems, a communication and collaboration tool such as IC3 should target as many users as possible. Thirdly, there should not be any critical bugs existing in the VCoIP system which prevents the whole system functioning, or at least the system must be able to be fixed up to an acceptable level in order to be integrated into the IC3 system. Lastly, the solution has to be successfully built in the current software development environment, which is Microsoft Visual Studio 2005 (MSVS 2005).

As VCoIP is a branch of the VoIP industry, the investigation of the communication part of the IC3 starts from exploring the potential VoIP solutions. It is found that many of the current open source VoIP solutions fail to meet the above requirements in order to be integrated into the IC3 system (Table 1). Problems encountered are summarised as follows:

1. Platform incompatible. Many current open source VoIP software only run on an operating system other than Microsoft Windows. For example, Ekiga, known as the second version of Gnomemeeting, is a very well developed and supported open source VoIP solution. However it only runs on X windows OS. Other examples include Kphone and Linphone.
2. Stopped developing a long time ago and critical bugs exist. Some open source VoIP solutions are developed as sample applications to demonstrate the early version of the H.323 standard. Thus these projects are either unfinished, finished with limited functionalities or finished with usage problems. Examples of this are openPhone and Cphone.

3. Build failure. Some solutions failed to be built by the development tool – MSVS 2005. Examples include openPhone, Freeswitch and Cphone. However this could due to limited knowledge about how to build the solutions. Also it addresses one important open source software problem-lack of documentation.

4. Backwards incompatible. Old version programs such as myPhone are not compatible with the new version of the run time libraries (openH323.dll & PWLib.dll). Furthermore, the related old versions of libraries are not available any more.

5. No video support. Video is seen as an additional feature upon existing VoIP infrastructure. Many VoIP solutions are designed as voice only, therefore do not support video feature such as ZAP, sipXphone and YATE.

6. Open source client, proprietary server. This is a particular problem for SIP protocol based VCoIP solutions, because SIP uses client-server architecture. One example is openWengo.

7. Still under heavy development. There are a couple of open source solutions still under development, therefore desired features are not available yet. Examples are openWengo and YATE.

8. Poorly designed solutions. This is another common open source software problem.
For example, VideoNet, an open source VoIP solution, is released with no audio, video or internet related configuration at all
### Table 1: Comparison of Open Source VoIP Software

<table>
<thead>
<tr>
<th>Platform</th>
<th>Incompatible</th>
<th>Build Failure</th>
<th>Backwards Incompatible</th>
<th>No Video Support</th>
<th>Open Source Client, Proprietary Server</th>
<th>Still Under Heavy Development</th>
<th>Poorly Designed</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenPhone</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ekiga</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenWengo</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Freeswitch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>YATE</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>SipXphone</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zap</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPhone</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MyPhone</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TalkEZ</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>VideoNet</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VIS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

IC3
Because of these problems, the open source VoIP solution this study uses is an open source solution called VIS (115studio, 2006). VIS has been released as a sample application for the demonstration of the VIS H.323 library. It is a MS Windows OS based solution that supports both audio and videoconference through H.323 standard. It has a simple user interface with sufficient functionalities. More importantly, VIS seems to be more stable than many other open source VoIP solutions, and is successfully compiled in the Visual Studio 2005. Moreover, VIS has some important features such as audio and video configuration, network configuration, peer-to-peer connection and remote/local video display. The audio codecs VIS supports include: ITU’s G.723.1 (default), G.729, G.729A, GSM-06.10, G.711 A-law and G.711 µ-law. The video codec used by VIS is H.263. Despite of these advantages, one shortcoming of VIS is that it has no documentation except the comments in source code. Another limitation is that the VIS application is published as open source but the VIS H.323 library is not. Without purchase of the authentication code, the VIS H.323 library will automatically shut down the video and audio connection after 3 minutes.

