Supporting Group Awareness in Synchronous Distributed Groupware: Framework, Tools and Evaluations

by

Minh Hong Tran

B.CS (MU)
M.IT (SUT)

A thesis submitted to
Faculty of Information and Communication Technologies
Swinburne University of Technology
for the degree of
DOCTOR OF PHILOSOPHY

June, 2006
Synchronous distributed groupware systems are computer-based applications that allow geographically dispersed people to work together at the same time. Developing such groupware systems is difficult and complex, and involves a variety of technical and social challenges. One of the challenges is to provide a user with a sense of being in a group and information about other users’ activities—supporting group awareness.

Supporting group awareness has been known as one of the most important factors in improving the usability of groupware systems. In face-to-face collaboration, people maintain group awareness naturally via various verbal and non-verbal cues. However, these rich and familiar cues are usually difficult to find over distance when a shared workspace is virtual and dispersed. As a result, group collaboration supported by synchronous distributed groupware is often unnatural, difficult and inefficient compared to the face-to-face counterpart.

The aim of this research is to improve support for group awareness in synchronous distributed groupware. This aim consists of two main sub-objectives: understanding the concept of group awareness and developing usable tools to support group awareness. In order to achieve these two objectives, a three-stage research process was carried out, involving operationalising the concept of group awareness, designing awareness tools, and evaluating the usefulness of resulting prototyped tools.

First, in order to operationalise the concept of group awareness, the research began with several empirical studies of distributed groupware, including Instant Messaging, collaborative authoring and multi-player computer games. Against the results of these user studies and existing theories of groups, the F@ (read as “fat”) framework of group awareness for synchronous distributed groupware has been developed. F@ contributes to the literature by identifying properties of four essential sub-types of group awareness, referred to as “awareness schools”, including conversational awareness, workspace awareness, contextual awareness and self-awareness. These four awareness schools cover principal knowledge that needs to be provided to maintain group awareness. In addition, F@ presents formal descriptions of fundamental aspects of these four
awareness schools using temporal logic. The principles addressed in F@ are important in extending groupware developers’ understanding of group awareness, and can be used as a guideline to design supporting mechanisms.

Second, using F@ as a guideline, four innovative awareness prototypes have been developed, including *Relaxed Instant Messenger* (RIM), *Conversation Dock* (ConDock), *Extended Radar View* (ERV) and *Modification Director* (MD). These prototypes were designed to support group awareness in Instant Messaging and collaborative authoring. For example, RIM integrates the threaded metaphor and the sequential metaphor to enhance awareness in a group conversation. ConDock applies a focus+context visualisation technique to improve users’ awareness of multiple, concurrent conversations. ERV maintains group awareness in collaborative authoring by showing simultaneously users’ working and viewing areas in a shared document. MD supports group awareness in collaborative authoring by helping users stay aware of changes that other users make to their text.

Third, several user-based studies were conducted to evaluate the four prototypes. In particular, laboratory-based empirical tests were used to assess the design of RIM, ERV and MD; and a field trial was used to evaluate ConDock. The evaluations confirm that these prototypes meet their design goals and enhance group awareness support significantly. The evaluations also yield several usability issues and provide valuable design feedback for improving user interfaces of the prototypes.

Finally, based on the principles of F@ and experience gained from developing the four awareness prototypes, the thesis discusses the potential use of F@ in designing awareness mechanisms for other synchronous distributed groupware systems.
The Author’s Publications

Patents

- *Methods and Systems for Managing Electronic Messaging*, a pending patent submitted by Smart Internet Technology Cooperative Research Centre (SITCRC).

Journal papers


Conference papers


Acknowledgements

There are so many people who I wish to thank for their support through my Ph.D. To me, their support has been enormous and has made my PhD such a wonderful experience.

I sincerely thank Professor Yun Yang, who has been my wonderful supervisor for the past years. Thanks Yun for his consistent supervision, for his continuous commitment to this research, and for his uplifting support for my Ph.D. research and even beyond. I deeply appreciate his considerable effort and invaluable comments in earlier drafts of this thesis. I also thank my associate supervisor, Dr. Gitesh K. Raikundalia, for his support and valuable guidance especially for his time in correcting the earlier draft of this thesis.

I would like to thank Smart Internet Technology CRC for their financial support throughout my Ph.D. Without their support, life would have been much harder otherwise. I would like to thank Faculty of Information and Communication Technologies for their assistance in various administrative issues.

I sincerely thank Professor Chengzheng Sun’s team for the CoWord system that has been so important for my research. I acknowledge the support of John Craick for his assistance in setting up the usability lab. I wish to extend my thanks to all people who participated in my experiments. I thank Neo Nguyen, Shane Grund and John Gheybi for their assistance in prototyping.

Leaving the best to last, I am grateful to my family, Dad, Mom, Han and Hien for their love and encouragement.

Thank you all, to these people I am in debt.
Declaration

This thesis contains no material which has been accepted for the award to the candidate of any other degree or diploma, except where due reference is made in the text of the thesis. To the best of the candidate’s knowledge, this thesis contains no material previously published or written by another person except where due reference is made in the text of the thesis.

Hong Minh Tran

June 2006
# Table of Contents

Abstract ..............................................................................................................................i
The Author’s Publications ...............................................................................................iii
Acknowledgements ...........................................................................................................v
Declaration .......................................................................................................................vi
Table of Contents ............................................................................................................vii
List of Figures ................................................................................................................xiv
List of Tables ................................................................................................................xvii

Chapter 1: Introduction .................................................................................................1
  1.1 Context of the research ....................................................................................2
  1.2 Research problem .........................................................................................3
  1.3 Aim of the research .......................................................................................3
  1.4 Structure of the thesis ....................................................................................4

PART I: THE EVOLUTION OF F@ ....................................................................................7

Chapter 2: Group Awareness and Cooperative Work ................................................8
  2.1 Introduction .....................................................................................................8
  2.2 Group awareness .............................................................................................9
  2.3 Group awareness and cooperative work .......................................................11
    2.3.1 Benefits of providing group awareness ..............................................11
      2.3.1.1 Support for communication ...............................................12
      2.3.1.2 Support for coordination ....................................................14
      2.3.1.3 Support for conventions .....................................................16
    2.3.2 Problems of providing group awareness ............................................16
      2.3.2.1 Privacy violation ................................................................17
      2.3.2.2 Disruption...........................................................................18
  2.4 Supporting mechanisms for group awareness ...............................................18
    2.4.1 Awareness mechanisms for communication ........................................19
      2.4.1.1 Text-based communication tools .......................................19
      2.4.1.2 Audio- and video-mediated systems ..................................20
      2.4.1.3 Virtual environments..........................................................21
    2.4.2 Awareness mechanisms for coordination ...........................................22
      2.4.2.1 Strict-WYSIWIS ................................................................22
      2.4.2.2 Relaxed-WYSIWIS............................................................23
      2.4.2.3 Sensor-enabled mechanisms ..............................................29
2.4.3 Awareness mechanisms for conventions ...........................................29
2.5 Frameworks and models of supporting group awareness ......................31
2.5.1 Spatial-based awareness models .......................................................32
  2.5.1.1 Rodden’s model of group awareness ........................................32
  2.5.1.2 The Aether model......................................................................33
2.5.2 Descriptive frameworks of awareness ..............................................34
  2.5.2.1 Gutwin and Greenberg’s framework of workspace awareness ........34
  2.5.2.2 Vertegaal et al’s framework of awareness ................................35
2.5.3 Notification-based models of awareness .........................................36
2.6 Group awareness matrix .......................................................................38
2.7 Summary ..............................................................................................39

Chapter 3: Empirical Studies of Synchronous Distributed Groupware Systems...41
3.1 Introduction ..........................................................................................41
3.2 Empirical study of Instant Messaging ..................................................42
  3.2.1 The method ....................................................................................42
    3.2.1.1 Online survey ........................................................................43
    3.2.1.2 Face-to-face interviews ............................................................43
  3.2.2 Findings ..........................................................................................44
    3.2.2.1 Awareness of presence ..........................................................44
    3.2.2.2 Awareness of turn-taking .......................................................45
    3.2.2.3 Awareness of conversational context ......................................46
    3.2.2.4 Awareness of emotions ..........................................................47
    3.2.2.5 Awareness of identities ..........................................................48
    3.2.2.6 Awareness of multiple concurrent conversations .................49
3.3 Empirical study of synchronous collaborative authoring .......................49
  3.3.1 The method ....................................................................................50
  3.3.2 Findings ..........................................................................................51
    3.3.2.1 Conversational awareness .......................................................52
    3.3.2.2 Workspace awareness ...........................................................52
    3.3.2.3 Contextual awareness ............................................................54
    3.3.2.4 Self-awareness .......................................................................54
3.4 Empirical study of multi-player computer games ....................................55
  3.4.1 The method ....................................................................................56
    3.4.1.1 Experimental design ...............................................................57
    3.4.1.2 Participants ...............................................................................59
    3.4.1.3 Experimental procedure .........................................................59
  3.4.2 Findings ..........................................................................................61
    3.4.2.1 Conversational awareness .......................................................61
    3.4.2.2 Workspace awareness ...........................................................64
PART II: THE F@ FRAMEWORK OF GROUP AWARENESS ........................................... 68