One vital problem identified in the implementation of VCoIP is that normal VoIP software are only designed to manage one communication channel, while the proposed IC3 system aims to send two audio and video streams (captured by the left and right cameras) to separate parties, as well as receive two audio and video streams from these two parties. This requires two separate communication channels for simultaneous data distribution. In order to achieve this, the VIS application end point has been replicated and further assigned to listen to different ports. As shown in Figure 5, each client communicates to the other two clients via two separate channels.
Figure 5: Two Separate Communication Channels for Each Client

Note to Figure 5:
1. 1159, 1720 and 1723 are examples of port numbers, 1720 is the default port for H.323.

If the IC3 system is designed with voice support only, the client-server architecture could be used to collect all voice data to the server and multicast it back to clients. However, with two separate video streams associated with each client, it is not realistic to use client-server architecture for the distribution of multimedia data. As opposed to a pure peer-to-peer architecture for H.323, SIP uses a mixture of client-server and peer-to-peer architecture. In SIP based VoIP solutions, client-server is used for the management of user names and locations, and peer-to-peer architecture is used for data communication. In this case, if SIP based solutions are chosen, an extra server needs to be set up. This extra server seems to be redundant and makes the whole system more complicated.

Three modifications have been made to the VIS application: the GUI language is translated from Chinese to English, the start-up positions of the remote peers’ video output windows are set to be at the left and right edges of the screen and lastly the parent-child window relationship has been broken up so that the main application can be hidden (minimised) while the video window is still visible. Few other modifications are needed on the VIS application.
4.1.2 Possible Shared Browser Design and Implementation

For the shared web browser part, VNC (Virtual Network Computing) is one of the solutions. VNC is a desktop view sharing and remote control system that uses the RFB (Remote Frame Buffer) protocol. VNC has two parts, a client and a server. The server is the program on the machine that shares its screen, and the client (or viewer) is the program that watches and interacts with the server. In addition, multiple client computers can share the view and control of one server computer simultaneously, thus supporting application sharing in the Computer Supported Cooperative Work (CSCW) style (Richardson, Stafford-Fraser, Wood & Hopper, 1998). Since VNC uses a lot of network bandwidth, various methods (encodings) are used to reduce the bandwidth.

There are several reasons to use VNC. Firstly, many commercial products announce that they provide application sharing. They are actually using very similar techniques to VNC – remote computer desktop view sharing and remote control. Examples of these products include:

- NetMeeting 3.0 (stand alone application)
- BeemYourScreen (desktop view sharing application very similar to VNC)
- Glance (desktop view sharing application built in web browser)

Secondly, because VNC is a desktop view sharing system, thus the integrated IC3 system can share the view of applications other than a web browser. The whole system can be easily extended to provide a collaborative environment for purposes other than web information retrieving. For example a software development tool such as Visual Studio can be shared by the IC3 system to facilitate pair programming.

Thirdly, many open source VNC software are well developed, documented, maintained and supported. Finally, there is little choice as other shared browser implementations are either still under research or closed source projects. An example of this is Esenther’s Collaborative Web Browsing (2002) solution, which uses a Java
servlet to periodically pull changes from a shared web browser and send to clients. Although single web information can be shared across clients, the periodical nature of this approach means it can never achieve a real-time collaboration environment. Another solution is P2P Shared Web Browser Using JXTA by Nakamura, Ma, Chiba, Shizuka & Miyosh (2003). The theory of this particular implementation is published but the final product and source code are not available.

In order to fulfill the objective of the IC3 system there are a few requirements for the VNC system:

1) It has to be a Windows OS based solution.
2) It has to be an open source code solution.
3) It has to be able to share a particular window (for shared web browser) or a particular area of the desktop other than the full desktop view.
4) It should support remote control for collaboration.

TightVNC is chosen to be integrated in the IC3 system, as it is a very well developed open source VNC system. It provides many useful features such as bandwidth-friendly "Tight" encoding, password authentication, window sharing. A particular GUI component of an application you want to share can even be specified, such as a panel inside a window.

However there are still some limitations of the current TightVNC system. Firstly, the VNC system will not trace the shared window movement or resizing and adjust the corresponding shared area. Once the sharing area is confirmed and VNC clients are connected, the shared application window on the server is better not to be resized or moved while the system is running. Otherwise, the shared area needs to be manually adjusted to the right size or position. Secondly, the TightVNC does not work with the scroll wheel on the mouse very well. When the scroll wheel is used for web browsing, the shared content on the VNC client machine starts to blur.
One of the goals of the IC3 system is to achieve the display of multiple mouse pointers and to distinguish them, for example in different colours or linking to the video window of the user. However this could be seen as a common limitation for current Windows based VNC systems. At present, Windows based VNC services can only achieve multiple clients sharing the same mouse pointer on the server because the standard Windows system provides only one cursor. Users either have to share the concurrent control of the server mouse pointer or use some floor control mechanisms to take turns to control the server mouse pointer.

One solution that fulfills the necessary requirements exactly is called the “Mighty Mouse” multi-screen collaboration tool, and is based on a VNC system (Booth, Fisher, Lin & Argue, 2002). However the software and its source code are not available at the moment. A patch for TightVNC called collaborative VNC (Collaborative VNC, 2006) claims to be able to provide a display of client mouse cursors and floor control. Nevertheless the patch is only for Unix/Linux version of TightVNC. The patch for Windows version of TightVNC is still on the authors’ to do list. The development of this kind of patch requires a clear understanding of TightVNC and the related RFB protocol for VNC system. Thus this goal is unachievable within the given timeframe.

4.1.3 The Final System

To integrate the two major parts (VCoIP and VNC) into one single system, these two solutions have to be developed in the same programming language. There are a few VCoIP and VNC solutions that are developed in Java. However the main stream is still C++ for Windows applications. For the chosen solutions in this study, both VIS and TightVNC are programmed in C++; therefore a common programming language is not a concern in this situation.

Finally the IC3 environment is accomplished by integrating VIS and TightVNC, and the integration is achieved by modifying the source code. The VIS for VCoIP
communication remains a peer-to-peer architecture, and the TightVNC for application view sharing remains in client-server architecture. In order to manage two separate VCoIP communication channels, the VIS part has been replicated and then integrated with either TightVNC server or TightVNC client. Due to the limitation of VNC itself, the telepointers and floor control are not supported (Figure 6).

Figure 6 The Final IC3 System (one client view)

4.2 Hardware Implementation and Physical Setup
Since this study focuses on a low-tech setup, the IC3 system requires only the following hardware: a server computer with Microsoft OS, a tag supported web browser application (such as Mozilla Firefox) installed (to be the IC3 server), one or two client computers with Windows OS (depending on two or three parties’ communication), two web cameras (supporting QCIF 176 x 144 video format) and one headset for each participant and network connection.
4.3 System Limitations

1) The goal of integrating telepointers into the IC3 system has not yet been achieved. This is purely a software programming issue. As stated, similar implementation has been achieved in a patch to existing Linux version of TightVNC called Collaborative VNC. In theory, the VNC client side needs to be modified in order to display other client’s pointers and also to handle appropriate floor control such as requiring and giving away control. On the server side, the modifications include managing the pointers’ position and control information and sending out to clients. In the current implementation, multiple clients have to share the control of the same server-side mouse pointer. Negotiations of who takes the control need to be made in advance.

2) The setup of the current system is complex. At present, each VCoIP call as well as client-to-server VNC connections need to be made manually. This process involves finding remote peers’ IP addresses, making 2 separate VCoIP calls to each remote peer, setting up the VNC server and initiating VNC connection. A single and simpler system workflow needs to be cautiously designed and implemented for such a system.

3) Because the implementation involves two audio channels (to each individual peer), two video channels and one VNC channel for each peer, it relies significantly on the network environment it is running. The existing implementation has only been tested on a local network. Therefore, a lot of heavy testing needs to be carried out to assess the usability of this system on the public internet.