Chapter 4: The F@ Framework of Group Awareness .............................................. 69
4.1 Introduction ................................................................................................. 69
4.2 Analysis of group components .................................................................... 71
   4.2.1 Coordination theory ........................................................................ 71
   4.2.2 Model of virtual teams .................................................................... 72
   4.2.3 Group components for the study of group awareness .................... 72
4.3 Abstract level of F@ ................................................................................ 74
   4.3.1 Conversational awareness ............................................................... 75
   4.3.2 Workspace awareness ................................................................... 79
   4.3.3 Contextual awareness ................................................................... 81
   4.3.4 Self-awareness ............................................................................. 86
4.4 Concrete level of F@ ................................................................................ 87
   4.4.1 Conversational awareness ............................................................... 92
   4.4.2 Workspace awareness ................................................................... 95
   4.4.3 Contextual awareness ................................................................... 98
   4.4.4 Self-awareness ............................................................................. 99
4.5 Comparison between F@ and other frameworks ........................................ 99
4.6 Summary .................................................................................................... 101

PART III: APPLYING F@ TO THE DESIGN OF AWARENESS MECHANISMS .......... 104

Chapter 5: Designing Awareness Support for Instant Messaging ....................... 105
5.1 Introduction ............................................................................................... 105
5.2 Support for conversational awareness in IM ............................................ 108
   5.2.1 Awareness of presence ................................................................. 108
   5.2.2 Awareness of turn-taking .............................................................. 109
   5.2.3 Awareness of conversational context ........................................... 111
   5.2.4 Awareness of emotions ................................................................. 112
   5.2.5 Awareness of identities ................................................................. 113
   5.2.6 Awareness of multiple concurrent conversations ......................... 114
5.3 Existing design metaphors ........................................................................ 115
   5.3.1 Sequential text-based metaphor .................................................... 115
   5.3.2 Threaded text-based metaphor ....................................................... 116
   5.3.3 Avatar-based metaphor ................................................................. 117
5.4 Design of awareness mechanisms ............................................................... 119
   5.4.1 Design of RIM ............................................................................. 119
5.8.2 Reflection on the concrete level.......................................................155
5.9 Summary .............................................................................................158

Chapter 6: Designing Awareness Support for Synchronous Collaborative Writing ........................................................................................................160
6.1 Introduction.............................................................................................160
6.2 Support for group awareness in collaborative writing .........................161
  6.2.1 Support for conversational awareness .........................................161
  6.2.2 Support for workspace awareness ..............................................164
  6.2.3 Support for contextual awareness ..............................................169
  6.2.4 Support for self-awareness .......................................................170
6.3 Design of CoWord+ ...............................................................................171
  6.3.1 Design of ERV .............................................................................172
    6.3.1.1 Design rationale ...............................................................172
    6.3.1.2 User interfaces of ERV ....................................................174
  6.3.2 Design of MD .............................................................................178
    6.3.2.1 Design rationale ...............................................................178
    6.3.2.2 User interfaces of MD ......................................................182
6.4 Implementation of CoWord+ .................................................................185
6.5 Evaluation of CoWord+ .........................................................................189
  6.5.1 Laboratory-based evaluation of CoWord+ ....................................189
    6.5.1.1 Goals of the evaluation ....................................................189
    6.5.1.2 Participants .......................................................................190
    6.5.1.3 Tasks .................................................................................190
    6.5.1.4 Experimental procedure ................................................190
  6.5.2 Results of CoWord+ evaluation ....................................................192
    6.5.2.1 Evaluation of ERV ...........................................................192
    6.5.2.2 Evaluation of MD .............................................................194
    6.5.2.3 Usefulness of a dual monitor setting ................................197
  6.5.3 Discussion of CoWord+ evaluation .................................................198
    6.5.3.1 Improving ERV .................................................................198
    6.5.3.2 Improving MD .................................................................199
6.6 Reflection on the use of F@ ....................................................................200
  6.6.1 Reflection on the abstract level ....................................................200
  6.6.2 Reflection on the concrete level ....................................................201
6.7 Summary .............................................................................................204

Chapter 7: Discussion ...................................................................................206
7.1 Introduction............................................................................................206
7.2 Discussion of awareness support in other synchronous groupware systems...
    ........................................................................................................207
7.2.1 Collaborative drawing ................................................................. 207
7.2.2 Electronic meetings ................................................................. 209
7.2.3 Collaborative programming ....................................................... 210
7.2.4 Multi-player computer games .................................................... 212
7.3 Discussion of F@ ........................................................................... 213
7.3.1 Analytical reflection ................................................................. 214
7.3.2 Critique .................................................................................... 216
7.4 Summary ....................................................................................... 217

Chapter 8: Conclusions and Future Work ................................................. 219
8.1 Summary of the thesis ................................................................. 219
8.2 Revisiting the research aim .......................................................... 222
8.3 Key contributions of the research .................................................. 223
8.4 Future work .................................................................................. 224

Bibliography ......................................................................................... 227
Glossary ............................................................................................... 253
Appendix A: Materials of the Empirical Studies ....................................... 256
A.1 Empirical study of Instant Messaging .............................................. 256
  A.1.1 Demographic questions ............................................................ 256
  A.1.2 7-point Likert scale questionnaire ........................................... 257
  A.1.3 Open-ended questions .............................................................. 258
A.2 Empirical study of collaborative authoring ....................................... 259
  A.2.1 5-point Likert scale questionnaire ............................................ 259
  A.2.2 Open-ended questionnaire ....................................................... 259
A.3 Empirical study of multi-player computer games ................................ 261
  A.3.1 Demographic questions ............................................................ 261
  A.3.2 Open-ended questions .............................................................. 261

Appendix B: Materials of the Evaluations ................................................. 263
B.1 Evaluation of RIM ........................................................................... 263
  B.1.1 7 Likert scale questionnaire ..................................................... 263
  B.1.2 Open-ended questions .............................................................. 265
B.2 Evaluation of ConDock ................................................................. 266
  B.2.1 7 Likert scale questionnaire ..................................................... 266
  B.2.2 Open-ended questions .............................................................. 266
B.3 Evaluation of CoWord+ ................................................................. 267
  B.3.1 7 Likert scale questionnaire ..................................................... 267
  B.3.2 Open-ended questions .............................................................. 268

Appendix C: Usefulness of Dual Monitors ................................................. 269
C.1 Dual monitors affect game performance ......................................... 269
  C.1.1 Dual monitors are useful in Counter-Strike ............................... 269
C.1.2 Dual monitors are less useful in WarCraft III .................................270
C.2 Participants enjoyed dual monitor gaming ...........................................270
C.3 Participants’ use of dual monitors .......................................................270
  C.3.1 Many quick glances; few quick glances followed by a long glance 270
  C.3.2 Physical layout of dual monitors ....................................................271
  C.3.3 How participants adapted to dual monitors .................................272
List of Figures

Figure 1.1: Thesis organisation ........................................................................................................4
Figure 2.1: Communication among team members, adapted from [156] ................................. 12
Figure 2.2: Babble [77] provides a user list and social proxy to support awareness .... 19
Figure 2.3: Examples of mediaspaces. (A) Snapshot of CoMedi [54]; (B) GAZE Groupware System [243] supports gaze direction in video-mediated communication ............................................. 20
Figure 2.4: Avatars communicate in CVEs [196] ........................................................................ 21
Figure 2.5: Augmented reality interfaces support remote interaction via live video avatars [23] ........................................................................................................ 22
Figure 2.6: Telepointers (adapted from [95]) ............................................................................ 23
Figure 2.7: GroupDesign [13] supporting viewports ................................................................. 24
Figure 2.8: Two variations of multi-user scrollbars. (A) Remote scrollbars align in the same region [10]; (B) Remote scrollbars located in different regions [105]. ...................................................................... 25
Figure 2.9: Radar view [110] ........................................................................................................ 26
Figure 2.10: Awareness mechanisms based on distortion-oriented displays [94]. (A) Fisheye views; (B) the Offset Lens; (C) the Head-up Lens .......................................................... 28
Figure 2.11: Describing awareness using the notions of nimbus and focus [193] .... 31
Figure 2.12: Models to support users’ awareness of context [99] .............................................. 37
Figure 2.13: Group awareness matrix .......................................................................................... 38
Figure 3.1: A snapshot of REDUCE (two users used different text colours) .................. 50
Figure 3.2: Dual monitor setting ................................................................................................. 58
Figure 3.3: Floor plan of the laboratory setting.
Figure 4.1: Operational structure of F@ .................................................................................. 70
Figure 4.2: Components of coordination theory [144] ............................................................ 71
Figure 4.3: Awareness schools and group components.
Figure 4.4: Relations between goals, tasks and results, adapted from [140] ................. 84
Figure 4.5: Two types of temporal logic .................................................................................... 88
Figure 5.1: Research process .......................................................................................................... 107
Figure 5.2: Support for presence awareness (snapshot of Trillian) .................................. 109
Figure 5.3: “Who is typing” cue supports turn-taking (snapshot of Trillian) ................. 110
Figure 5.4: Quasi-shared windows: Minh and exitobay’s conversation windows are different (snapshot of Gaim) ................................................................. 111
Figure 5.5: Emoticons support emotional awareness (snapshot of Trillian). ..........112
Figure 5.6: Customised avatars (snapshot of Yahoo Messenger)........................113
Figure 5.7: Tabbed conversations (snapshot of Gaim)........................................114
Figure 5.8: Sequential text-based metaphor (snapshot of Gaim)........................115
Figure 5.9: Threaded-based metaphor (snapshot of Threaded Chat [207])..........116
Figure 5.10: Avatar-based metaphor (a snapshot of The Palace [209]). .........118
Figure 5.11: Relaxed Instant Messenger (RIM). (A): Tree Canvas organises messages in a tree layout, and shows a textual “Who is typing” cue; (B): Message Canvas displays messages in a linear order; (C): Chat Area includes a text field which implements URP; (D): Buddy List displays multiple visual cues of “Who is typing”, and shows how long users have been logged onto the conversation; (E): Topic List shows a list of top-level threads of a conversation as well as the number of replies in each thread. .................121
Figure 5.12: Conversation Dock (ConDock). .......................................................124
Figure 5.13: Class diagram of RIM. ....................................................................128
Figure 5.14: Typical interactions between a RIM client and a RIM server.........129
Figure 5.15: System architecture of ConDock.......................................................130
Figure 5.16: Adaptation of the MSN Messenger APIs using COM technology. ......131
Figure 5.17: Mean of the numbers of times that participants used Tree Canvas and Chat Area to compose messages.................................................................137
Figure 5.18: Error rate (%) of URP. ......................................................................138
Figure 5.19: Structural order of messages. ............................................................140
Figure 5.20: Tree Canvas with fish-eye views (mock up).......................................142
Figure 5.21: A clock-like visual cue (mock-up). ....................................................143
Figure 5.22: Weekly measure of participants’ responses to usefulness of ConDock in managing multiple conversations.........................................................146
Figure 5.23: Avatars and user names enhance conversation identification of ConDock (mock up). .................................................................148
Figure 5.24: Track View supports awareness of viewing- and listening-in-progress...150
Figure 5.25: Buddy List ......................................................................................151
Figure 5.26: Class diagram showing the framework of multiple identity support. .....152
Figure 5.27: High-fidelity prototype supporting multiple identities....................153
Figure 5.28: RIM supports presence awareness of past and current conversants.....156
Figure 5.29: RIM supports CA1 and CA3............................................................157
Figure 6.1: Text-based chat tool. (A) chat window in NetEdit [264]; (B) scratchpad in Calliope [158].................................................................163
Figure 6.2: Snapshot of Flexible JAMM [16] (two viewports at the right)..........166
Figure 6.3: Symbolic indicator of indirect manipulation (mock-up)....................168
Figure 6.4: Presentation of changes in Flexible Diff [169]. ............................................168
Figure 6.5: Calliope [158] displays different colours for different users. .......................169
Figure 6.6: Support for contextual awareness in collaborative authoring. Snapshots of (A) GROVE [73]; (B) Quilt [135]..................................................................170
Figure 6.7: Snapshot of CoWord. (A) a shared Word document with telepointers; (B) the radar view. .......................................................................................................171
Figure 6.8: Extended Radar View (ERV). .........................................................................175
Figure 6.9: View extents and view sizes of these views in ERV....................................178
Figure 6.10: Initial mock-up of MD [227].......................................................................180
Figure 6.11: Change tracking feature of Word. (A): a Word document. (B): a reviewing pane. ..............................................................................................................181
Figure 6.12: Modification Director (MD)........................................................................183
Figure 6.13: System architecture of ERV and MD............................................................186
Figure 6.14: Four-layer architecture of ERV and MD.....................................................187
Figure 6.15: Adaptation of the Word APIs using COM technology. ...............................188
Figure 6.16: The dual monitor setting. (A) primary monitor shows a shared Word document; (B) secondary monitor shows ERV and MD awareness mechanisms. .........................................................................................191
Figure 6.17: Presence awareness of past, current, future group members (mock up). 202
Figure 6.18: Support for awareness of artefacts (mock up).............................................203
Figure C.1: The layout of primary and secondary monitors.............................................272
## List of Tables

| Table 2.1: | Elements of workspace awareness [107]. | 34 |
| Table 2.2: | Presenting elements of workspace awareness and conversational awareness using attentive states [246]. | 36 |
| Table 3.1: | Experimental conditions. | 60 |
| Table 3.2: | Examples of verbal expressions when dual monitors were or were not available. | 63 |
| Table 4.1: | Components and principles of a virtual team [140]. | 72 |
| Table 4.2: | Summary of group components. | 74 |
| Table 4.3: | Associated process and components of conversational awareness. | 76 |
| Table 4.4: | Dimensions of conversational awareness. | 78 |
| Table 4.5: | Associated process and components of workspace awareness. | 80 |
| Table 4.6: | Dimensions of workspace awareness, adapted from [107] and added with "Why" dimension. | 81 |
| Table 4.7: | Associated process and components of contextual awareness. | 85 |
| Table 4.8: | Dimensions of activity contextual awareness. | 85 |
| Table 4.9: | Dimensions of self-awareness. | 87 |
| Table 4.10: | Similarities between F@ and other existing frameworks of group awareness. | 99 |
| Table 4.11: | Differences between F@ and other existing frameworks of group awareness. | 101 |
| Table 5.1: | Experimental procedure of each session. | 134 |
| Table 5.2: | Structure of a message tree in RIM. | 135 |
| Table 5.3: | Message Canvas supports a focal point. | 136 |
| Table 5.4: | ANOVA test of the number of messages posted via Chat Area and Tree Canvas. | 137 |
| Table 5.5: | Participants’ learnability. | 144 |
| Table 5.6: | Repeated measures ANOVA: Mauchly’s Test of Sphericity. | 146 |
| Table 5.7: | F@ and mechanisms for conversational awareness. | 155 |
| Table 6.1: | Word supports modification awareness. | 182 |
| Table 6.2: | MD supports modification awareness. | 185 |
| Table 6.3: | Three test conditions of the CoWord+ evaluation. | 191 |
| Table 6.4: | Participants' responses to the usefulness of ERV. Scores range from 1 = "strongly disagree" to 7 = "strongly agree". | 192 |
Table 6.5: Participants' responses to the usefulness of MD. Scores range from 1 = “strongly disagree” to 7 = “strongly agree”. ............................................195

Table 6.6: Participants' responses to the dual monitor setting. Scores range from 1 = “strongly disagree” to 7 = “strongly agree”. ............................................197

Table 6.7: A comparative analysis between dimensions of workspace awareness and awareness support of ERV and MD. .........................................................201

Table C.1: Relation between a participant’s primary hand and their preferred side for the second monitor. .................................................................272
Chapter 1

Introduction

Synchronous distributed groupware systems are computer-based applications that allow members of a distributed group to interact with one another at the same time [73]. Developing such computer-based systems to support cooperative activity is highly necessary. These days, there is a growing demand for distributed group collaboration in organisations. People are expected to collaborate in a group even if they are situated at different locations [8, 9].

In order to support a group in performing cooperative activities effectively and efficiently, groupware systems need to provide group members with information about other members’ activities. It is a fundamental requirement for collaborating individuals to know with whom they cooperate, which tasks other people are doing, what tasks have been completed, and so on. This type of information is referred to as awareness or group awareness [67, 107, 193] (henceforth, this thesis uses these two terms interchangeably).

In face-to-face situations where people gather at the same place, awareness information is often perceived implicitly. Humans possess natural abilities to perceive and process information at the periphery of their attention. Thus, it is naturally easy for people to maintain a sense of awareness about who else is present in a group, what other people are doing, and so on. However, in distributed situations where people are located at different geographical sites, maintaining this sense of awareness becomes much more difficult due to various reasons, including limited capabilities of input and output devices, restricted views, and weak communication [36, 100, 107, 218]. As a result, group collaboration performed in a distributed setting is often unnatural and awkward as opposed to the face-to-face counterpart.

Research on group awareness has drawn rising interest from researchers of the fields of Computer-Supported Cooperative Work (CSCW) and Human-Computer Interaction.
(HCI). From early research, such as [102, 210], to more recent research, such as [99, 109, 200, 261], it has been emphasised that supporting group awareness helps improve the usability of groupware systems, and thereby enhances group collaboration.

Previous research on group awareness has focused on investigating the concept of group awareness and designing usable mechanical techniques to support group awareness. This thesis responds to these two research activities by:

1. developing a novel framework of group awareness to gain a useful and relevant understanding of group awareness, and
2. applying the framework to the design of innovative awareness mechanisms.

The remainder of this chapter presents the context of this research, states the problem and aim of the research, and outlines the structure of this thesis.

1.1 Context of the research

This research fits into the fields of CSCW and HCI. These research fields focus on understanding how members of a group collaborate with one another and designing technical support for group collaboration [8, 9, 97]. One research direction of CSCW and HCI targets the development of computer-based systems (that is, groupware systems) that are used to support cooperative activity of a group. This research focuses on one type of groupware systems, namely synchronous distributed groupware systems.

More specifically, this research examines the concept of group awareness and the support for group awareness in synchronous distributed groupware systems.

Synchronous distributed groupware systems allow members of a collaborating group to interact at the same time over distance [73]. Popular examples of these groupware systems include, but are not limited to, chat applications [124], audio- and video-mediated systems [68], collaborative editors [214], shared drawing tools [57], electronic meeting systems [261] and multi-player online computer games [215]. Synchronous distributed groupware systems are developed to support a wide range of cooperative activities, involving groups of different sizes, formal meetings, informal communication, tightly-coupled coordination, loosely-coupled coordination, and so on. Given this diversity of synchronous distributed groupware systems, the scope of this research is constrained to the investigation of awareness in groupware systems that
support collaboration of small groups. In this research, “small groups” refers to a group of usually about between two to five people [107].

1.2 Research problem
The main problem addressed by this research is the insufficient support for group awareness in prevalent synchronous distributed groupware systems. Because of limited support for group awareness, collaboration via synchronous distributed groupware systems is often unnatural, less effective and inefficient in comparison to a face-to-face situation.