4) Some other usability issues may potentially exist, since many users are not familiar with this kind of system. For example, the video communication may distract users focus on the content. Users may prefer a maximised web browser window rather than a non-maximised content window placed in the middle of the screen. Many of these usability issues may be largely influenced by users’ personal preferences and habits.
This chapter firstly provides a brief description of the testing environment. Then an illustration of a simple scenario is given for IC3 testing. After that, method and procedures for conducting the evaluation are described. Finally, the results are reported.
5.1 Testing Environment

A preliminary trial is conducted to get a sense for the overall value of this IC3 system. While this study has not yet performed enough trials to draw strong conclusions that are statistically significant, the trials do allow a qualitative understanding of how IC3 might fare in distance communication and collaboration. In particular, the system evaluation can help answer the questions such as: does this IC3 system improve users’ eye contact and increase the feeling of sitting around a table? Is the IC3 system an effective communication and collaboration tool?

As mentioned in section 4.1.1, the integrated VIS has a “3 minutes” limitation. Therefore, in order to successfully carry out the trials, an IC3 system hi-fi prototype is used. This hi-fi prototype is a combination of the unlimited time version of VIS and TightVNC. In terms of the hardware, the testing environment is set up exactly as described in section 4.2 (Hardware implementation and physical setup), except that the tests are carried out in a local network environment.

5.2 Scenario

A scenario is developed for the IC3 system tested in this study. This scenario assigns roles to three participants. Participant A wants to buy a car from an online trading website - trademe (www.trademe.co.nz), and would like his friends B and C to provide some suggestions. A has a budget of $5000, and could borrow another $1000 from his parents if he really likes the car. For the preferences of the car, B and C have to ask A. The possible topics that could be discussed are car make, model, colour, year, mileage and other features.

5.3 Methods

The original use of heuristic evaluation such as Nielsen’s (1994) methodology was to evaluate the usability for single user applications. However more recent studies have applied the technique successfully to groupware applications. Baker, Greenberg &
Gutwin (2001) adapt Nielsen’s (1994) heuristic evaluation methodology to help inspectors identify usability problems within groupware systems, and their framework has proven to be effective at identifying collaboration and communication usability problems. This study uses a modified Baker et al (2001) heuristic evaluation to assess the usability issues for IC3 system.

Nielsen (2001) develops a curve showing the proportion of usability problems found by heuristic evaluation using various numbers of evaluators (Figure 4). Since different evaluators tend to find different problems, by aggregating the evaluations from several evaluators, a substantially better performance can be achieved. Nielsen (2001) suggests that the use of at least five evaluators can detect approximately 75% of the problems. Similar results are also shown by Baker et al (2001), who use heuristic evaluation to evaluate groupware applications. Their study indicates that a single evaluator on average only discovers around 22% of the problems in a system. Performing the evaluation with 3-5 evaluators can locate 40%-60% of the problems. To locate 75% of the problem, which is a reasonably high proportion of the usability problems, between 6 and 10 evaluators are required. This study employs 9 evaluators to conduct the IC3 system evaluation due to the above arguments.

Figure 7: The Proportion of Usability Problems vs Numbers of Evaluators

A modified version of Baker et al’s (2001) groupware heuristic evaluation is used in this study to assess the usability issues for the IC3 system (Appendix 1). There are, in
total, 12 evaluation criteria using 5 unit scales and 1 free style question. The first 8 criteria are directly selected from Baker et al’s (2001) groupware heuristic evaluation. Because these criteria are general evaluation criteria for groupware, the importance of the criteria is firstly evaluated. From this how the targeting system satisfies these criteria is assessed. The following 4 criteria specifically assess the potential usability issues of the proposed IC3 system.

The evaluators are asked to carry out the task described in the scenario using the IC3 system and then answer the evaluation sheet based on their understanding and feeling of the system.

5.4 Results and Discussion

The results of the evaluation are categorised into two parts and analysed separately. The first part contains the results obtained from Baker et al’s (2001) groupware heuristic evaluation criteria (questions 1-8). The second part illustrates the results for questions 9-12, which are specifically designed for this study.