Group awareness is an essential element of collaboration [107, 193, 200]. Previous studies of group awareness have shown that providing better support for group awareness increases the usability of groupware, and consequently delivers more natural and higher quality collaboration. Group awareness helps members of a group stay aware of the presence and activities of other members. Group awareness aids collaboration by facilitating communication, coordination, coupling and conventions [107, 146]. In a face-to-face condition, people are able to establish and maintain group awareness naturally via a variety of rich visual and verbal cues. Unfortunately, such cues are difficult to support in a distributed condition. Therefore, supporting group awareness for distributed groupware systems is challenging.

Researchers of synchronous distributed groupware systems have been devoted to investigating the concept of group awareness and designing usable mechanisms to support group awareness. Despite the significant progress that has been made and that many mechanisms have been developed, support for group awareness is still insufficient. Insufficient support for group awareness is mainly due to the lack of understanding of group awareness and the lack of effective awareness supporting mechanisms [200].

1.3 Aim of the research
The aim of this research is to improve support for group awareness in synchronous distributed groupware systems. This aim is divided into two objectives:

(1) Developing a framework of group awareness to gain a useful and relevant understanding of group awareness: This research works towards constructing a
framework of group awareness. The framework is able to describe different aspects of group awareness and relationships between these aspects. In particular, the framework aims to identify sub-types of group awareness and determine properties of each of these sub-types. The framework is constructed based on existing theories of groups and coordination, previous research on group awareness, and the findings of empirical studies of distributed collaboration.

(2) **Designing and implementing usable mechanisms to support group awareness:** The framework is applicable in developing mechanisms to support group awareness. This thesis shows how the framework can be applied in building awareness mechanisms. This objective is fulfilled when the thesis illustrates case studies that involve developing awareness mechanisms for Instant Messaging and collaborative authoring.

### 1.4 Structure of the thesis

The remainder of the thesis is organised into three parts, as illustrated in Figure 1.1, which correspond to research processes.

**Figure 1.1: Thesis organisation.**

Part I—*The Evolution of F@*—lays a foundation for the F@ framework of group awareness. Part I comprises Chapter 2 and Chapter 3. The aim of Chapter 2 is to present a background on group awareness, whilst Chapter 3 reports empirical studies that were carried out to investigate how group awareness is maintained in synchronous distributed groupware systems.
More specifically, Chapter 2 reviews the concept of group awareness by analysing various definitions of group awareness that researchers of CSCW have formulated to describe the concept. The analysis shows the complexity of group awareness and indicates a need for further investigation so that a better understanding of the concept can be gained and more effective awareness mechanisms can be developed. In addition, Chapter 2 examines group awareness and its relationships with cooperative activity by analysing benefits and problems of supporting awareness in groupware systems. Chapter 2 also presents an overview of existing mechanisms that are used to support awareness. The reviewed mechanisms include text-based chat tools, graphical widgets, audio- and video-mediated tools, and virtual environments. Furthermore, Chapter 2 analyses existing models and frameworks that have been developed to conceptualise the notion of group awareness. The existing models are valuable in setting up a fundamental knowledge of group awareness.

Chapter 3 reports the results of three empirical studies, including studies of awareness in Instant Messaging, synchronous collaborative authoring and multi-player computer games. The purpose of these studies is to investigate the importance of various aspects of awareness in supporting group communication and task coordination. The results of these studies contribute to the formation of the F@ framework reported in Part II.

Part II—The F@ Framework of Group Awareness—includes Chapter 4, which presents the F@ framework. The aim of F@ is to provide groupware designers with a better understanding of group awareness and useful guidelines of designing awareness mechanisms. F@ includes two parts: the abstract level and the concrete level. On the one hand, the abstract level adopts an informal method to provide in-depth descriptions of different aspects of awareness. On the other hand, the concrete level uses temporal logic to present formal descriptions of fundamental time-related aspects of group awareness. Chapter 4 is structured into three sections. The first section discusses coordination theory and the model of virtual teams that are used to form the F@ framework. The second section presents the F@ framework by describing the abstract level and the concrete level. The third section of Chapter 4 shows an analytical comparison between F@ and existing models of group awareness.

Part III—Applying F@ to the Design of Awareness Mechanisms—aims at illustrating the use of F@ in designing awareness mechanisms. Part III includes Chapters 5, 6 and
7. In particular, Chapter 5 and Chapter 6 present case studies of developing mechanisms to support awareness in Instant Messaging and collaborative authoring, respectively. These case studies describe the design, implementation and evaluation of prototyped awareness mechanisms. The design of resulting mechanisms presented in Chapters 5 and 6 has evolved in part from the F@ framework described in Part II. Chapter 7 presents an interpretive analysis of how F@ can be used to design mechanisms to support awareness in other groupware systems, including systems for collaborative drawing, electronic meetings, collaborative programming and multi-player computer games. Chapter 7 also presents a discussion on the use of F@, based on the development of mechanisms in Chapter 5 and Chapter 6, and the interpretive analysis of F@.

Finally, Chapter 8 concludes the thesis by responding to the aim of the research and summarising original findings and contributions of this research. Chapter 8 then presents possible directions for future research.
PART I:
THE EVOLUTION OF F@
Chapter 2

Group Awareness and Cooperative Work

2.1 Introduction

In today’s era, groupware systems have become an important element in supporting cooperative activity performed by a group [103]. Many groupware systems, such as electronic mail, discussion boards, desktop conferencing, file sharing and group decision support systems, have been used widely to support a range of group collaboration. For example, groupware systems are able to support social interaction, educational learning and teaching, and commercial collaboration. Groupware systems allow people to perform collaborative tasks when they meet in the same place (co-located) or dispersed over distance (distributed), and when they contribute to the task at different times (asynchronous) or gather together at the same time (synchronous) [73].

For a long time, face-to-face interaction has been realised as an ultimate and ideal environment for a group of people to collaborate. Unfortunately, for various reasons, such as differences in locations of workspace, time, cost and tight deadlines, a group cannot always afford to gather together in the same room, at the same time to perform their cooperative tasks. Groupware systems have been used as an alternative and complement to face-to-face interaction. In order to provide effective and efficient collaboration of a distributed group, it is important for groupware systems to support both users’ individual task—taskwork—and the group’s cooperative tasks—teamwork [104, 107]. Whilst supporting taskwork of an individual has been investigated for a long time in the design of single-user applications, supporting teamwork is relatively new.

The first conference which brought researchers from different disciplines together to discuss designing groupware to support teamwork was held in 1986: the first CSCW conference.

Previous research has emphasised that group awareness is highly important to the process and outcome of group collaboration. In particular, group awareness aids communication between members of a group, supports coordination and facilitates
conventions [99, 102, 109, 200, 210, 261]. It is fundamental for people to know with whom they cooperate, other peoples’ goals, their activities, how they progress with their tasks, and so on. Unfortunately, due to the nature of distributed locations, it is difficult to provide technical support for group awareness [36, 100, 107, 218]. This chapter examines the concept of group awareness, analyses the benefits and issues of providing awareness in groupware systems, and reviews related research on group awareness.

The remainder of this chapter is structured as follows. The next section examines briefly some of the existing definitions of group awareness and describes fundamental characteristics of awareness. Section 2.3 analyses the role of group awareness in facilitating cooperative activity and examines potential issues of providing group awareness. Section 2.4 reviews existing mechanisms used to support group awareness in synchronous distributed groupware systems. Section 2.5 examines frameworks and models that are developed to conceptualise the concepts and implementations of group awareness. Section 2.6 presents the author’s matrix of group awareness, based on discussion presented in the previous sections.

2.2 Group awareness

Before considering benefits and problems of group awareness, it is important to ask the question “What is group awareness?”. It is known for a long time that human interaction is influenced by the degree of how much people know each other, and collaboration is affected by shared information [91, 218]. In broad terms, group awareness is generally known as the reciprocal knowledge that people have about each other’s activity and the state of an environment [3]. Endsley [76] regards awareness as “knowing what is going on” (p. 36).

How have CSCW researchers defined group awareness? Interestingly enough, various definitions of group awareness have been formulated, amongst which some may differ subtly from others. CSCW researchers often use adjectives to indicate specifically what they mean by group awareness in a particular context [200]. For instance, “social awareness” [223], “collaboration awareness” [133], “peripheral awareness” [26], “workspace awareness” [107] and “mutual awareness” [18], and so on. The following analyses some of these definitions to showing diverse opinions about group awareness.
One early definition of group awareness, and probably one of the best-known definitions, was presented by Dourish and Bellotti [67]. They define group awareness as “an understanding of the activities of others, which provides a context for your own activity” (p. 107). The researchers argue that a context of activity involves both artefacts that people use to carry out their cooperative tasks, and the way in which artefacts are used. Supporting group awareness is particularly important as it ensures that an individual’s activity and contributions are relevant to the group’s activity as a whole.

Some other representative definitions of group awareness include:

- Gutwin and Greenberg [107] defined workspace awareness as “the up-to-the-moment understanding of another person’s interaction with the shared workspace”;

- Beaudouin-Lafon and Karsenty [13] defined group awareness as “each user should be aware of what others are doing, to facilitate coordination but not suffered from constraints related to group”; and

- Tollmar et al. [223] defined social awareness as an understanding of the “social situation of the members, i.e., awareness about what they are doing, if they are talking to someone, if they can be disturbed”.