The results (Table 2) regarding groupware usability (questions 1-8) can be further split up into three categories: (1) Important criteria that are largely satisfied by targeting system (criteria 1, 2, 3, 4, 7), (2) Important criteria that are poorly satisfied by targeting system (criteria 5, 8), and (3) Importance of the criteria that is arguable (criteria 6). The two most important requirements found from evaluators which are also well satisfied by the IC3 system are that it “provides intentional and appropriate verbal communication”, and “allows people to coordinate their actions”. The importance of these two criteria for the IC3 system is agreed upon by almost everyone. Criteria such as that it “has to facilitate gestural communication”, “provide consequential communication of an individual’s embodiment”, and also “provide consequential communication of shared artefacts”, are assessed by evaluators as being relatively important. IC3 has performed relatively well with regard to these criteria.
Some criteria are identified to be unsatisfactorily applied. Evaluators have rated “protection”, and “finding collaborators and establishing contact” as important criteria, however they believe that these are poorly satisfied by the current IC3 system. The importance of criteria 6, which is the management of tightly and loosely-coupled collaboration, was unclear. Five evaluators have noted it as relatively important, whereas four has rated it as not very important. However, they agree that the targeting system does not satisfy this criterion well.

Question 9 and 10 (Table 3) ask evaluators how well they think the IC3 system facilitates eye contact and simulates the environment of sitting around a table. The results are consistent, as most evaluators believe these two criteria are satisfied very well. This indicates that IC3 system has performed well as an effective communication tool.

Questions 11 and 12 (Table 4) investigate whether video communication distracts participants’ focus on the content, and whether maximising the content window is essential. No evaluators marked these two criteria as very important. Most believe that maximising the content window is not a big issue, while the responses regarding whether video communication distracts focus are relatively spread out.

Five evaluators answered the open-ended question. They confirmed that the IC3 system does provide good interaction between members, however several problems were raised. A common issue identified by three evaluators is the ignorance of video when people are focusing on the collaborated content. Two evaluators point out that the control over a shared mouse pointer among participants is a vital issue. One evaluator identifies that sound effects need to be improved. Another two mention the overlap of images on the VNC client side when one person rolls the scroll wheel on the mouse.
Table 2: Results for Scenario Questions 1-8

<table>
<thead>
<tr>
<th></th>
<th>a*</th>
<th>b**</th>
<th>Very important</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Not important</td>
<td>Poor</td>
<td>Perfect</td>
</tr>
<tr>
<td>1(a)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1(b)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2(a)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2(b)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3(a)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3(b)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4(a)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4(b)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5(a)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5(b)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6(a)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6(b)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7(a)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7(b)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8(a)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8(b)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes to Table 1:
* Questions to assess the importance of the criteria
** Questions to assess how well the targeting system satisfies the criteria

Table 3: Results for Scenario Questions 9-10

<table>
<thead>
<tr>
<th></th>
<th>Poorly</th>
<th>Perfectly</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 4: Results for Scenario Question 11-12

<table>
<thead>
<tr>
<th></th>
<th>Not at all</th>
<th>Very</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

The overall evaluation for IC3 shows that it is a great tool for communication and collaboration. It achieves most of the goals from the initial design, such as video/audio communication, a shared web browser, gaze awareness support and the
simulation of sitting around a table. However, there is still a gap between the existing system and the expected system. Most evaluators agree that the “ability to provide protection” and “facilitating finding collaborators and establishing contact” are the two critical problems for the current system. As discussed in chapter 3, advanced techniques such as floor control, telepointers and overall system initiating design are the solutions for these problems. To perfect the system, these techniques should be further added to the IC3 system.