Given that a variety of definitions of group awareness has been presented, this research does not endeavour to contribute a new definition to an already-crowded pool of definitions. This thesis uses the term group awareness to refer to a user’s state of “being aware of”, and an “understanding” and “knowledge” of the presence and activities of other users and their own presence and activities in a shared workspace. Information about awareness of people’s presence and activity spreads across the past, current and future temporal spectrum.

Prior to the investigation of support for group awareness in groupware systems, considerable effort has been devoted to studying the concept of awareness between an actor and its environment (referred to as situation awareness), such as aviation [142] and underground traffic control [116]. Previous research has identified four major characteristics of situation awareness [3, 75, 107]:
(1) Awareness is people’s knowledge of the state of a surrounding environment involving both temporal and spatial aspects.

(2) Awareness needs to be established and maintained in order to keep updated with the change of the environment.

(3) Awareness is maintained through perceptual information gathered from the environment and people’s interaction with the environment.

(4) Awareness is a derived goal of people’s activity. That is, in order to perform their main task, users need to maintain awareness.

This thesis considers group awareness as a specific form of situation awareness where an environment is comprised of both the distributed users’ physical workspace and the shared electronic workspace. In addition, people do not only interact with the environment but also with other people. Therefore, these four characteristics also apply to group awareness.

2.3 Group awareness and cooperative work

“How can group awareness support cooperative work?”. To answer this question, this section examines the benefits of group awareness and highlights concerns with providing group awareness.

2.3.1 Benefits of providing group awareness

The growing interest in awareness-related topics results from the fact that group awareness has been increasingly identified as a crucial part of successful collaboration. Perceiving and understanding the activities and intentions of other members of the collaborating ensemble is a basic requirement for human interaction. People require sufficient knowledge of the presence, activities and intentions of others in order to perform cooperative activity effectively and efficiently. Group awareness plays an essential and integral role in cooperative work. Gaver (1991) stated that “awareness is necessary for all collaborative work, but the degree to which its focus is shared varies” (p. 295). Less focused collaboration also requires less awareness, but some awareness support is required in all cases. The following sub-sections examine how group awareness can reduce people’s effort and increase efficiency for activities of collaboration by facilitating communication, supporting coordination and assisting conventions.
2.3.1.1 Support for communication

One of the fundamental functions of groupware systems is to support communication between people [11, 73, 157]. Communication supported by groupware systems can be classified as *asynchronous* or *synchronous* [239]. Asynchronous communication occurs when people do not communicate at the same time. Common examples of groupware systems supporting asynchronous communication include email and bulletin boards. Synchronous communication, on the other hand, allows people to communicate at the same time. Examples of groupware systems used for synchronous communication include systems for desktop conferencing. This research merely focuses on examining the support of group awareness for synchronous communication.

Synchronous groupware systems support both *direct* communication and *indirect* communication [62], as illustrated in Figure 2.1. Direct communication involves a direct transmission of messages between people, such as messages in the form of verbal speech and written messages. Indirect communication, on the other hand, involves people’s manipulation of shared artefacts (for example, moving or modifying artefacts). Supporting group awareness can enhance direct communication and indirect communication.

![Figure 2.1: Communication among team members, adapted from [156].](image)

Group awareness provides helpful cues that assist in keeping direct communication going smoothly. In conversation, people’s behaviours are often adjusted accordingly to cues picked up from their conversational partners [44]. Tang [217] stated that “ideas are often enacted gesturally in order to express them effectively to others, especially if they involve in a dynamic sequence of actions” (p. 76). For example, eye contact is important to determine whether or not others pay attention. People often provide gestures, such as head nods to indicate that they have understood an utterance. Additionally, gestures are often used by people to demonstrate actions. For instance,
speakers use fingers to point to the artefact about which they are talking. These forms of immediate awareness serve as communicational catalysts that drive direct communication naturally.

Providing group awareness also helps utilise the potential of indirect communication. In order to understand indirect communication, it is important that remote users’ actions are visible and perceived. It is difficult to interpret and comprehend invisible actions. For example, if user A is not aware of what user B is doing in the workspace, it is difficult for user A to comprehend to which artefact user B is pointing. Furthermore, the sender and the receiver need to have a common context in which visible actions occur, because the meaning of an action can be ambiguous without certain information. Awareness information regarding the sender, actions, invocation, time, motivation, and location need to be gathered, processed and made available through the distributed system.

In addition to assisting direct communication and indirect communication, group awareness contributes to reducing the complexity and length of dialogue. A person does not need to describe his/her activities in full detail when other people are aware of the person’s actions [31, 107]. For example, if user A knows that user B is modifying a text of a shared document, user B can ask a simple question like “Do you agree with that?” instead of “Do you agree with what I have changed in section 3, paragraph 2, line 39”, which is a much more complicated utterance.

Furthermore, supporting group awareness encourages informal communication among members of a collaborating group [124]. Informal communication refers to opportunistic, spontaneous and unplanned interaction between group members. Informal communication builds and relies upon people’s awareness of a common context, such as what actions others are carrying out, what artefacts others are manipulating, and so on. Providing group awareness is crucial in keeping people up to date with important events and activities in the shared workspace. Informal communication often occurs when people know that their partners are not too busy and can be interrupted without interfering too much with their ongoing work.
2.3.1.2 Support for coordination

Coordination is an important aspect of carrying out collaborative activity [73]. Coordination is broadly defined as “the act of working together harmoniously” [143 p. 358]. Coordination entails the combination of independent actions in an appropriate manner (for instance, at the right time, in the right order and satisfying task constraints) towards the accomplishment of a common goal.

Generally speaking, groupware systems can support coordination by utilising explicit communication and shared artefacts in the workspace [192]. Both of these approaches to supporting coordination are fundamental aspects of group awareness. The discussion above in Section 2.3.1.1 shows how group awareness can benefit direct communication, indirect communication and informal communication. Group awareness allows people to perceive and understand the temporal and spatial boundaries of others people’s actions, thereby aiding coordination by fitting the next action into the stream of actions appropriately and harmoniously.

Additionally, group awareness is useful in making coordination more flexible. Providing group awareness becomes important, particularly when people are interacting with the same artefact. Some CSCW researchers have argued that effective awareness mechanisms can replace strict concurrency and rigid access control mechanisms. For instance, if members of a group are able to see who is accessing shared artefacts and other peoples’ actions, conflicts in manipulating shared artefacts could be avoided [107]. More specifically, previous research has examined the support of group process for coordination in various domains such as collaborative writing [10], virtual environment [80], software development [109], electronic meetings [261], emergency services [181], air traffic control [142], and underground traffic control [116]. These studies show that harmonious coordination of collaborative activity is mediated by people’s knowledge of others’ activities and shared artefacts. Aytes’ empirical study of a physical whiteboard and an electronic whiteboard [128] showed that groups using the physical whiteboard performed better than those using the electronic whiteboard. The study attributed this finding to the lack of awareness support in the groups using the electronic whiteboard.

Research on group collaboration shows that members of a group often shift back and forth between individual tasks and cooperative tasks [67, 218]. Group awareness plays
an important role in facilitating smooth transitions between these two modes of work, known as coupling [105]. Coupling refers to the degree that members of a group collaborate with one another [197]. Coupling often involves communication, discussion, assistance and action. Under various circumstances, coupling between group members can be loose or tight. For example, the rIBIS hypertext editor [188] supports loose coupling by allowing users to scroll, move a mouse and edit a shared document independently whilst still being able to see other users’ actions. In a tightly-coupled mode, views on a shared document of users’ of rIBIS are locked tightly together: one user scrolls his/her local view resulting remote users’ views scrolling accordingly. Group awareness helps people recognise what level of coupling is appropriate and move smoothly between loose coupling and tight coupling. Group awareness conveys information about what other people have done, what they are currently doing and what they are going to do. Thus, awareness information helps people know whether or not their partners are available and can be interrupted without interfering too much with their work.

Furthermore, the persistent record of awareness information can contribute to building and maintaining “group memory” [182], or even “organisational memory” [2, 27] on a larger scale. Posner and Baecker’s study of collaborative writing [182] shows that group memory is useful in assisting coordination of co-authoring tasks.

One aspect of coordination is assistance that also benefits from group awareness [107]. Although assistance is usually explicitly shown in collaboration, it is a common behaviour in group collaboration. Assistance is often initiated by people’s request for help. In such cases, a person providing assistance would often stop their currently-engaged tasks for a period of time to help out their group members. In order to make assistance efficient and less disruptive, both a person seeking help and people offering assistance need to be aware of each other’s goals, tasks and the progress of their work. Group awareness is useful for assistance because it helps people understand a type of assistance that another person requires, and therefore responds to the request appropriately. In addition, it is observed that when group awareness is sufficiently supported, members of a group can offer assistance to one another without a prior request. For example, in Gutwin’s [105] observational study of joint jigsaw puzzles,
participants sometimes found and placed puzzle pieces for others without request from the others.

### 2.3.1.3 Support for conventions

Conventions refer to managerial and social rules that are commonly agreed and used by all members of a group to carry out interaction like communication, processing and decision-making [104, 146]. Simple examples of how conventions might affect people’s work include: a person changing a file name could result in an extra effort for other people to search for the file, or a person changing access permissions can affect other people’s interaction with shared artefacts. “Often unconsciously, our actions are guided by social conventions and by our awareness of the personalities and priorities of people around us” [104 p. 97]. Establishing and maintaining conventions is an important element governing cooperative activity, as stated by Mark [146] “such agreed-upon rules of interaction increase the changes that cooperative work can result in process gain, rather than process loss” (p. 349). Conventions are often not established formally, but rather are built over time upon a shared background of understanding, known as “common ground” [45], among members of a group. As an example of this, the case study of the PoliTeam system [208] showed that it took considerable time for users of PoliTeam to work out which conventions were required and suitable for their interaction.

Group awareness supports the emergence and maintenance of conventions in a group by helping people build up knowledge of how others carry out their work and how they deal with conventions. Group awareness allows people to know what artefacts other people use, what communicative channels they adopt, and how they progress with their tasks. This “common ground” affects people’s knowledge and beliefs about other people’s activity. When common information is made available and can be easily communicable in a group, members of the group can develop methods to govern their actions appropriately to avoid convention violation [146].

### 2.3.2 Problems of providing group awareness

The previous section (Section 2.3.1) examines three major benefits of group awareness in supporting cooperative activity. Unfortunately, these beneficial effects do not come for free. This section examines two major issues, privacy and disruption, that are dominant concerns for the provision of awareness information.
2.3.2.1 Privacy violation

Providing awareness information helps maintain natural and effective communication, coordination and conventions among members of a group. Yet, this information is fraught with privacy concerns [121], as remarked by Clement [47 p. 87] “it is an intrinsic feature of all CSCW applications that detailed information about personal behaviour is made available to others... Information about individuals’ activities useful for cooperation and optimizing collective performance also may become a threatening resource in the hands of others”.

A fundamental tradeoff between providing awareness information and preserving privacy has been emphasised in the literature [177]. On the one hand, the more information about other people and the workspace is conveyed, the more potential for group awareness exists in receiving information. On the other hand, the more information about other people is exposed, the more potential there is for violation of people’s privacy [49]. Obviously, people do not necessarily need to know all information about what others are doing and do not want to reveal all details about what they are doing. Finding the balance between an amount of information that needs to be provided to support group awareness and an amount of information that should not be provided to protect privacy is notoriously difficult. Achieving such a balance is inherently problematic [1, 30].

Researchers of CSCW have emphasised that privacy problems are caused by both technological and social factors [30, 59, 174]. Some studies have investigated techniques, such as blur distortion filter [29] and shadow-view filter [121], to protect users’ privacy whilst still being able to provide some degree of group awareness. Other studies, such as [4, 66, 174], have argued that privacy issues cannot be addressed by technical solutions alone because they involve social aspects that are beyond technical control.

Whilst providing awareness information is often known for privacy violation, it should be noted that the absence of awareness information might also violate privacy [17]. For example, a user could broadcast video images without having knowledge of viewers of these images. In addition, the establishment of appropriate social conventions is itself dependent on a sufficient level of group awareness, as discussed in Section 2.3.1.3.
2.3.2.2 Disruption

Instantaneous update of information for group awareness is useful in assisting seamless coordination of cooperative activity, but can be disruptive [55]. The fundamental purpose of group awareness is to support natural communication and smooth cooperative activity. Disruption causes the opposite effect and therefore, should be avoided. Disruption concerning the availability of awareness information involves two aspects:

(1) disruption due to the addition of awareness information, and

(2) disruption caused by the presentation of information itself.

First, information overload is a dominant issue in an electronic environment [33, 120]. Group awareness is known to be useful in helping users stay connected with the progress of others’ work. However, the danger is that the addition of awareness information may overwhelm users and deflect them from the essential aspects of their work [208]. Too much information often makes users conscious about the system and makes it difficult to concentrate on their tasks at hand.

Second, the presentation of awareness information can also lead to disruption. This occurs if group awareness is presented inadequately and unexpectedly [200]. Unexpected changes in the information presentation may easily distract the user from their work, especially when there is no clear mapping between the presentation of changes and the display of shared artefacts.

However, group awareness can also contribute to reducing interruption. As discussed in Section 2.3.1.2, group awareness facilitates a smooth transition between tight coupling and loose coupling. With sufficient support for group awareness, users would likely interrupt other people appropriately, thereby minimising disruption [107].

2.4 Supporting mechanisms for group awareness

This section reviews existing mechanisms that have been developed to support group awareness in synchronous distributed groupware systems. In order to make the discussion of existing awareness mechanisms manageable and easy to follow, mechanisms are examined in terms of the kind of support that they provide, including communication, coordination and conventions.
Further discussion of awareness mechanisms in specific domains, including Instant Messaging, collaborative authoring, collaborative drawing and electronic meetings, is presented later in this thesis. For example, Chapter 5 examines awareness mechanisms that are used to support communication in Instant Messaging. Chapter 6 discusses mechanisms used to support coordination in collaborative authoring.

2.4.1 Awareness mechanisms for communication

2.4.1.1 Text-based communication tools
Text-based messaging tools, such as chat and Instant Messaging (IM) applications, are commonly used to support communication between members of a distributed group. Awareness mechanisms have been developed to improve the usability of text-based messaging tools. As examples of this, IM applications provide a user list (or a buddy list) that is able to provide information about the presence of users. A user list helps people know if and when they should initiate a conversation [165]. Several chat tools, such as Babble [77], Chat Circles [246] and Coterie [209], investigate the use of graphical presentation to indicate social relations between conversants and provide an intuitive sense of context in communication. Later in the thesis, Chapter 5 presents a comprehensive review of mechanisms designed to enhance awareness in IM.

![Babble screenshot](image)

*Figure 2.2: Babble [77] provides a user list and social proxy to support awareness.*
Figure 2.2 shows examples of a user list and a social proxy implemented in Babble to enhance awareness in text-based conversation. The user list shows inactive users with parentheses around their nicknames and active users with marbles (that is, coloured dots) displayed next to their nicknames. The social proxy is designed to provide awareness information about who is currently participating in the conversation. In the social proxy, active users stay near the centre of the circle whereas inactive users drift toward the periphery of the circle.

2.4.1.2 Audio- and video-mediated systems

Audio- and video-mediated systems have been widely used to support group awareness in a distributed group [13, 26, 68]. These tools are referred to as “mediaspaces”—computer controlled networks of audio and video equipment [26, 89]. As seen in Figure 2.3a, mediaspaces provide visual and auditory information about the physical space at which team members are located. The purpose of maintaining two-way visual and auditory transmission is to provide team members with a sense of proximity and co-location. For example, mediaspaces such as Portholes [68], CoMedi [54] and Skype [206], are able to support communication and increase social awareness of people in a shared workspace. Mediaspaces are also found useful in stimulating informal communication amongst a distributed group [54, 88, 242]. Furthermore, some studies such as [243], have investigated ways to enhance conversational awareness by supporting gaze direction in video-mediated communication, as shown in Figure 2.3b.

Figure 2.3: Examples of mediaspaces. (A) Snapshot of CoMedi [54]; (B) GAZE Groupware System [243] supports gaze direction in video-mediated communication.
Mediaspaces are rich in terms of supporting virtual co-presence. Adding video links is found useful in enhancing group collaboration in some tasks, such as meetings [125]. However, some other studies such as the study conducted by Matarazoo and Sellen [148], reported that there was no significant difference in users’ performance when either providing high quality video or poor quality video. Participants of Matarazoo and Sellen’s study even completed their tasks faster using poor quality video.

Furthermore, mediaspaces are often used to show visual appearance of users and their physical spaces, but provide limited visual information about virtual shared artefacts, such as shared drawing and shared documents, over which a team collaborates. Another challenge of adopting mediaspaces concerns the issue of how to provide a rich set of information about users in order to maintain group awareness while still assuring privacy protection (discussed in Section 2.3.2.1) [54, 179].

2.4.1.3 Virtual environments
Several studies in CSCW, such as [21, 70, 194], have investigated the potential of Collaborative Virtual Environments (CVEs) to support awareness of virtual co-presence. In CVEs, people are represented as graphical embodiments, which are commonly known as avatars. People use avatars to interact with each other and with a virtual workspace [19, 196]. The illustration in Figure 2.4 shows a scenario of six avatars participating in a conversation. In some groupware systems, such as multiplayer computer games, representing people’s interactions using avatars can be manageable and has shown great success.

Figure 2.4: Avatars communicate in CVEs [196].
Another aspect of CVEs is the *augmented reality* (AR) technology that allows people to interact remotely via physical objects and tangible interfaces on which computer graphics are superimposed [23] (Figure 2.5). Users of AR systems often obtain 2D and 3D graphical views of real objects through optical head-mounted displays (HMDs). AR technology also allows users to arrange virtual images of remote users freely for more natural conversational behaviour, such as a more appropriate eye level and richer non-verbal cues [22]. However, one of the many challenges to the adoption of AR technology for supporting group collaboration is the issue of display. For example, as shown in Figure 2.5, an AR user is able to see remote desktop users’ (who do not wear HMDs) eyes through HMDs, but not vice-versa. It is because the AR user is wearing HMDs, thus their eyes are invisible to the desktop users.

![Figure 2.5: Augmented reality interfaces support remote interaction via live video avatars [23].](image)

2.4.2 Awareness mechanisms for coordination

2.4.2.1 Strict-WYSIWIS

The early form of 2D-based awareness mechanisms is *strict-WYSIWIS* (What You See Is What I See), which enforces all users to see the same view of information [210]. Strict-WYSIWIS is relatively easy to implement and conceptually easy to use. Provided with the ability to see the same view, users can identify easily what other users are doing. This helps maintain users’ awareness of others’ activities, thereby aiding coordination.