Some problems and evaluation limitations were observed during the evaluation process:

- Evaluators who are good at communicating may reflect the positive result
- Evaluators may or may not have had previous experience in using VCoIP and the shared view collaborative tool. Some of them even felt uncomfortable with the headset on.
- Most evaluators tried to compare the computer based communication system IC3 with real life communication. Unsatisfactory feedbacks such as that the sound is not 3D sound, were made.
- Testers knew each other well, so it was difficult to simulate geographical separation
- Problems with the first time use of a new system. Some evaluators complained that the GUI is too complex to understand.
- Some criteria from Baker et al’s (2001) groupware evaluation are difficult to understand or unclear. Some criteria are too general, which can be interpreted in different meanings when applying to the targeting system.
- Some of the usability issues depend very much on the purpose of the use of the system as well as users’ habits and personal preferences.
Chapter Six

Conclusions

This chapter provides the conclusions of the study drawn from the previous chapters. Limitations and several directions for future research are also identified.
6.1 Conclusions and Future Research Direction

This study examined the possibility of achieving the pre-designed IC3 (integrated communication and collaboration) environment by integrating existing open source projects. Many open source VoIP solutions have been reviewed according to the IC3 system requirements and it was found that the majority of them failed to meet the requirements. In terms of shared web browsers, the VNC system seems to be the only open source solution. It is regarded as a better solution because of its real-time information sharing capacity, application transparency and mature development. Finally, the open source software VIS and TightVNC were integrated to achieve the IC3 system and the effectiveness of this system was tested. The results aligned with the view that this IC3 system is an effective communication and collaboration tool. Although there are still limitations associated with the results, most evaluators agree that this system improves their eye contact and feeling of sitting around a table.

There are a few limitations of the IC3 system, which have been identified and discussed in the system implementation stage. These limitations are further confirmed by the system evaluation results. In summary, these limitations include telepointer support, floor control support and simple system initiation process support. Future research could investigate solutions to overcome these limitations and further test the system’s usability.
References


Appendix

Appendix 1: Scenario for IC3 Testing

There are 3 people in this scenario, person A, B and C. A wants to buy a car from trademe (www.trademe.co.nz) and would like his friends B and C to provide some suggestions. A has a budget of $5000. He could borrow another $1000 from his parents if he really likes the car. For other preferences of the car, B and C have to ask A. The possible topics that could be discussed:

- Make
- Model
- Color
- Year
- Mileage
- Other features

Note:
The IC3 system currently does not support multiple pointers, thus only one person can control the mouse at a time. Three of you need to negotiate who will take the control of the mouse pointer. Please do not resize the content window.

Procedure:
1) Assign the roles (A, B and C) to each of you.
2) Have a general chat (about anything you like) for at least 5 mins before you start browsing on the internet.
3) Carry on the scenario.
4) This scenario should take no more than 30 mins.
5) Complete the evaluation sheet.

Thanks for your participation.
1) Provide the means for intentional and appropriate verbal communication
   a) How would you rate the importance of this criterion for the targeting system?
      Not important very important
      □ □ □ □ □ □
   b) How does the targeting system satisfy this criterion?
      Poor OK Perfect
      □ □ □ □ □ □

2) Provide the means for intentional and appropriate gestural communication
   a) How would you rate the importance of this criterion for the targeting system?
      Not important very important
      □ □ □ □ □ □
   b) How does the targeting system satisfy this criterion?
      Poor OK Perfect
      □ □ □ □ □ □

3) Provide consequential communication of an individual’s embodiment
   a) How would you rate the importance of this criterion for the targeting system?
      Not important very important
      □ □ □ □ □ □
   b) How does the targeting system satisfy this criterion?
      Poor OK Perfect
      □ □ □ □ □ □

4) Provide consequential communication of shared artifacts
   a) How would you rate the importance of this criterion for the targeting system?
      Not important very important
      □ □ □ □ □ □
   b) How does the targeting system satisfy this criterion?
      Poor OK Perfect
      □ □ □ □ □ □

5) Provide protection
   a) How would you rate the importance of this criterion for the targeting system?
      Not important very important
      □ □ □ □ □ □
   b) How does the targeting system satisfy this criterion?
      Poor OK Perfect
      □ □ □ □ □ □

6) Management of tightly and loosely-coupled collaboration
   a) How would you rate the importance of this criterion for the targeting system?
      Not important very important
      □ □ □ □ □ □
b) **How does the targeting system satisfy this criterion?**

<table>
<thead>
<tr>
<th>Poor</th>
<th>OK</th>
<th>Perfect</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