Unfortunately, the enforcement of all users seeing the same view makes strict-WYSIWIS an inflexible style of collaboration in which the users are unreasonably expected to work as a tightly-coupled unit [13]. Although strict-WYSIWIS ensures that
users can stay aware of one another’s activities, it is often too restrictive for many kinds of collaboration where people regularly move back and forth between individual and shared work [197]. As a result, a looser variation of WYSIWIS, called relaxed-WYSIWIS, has been developed to accommodate flexible interaction in group collaboration [210].

2.4.2.2 Relaxed-WYSIWIS

The concept of the relaxed-WYSIWIS interface was invented to allow users to move and change their viewports independently. Unfortunately, when users are able to view different parts of the shared workspace, they are effectively blinded to the actions happening outside their viewports. To address this limitation, several mechanisms have been developed specifically for relaxed-WYSIWIS, such as telepointers, viewports, the radar view, multi-user scrollbars and distortion-oriented views.

Telepointers

Telepointers provide information about the mouse positions and movement of other users in the shared workspace. Telepointer activity means that users are not only able to see their own mouse cursors, but also able to see other users’ mouse cursors [16, 71, 95]. In order to distinguish telepointers of different users, each telepointer is assigned a unique colour, a different shape or even has the user’s name and/or image attached to it, as shown in Figure 2.6. Telepointers have been widely adopted by many groupware systems, including systems for collaborative authoring, such as Flexible JAMM (Java Applets Made Multiuser) [16] and CoWord [258], collaborative drawing, such as GroupDraw [96], and desktop conferencing, such as Microsoft NetMeeting [155].

![Figure 2.6: Telepointers (adapted from [95]).](image)
As reported by Hayne et al. [115], the major advantage of telepointers is that they support gesturing and provide awareness information about the presence of remote users, and their activities, foci of attention and degree of interest. By watching telepointers move, a local user knows that remote users are currently working in the shared workspace. In addition, telepointers inform users of what activity is occurring and often the kind of actions as well, if short descriptions of users’ actions are attached to the telepointers, as depicted in Figure 2.6. However, telepointers fail to convey awareness information when they are removed from the local user’s view. In addition, telepointers can sometimes be disruptive when remote users move their cursors around substantially.

**Viewports**

A user’s viewport into the shared workspace indicates the area that a user can see and where they can interact within the workspace [105]. Figure 2.7 shows a snapshot of GroupDesign [13], a shared drawing application that supports viewports. In the figure, two viewports are presented as shared rectangles in the window at the right. Presenting viewports is useful in showing information about the presence and location of users in the shared workspace. The existence of the viewport indicates the presence of a user (for example, the viewport’s owner). Additionally, users’ positions are reflected by the locations of viewports.

![Figure 2.7: GroupDesign [13] supporting viewports.](image)
Furthermore, additional information, such as identity information, can be added to the presentation of viewports to make them more expressive. For example, viewports can be presented in different colours, as seen in Figure 2.7, or show users’ images [105]. Like telepointers, when viewports do not intersect, they will not be seen by remote users.

**Multi-user scrollbars**  
Multi-user scrollbars show a local user’s position and remote users’ positions in the shared workspace at the same time. The viewport of each user is represented in multi-user scrollbars as a coloured bar located at the right-hand side of the main window [10, 110]. Scrollbars of remote users can be displayed in the same region or different regions, as depicted in Figure 2.8a and Figure 2.8b, respectively. Scrollbars move up and down synchronising with the corresponding users’ movement in the workspace.

![Figure 2.8: Two variations of multi-user scrollbars. (A) Remote scrollbars align in the same region [10]; (B) Remote scrollbars located in different regions [105].](image)

Multi-user scrollbars convey awareness information about locations of other users’ activity in the shared workspace. In situations where users have completely different views of the workspace, telepointers and viewports are not useful as they are invisible, but multi-user scrollbars are still capable of conveying awareness information about remote users’ whereabouts in the workspace. In addition, multi-user scrollbars are able to provide information about relative positions amongst users. For example, whether a
local user is located in the workspace close to remote users, or whether their viewports intersect one another. Another benefit of the multi-user scrollbar is that it makes navigating to another user’s location simple by moving the local scrollbar to the same level as the remote user’s scrollbar.

As reported by Gutwin et al. [110] in their usability study of different awareness widgets, multi-user scrollbars make it difficult to interpret the exact positions in the shared workspace where users are located, what they can see, and what they are doing. In the case when remote users’ scrollbars share the same display region (Figure 2.8b), when views of more than two users intersect it is difficult to know exactly the locations of remote users because remote scrollbars overlap one another. Furthermore, multi-user scrollbars can be disruptive, especially when remote users perform a substantial amount of movement in the workspace. Since remote users’ scrollbars are shown next to the view of the shared workspace, constant movement of remote users can be distracting to a local user.

**Radar view**

The radar view renders the entire shared workspace within a miniature overview window on which users’ locations of activity are superimposed [105, 164]. In a miniature view of the shared workspace, the radar view presents the locations of users as rectangles (Figure 2.9). To indicate which rectangle belongs to whom, each rectangle can have a different text label, or a picture of the user. Telepointers can also be displayed in the radar view to indicate other users’ mouse positions and movement.

![Figure 2.9: Radar view [110].](image)
The radar view is able to provide a high-level view of the entire shared workspace. It conveys awareness information about users’ locations and activity in the workspace regardless whether or not users’ views intersect.

Although the radar view has proven to be useful in maintaining group awareness, two major issues associated with it need to be overcome. First, the major problem with a miniaturisation technique is that it has limited scalability. The low-resolution representation of the radar view conceals details of other users’ actions. The radar view of an extremely large data space contains too little detail to be useful. Consequently, to determine other users’ working locations exactly, a local user has to align viewports inconveniently by either dragging a local user’s viewport in the radar view or scrolling a local user’s detailed view. Second, as Greenberg et al. [93] pointed out, the radar view creates a “physical and contextual virtual gap” between local details and global contexts. Unnaturally, the radar view forces users to make an abrupt context shift between views of different scale when users interpret awareness information provided by the radar view.

**Distortion-oriented views**

To address the radar view’s limitations, especially to bridge the gap between the global overview of a shared document and local detail of users’ views, distortion-oriented displays are developed. Distortion-oriented visualisation techniques allow a smooth integration of the global view and local view within a single window [136]. That is, users’ positions and actions are shown within the global view whilst magnified areas show local details of users’ areas of interest. Popular awareness mechanisms utilising distortion-oriented displays include fisheye views and magnification lens [94].

*Fisheye views* include a single view that displays both local detail and global context on a continuous display [205, 211]. Fisheye views support awareness by displaying a focal point of each user and allowing users to tailor the magnification of the focal point (Figure 2.10a). By showing multiple focal points of all users, fisheye views convey awareness information about the locations of users and the details of their actions performed within the shared workspace.
Chapter 2 Group Awareness and Cooperative Work

Figure 2.10: Awareness mechanisms based on distortion-oriented displays [94]. (A) Fisheye views; (B) the Offset Lens; (C) the Head-up Lens.

Although fisheye views help improve support for group awareness, two issues need to be solved. First, since magnified regions representing other users’ working areas are adjusted by a local user, these magnified areas may not necessarily match the actual viewport size exactly. Second, difficulty of navigation is a common problem with fisheye views especially when more than two enlarged areas overlap, as pointed out by Dyck and Gutwin [70]. This is problematic, since part of the workspace appears to be lost and a local user could be misled when assuming that a hidden remote user has left the shared workspace. The reader is referred to [106, 205, 251] for a comprehensive description of fisheye views.

Magnification lens, such as the Offset Lens (Figure 2.10b) and the Head-up Lens (Figure 2.10c) [94], is a distortion-oriented display that shows the global view of a shared workspace and magnified areas of the local and remote users. Magnified areas can be shown either on the same surface as the global view with offset to the side of the area of interest (that is, the Offset Lines), or layered on top of the global view (that is, the Head-up Lens). Magnification lens-based mechanisms support awareness by allowing users to concentrate on their own tasks, such as fine-grained object
manipulation in magnified areas, while still being able to maintain actions occurring in the workspace by looking at the global view in the background.

### 2.4.2.3 Sensor-enabled mechanisms

Using sensors and personalised indicators is another approach used to provide awareness information about users’ presence and their actions. Specialised sensors are used to capture information about users’ interaction in both physical space and the shared electronic workspace. For example,

- Vertegaal et al. [244] utilised the eye-tracking device to support awareness in distributed conversation;
- Want et al. [250] used electronic badges to transmit information about users’ physical locations;
- Prinz [185] and Gellersen et al. [90] used sensors to capture events occurring in the electronic space and the physical space;
- Carroll et al. [36] developed a notification system to support users’ awareness of presence, actions and tasks; and
- Kortuem and Segall [130] and Kern [129] used wearable appliances to support awareness in interaction of a distributed group.

Sensors can also be used to support users’ awareness of artefacts in the shared workspace. In particular, sensors are used to detect operations (such as move, edit and delete) that users perform upon shared artefacts. For example, the Flexible Diff collaborative authoring tool keeps track of changes occurring in a shared document. This example is described in detail in Section 6.2.2.

In addition, groupware systems provide configurable indicators that allow users to specify how they want to be informed of events and actions captured by specialised sensors [33]. Examples of configurable indicators range from applets in Web pages [112], ticket tapes [79], awareness widgets [110] to 3D graphical presentation [70, 184].