7) **Allow people to coordinate their actions**

a) **How would you rate the importance of this criterion for the targeting system?**

<table>
<thead>
<tr>
<th>Not important</th>
<th>Very important</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

b) **How does the targeting system satisfy this criterion?**

<table>
<thead>
<tr>
<th>Poor</th>
<th>OK</th>
<th>Perfect</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

8) **Facilitate finding collaborators and establishing contact**

a) **How would you rate the importance of this criterion for the targeting system?**

<table>
<thead>
<tr>
<th>Not important</th>
<th>Very important</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

b) **How does the targeting system satisfy this criterion?**

<table>
<thead>
<tr>
<th>Poor</th>
<th>OK</th>
<th>Perfect</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

9) **How do you think the system facilitate eye contact?**

<table>
<thead>
<tr>
<th>Poorly</th>
<th>OK</th>
<th>Perfect</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

10) **How well do you think the system simulate the environment of sitting around a table?**

<table>
<thead>
<tr>
<th>Not at all</th>
<th>Very</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

11) **To what extend do you think video communication distracts your focus on the content?**

<table>
<thead>
<tr>
<th>Not at all</th>
<th>A lot</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

12) **To what extend do you think to be able to maximize the content window is important for you?**

<table>
<thead>
<tr>
<th>Not at all</th>
<th>Very</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

13) **Is there any other usability issue for this system you want to address (good or bad)? And how would you rate it? (1 poorly presented – 5 perfectly presented)**
Summarised Heuristic Evaluation for Groupware (Baker, Greenberg & Gutwin, 2001)

1) **Provide the means for intentional and appropriate verbal communication**
   The verbal conversation is an important way and is typically used to establish a common understanding of a task in a group environment. Direct verbal communications or other communication channels (video, text) together indicate how well the groupware support communication activity.

2) **Provide the means for intentional and appropriate gestural communication**
   Gestures and other visual actions are also often used to carry out intentional communication. Many studies show gestures play significant role in a communication and collaborative environment. Typical gesture supports in groupware include telepointers, avatars and video images. Telepointer is basically means make one’s pointer visible to all so that allows one to gesture or point to object in a workspace. Avatars are synthetic bodies representing the people who populate a 3D landscape.

3) **Provide consequential communication of an individual’s embodiment**
   A person’s body actions such as position, posture, and movement of head, arms and eyes can unintentionally give information to others. These visual cues provide conversational awareness that helps people maintain a sense of what is happening in a conversation. Groupware implementations such as telepointer, audio, video and special camera arrangement to facilitate eye contact and other more advanced system with sensors to detect person’s body movement.

4) **Provide consequential communication of shared artifacts**
   In a shared collaborative environment, consequential communication also involves information unintentionally given off by artifacts when they are manipulated by individuals. This information is the feedback to inform others who is manipulating the artifacts. Early groupware systems imposed “what you see is what I see” (WYSIWIS) view sharing where all participants saw the exact same actions as they occurred in the work space.

5) **Provide protection**
   In a groupware environment, concurrent access to objects can introduce the potential for conflicts. People can inadvertently interfere with work that others are doing now, or alter or destroy work that others have done. Thus protection techniques such as access control, concurrency control, undo, version control, turn-taking should be implemented in groupware where appropriate.

6) **Management of tightly and loosely-coupled collaboration**
   Coupling is the amount of work that one person can do before they require discussion, instruction, information or consultation from other members of a group. People should be able to move from their individual work (loose coupling) to work with others (tight coupling) and vise versa. This requires techniques to manage and allow a person to do its own tasks as well as maintain awareness of others.

7) **Allow people to coordinate their actions**
In order to maintain effective interaction and avoid conflicts in a shared workspace, coordination should be accomplished explicitly (through direct communication) or implicitly (mediate by the awareness of the shared material used in the workspace). Thus visual techniques presented in heuristics 1 to 5 will help to establish communication and awareness.

8) **Facilitate finding collaborators and establishing contact**

How people begin their collaborative session using a groupware sometimes can be a problem. A good groupware should facilitate easy finding collaborators and establishing contact.