### 2.4.3 Awareness mechanisms for conventions

Conventions are supported by a collective set of awareness mechanisms, including mechanisms designed to support communication (discussed in Section 2.4.1) and those
used to support coordination (discussed in Section 2.4.2). These awareness mechanisms assist members of a group in communicating, sharing feedback and interacting with one another to learn about social aspects of the group, more specifically, to develop appropriate rules of interaction for the group.

In comparison to awareness mechanisms developed to support communication and coordination, designing specific mechanisms to support conventions is difficult due to the complexity of the process of forming and maintaining conventions in a distributed group. As emphasised in Mark’s [146] comprehensive discussion of conventions, it takes time for members of a group to know and agree on which conventions suit them. In addition, the process of forming, following and managing conventions vary across groups and even within a group over time. Therefore, it is difficult to develop technical solutions for conventions because the issue is constrained by many social processes and factors.

Having said that, groupware systems can provide awareness mechanisms to aid the development of conventions (for example, by providing sufficient feedback of users’ actions and behaviours in the workspace). Feedback is useful in shaping users’ behaviour that affects the formation and maintenance of conventions. People give feedback to each other about how they see appropriate behaviour, which could reinforce or reverse other people’s behaviour. For example, positive feedback, such as praise and encouragement, strengthens behaviour, whereas negative feedback, such as criticism or disagreement, reverses behaviour.

Additionally, in specific domains, such as synchronous collaborative writing discussed in Chapter 6, mechanisms that provide awareness of artefacts and awareness of authorship in collaborative authoring (Section 6.2.2) can be used to support conventions.
2.5 Frameworks and models of supporting group awareness

In addition to a range of awareness mechanisms examined in the previous section (Section 2.4), significant effort has been devoted to developing generic models and frameworks\(^1\) to conceptualise the concept and implementations of group awareness.

Examples of such models include:

- Rodden’s [193] model of group awareness adopts the notion of spatial metaphors to model interaction between objects, such as people and shared artefacts, and formulate awareness between objects;

- Gutwin and Greenberg’s [108] descriptive framework of workspace awareness (a more comprehensive version of the framework is presented in [107]) identifies a set of elements of workspace awareness and proposes methods to support workspace awareness;

- Tam and Greenberg [216] extend and elaborate awareness elements addressed in Gutwin and Greenbergs’ descriptive framework to examine change awareness in asynchronous workspaces. Tam and Greenberg develop a framework that articulates what awareness information needs to be provided so that people are able to track asynchronous changes in shared workspaces over time;

- The Aether model proposed by Sandor et al. [198] describes the relations between objects in a shared workspace using semantic networks and spatial metaphors;

- Vertegaal et al. [245] extend Gutwin and Greenberg’s framework to address conversational awareness and define awareness elements in terms of attentive states;

- NESSIE (awareNESS envIronmEnt) and ENI (Event and Notification Infrastructure) developed by Gross and Prinz [100, 185] are models of a generic awareness environment. NESSIE and ENI utilise sensors and indicators (discussed above in Section 2.4.3) to capture events for group awareness;

---

\(^1\) In this thesis, the terms “model” and “framework” are used interchangeably. They are both used to refer to a conceptual description of group awareness.
The AREA model presented by Fuchs [81] and Sohlenkamp et al. [208] allows users to tailor awareness preferences to their work practice; and Simone and Bandini [204] presents Model of Modulated Awareness (MoMA) that applies a reaction-diffusion metaphor to describe the notion of group awareness.

Existing models and frameworks provide a comprehensive coverage of different aspects of group awareness. This research categorises these models into three groups based on underlying concepts that the models use. These three groups include spatial-based models, descriptive models and notification-based models.

2.5.1 Spatial-based awareness models

2.5.1.1 Rodden’s model of group awareness

Rodden [193] proposed one of the first models of group awareness. Rodden’s model applies the concept of spatial metaphors and the Spatial model of interaction [20] to gauge the level of awareness between objects in the shared workspace. The model presents a high-level abstraction of awareness, in which awareness between objects, including people, information and artefacts in a shared workspace, is manipulated via focus and nimbus.

Focus is defined as “the more an object is within your focus, the more aware you are of it” and nimbus is defined as “the more an object is within your nimbus, the more aware it is of you” [20]. By manipulating and adjusting focus and nimbus, objects manage their awareness of other objects within a shared workspace. Figure 2.11 illustrates typical scenarios of how the nimbus and focus of objects can be used to describe if and how two objects are aware of each other.

Furthermore, based on the notions of focus and nimbus, Rodden defined the continuous quantitative aspect and the concrete qualitative aspect of awareness that are used to determine the strength of awareness between two objects, and describes the relations between awareness and distance. The researcher applied the model to interpret users’ presence awareness in specific domains including shared hypertext, workflow systems, shared databases, and shared desktops.
Whilst the model is useful in understanding the level of awareness in a shared workspace, it provides little detail of what information contributes to manipulating an object’s focus and nimbus. The model also presents limited guidelines of how to represent the concepts of focus and nimbus in user interfaces of groupware systems.

2.5.1.2 The Aether model

The Aether model [198] is another influential awareness model that is developed based on the Spatial model [20] and semantic networks. Aether emphasises the need for considering group awareness as an essential and fundamental element of a groupware system architecture. Aether addresses two issues in supporting group awareness. First, the model examines the concept of awareness by reinterpreting the notion of focus and nimbus beyond the scope of virtual environments. Aether also uses the concept of semantic networks to create a space in which the nimbus and focus of objects are defined and the awareness levels between objects are calculated. Second, the concept of awareness addressed in Aether is applied to the development of groupware systems, using the idea of an “awareness engine”.

Aether extends the Spatial model by including the concepts of “time” and “medium”. According to Aether, time is one of the dimensions of the space, to which the Spatial model can be applied as to other dimensions. By including the time dimension, Aether can compute the awareness levels related to past and current events. In addition, Aether defines “medium” as a well-defined form of information which objects can “understand”. In general scenarios, information is generated by one object, carried by a...
medium and received by another object. Medium is used to define communicative relations between two objects. Sandor et al. applied the Aether model to developing groupware systems for versioning and access control.

As pointed out by Gross and Prinz [100], although Aether is sophisticated with comprehensive modelling of reality, the model is complicated, and difficult to integrate and adapt. Furthermore, it seems that Aether provides limited insights into how to support users’ awareness of context in the shared workspace. This form of context might involve such elements as goals, tasks and results, whose focus and nimbus are difficult to define.

2.5.2 Descriptive frameworks of awareness

2.5.2.1 Gutwin and Greenberg’s framework of workspace awareness

Gutwin and Greenberg [107, 108] developed a framework of workspace awareness. The framework operationalises the concept of workspace awareness by identifying a set of fundamental elements of workspace awareness. The framework also presents generic methods of capturing and presenting awareness information. As opposed to Rodden’s model (discussed in Section 2.5.1), Gutwin and Greenberg’s framework is presented in a descriptive and informal form and focuses more on design principles of awareness mechanism design. The framework includes a list of awareness elements that compose workspace awareness, as illustrated in Table 2.1.

<table>
<thead>
<tr>
<th>Category</th>
<th>Element</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who</td>
<td>Presence</td>
<td>Is anyone in the workspace?</td>
</tr>
<tr>
<td></td>
<td>Identity</td>
<td>Who is participating? Who is that?</td>
</tr>
<tr>
<td></td>
<td>Authorship</td>
<td>Who is doing that?</td>
</tr>
<tr>
<td>What</td>
<td>Action</td>
<td>What are they doing?</td>
</tr>
<tr>
<td></td>
<td>Intention</td>
<td>What goal is that action part of?</td>
</tr>
<tr>
<td></td>
<td>Artefact</td>
<td>What object are they working on?</td>
</tr>
<tr>
<td>Where</td>
<td>Location</td>
<td>Where are they working?</td>
</tr>
<tr>
<td></td>
<td>Gaze</td>
<td>Where are they looking?</td>
</tr>
<tr>
<td></td>
<td>View</td>
<td>Where can they see?</td>
</tr>
<tr>
<td></td>
<td>Reach</td>
<td>Where can they reach?</td>
</tr>
<tr>
<td>When</td>
<td>Event</td>
<td>When did that event happen?</td>
</tr>
<tr>
<td>How</td>
<td>Action</td>
<td>How did that operation happen?</td>
</tr>
<tr>
<td></td>
<td>Artefact</td>
<td>How did this artefact come to be in this state?</td>
</tr>
</tbody>
</table>

Table 2.1: Elements of workspace awareness [107].
As examples of awareness elements,

- the presence element refers to information about who is present in the shared workspace;
- the action element refers to information about what users are doing in the workspace;
- the view element concerns what users can see in the workspace; and so on.

These awareness elements are organised in terms of “Who”, “What”, “Where”, “When” and “How” categories.

In addition, the framework examines different sources of awareness information and corresponding techniques used to gather awareness information from the shared workspace. For example, awareness information can be captured from the states of users’ bodies such as their bodily actions, positions and gestures. The framework was applied to the design and development of mechanisms to support workspace awareness, including the radar view and fisheye views (described in Section 2.4.2.2).

However, the framework constrains the scope to workspace awareness that focuses primarily on users’ awareness of actions and shared artefacts. Thus, it does not address enough awareness aspects related to communication and context in the shared workspace. For example, the framework provides insufficient discussion about awareness element related to the “Why” category (for example, why does a user carry out a particular action?). This thesis applies and extends Gutwin and Greenberg’s framework to address other aspects of group awareness, such as users’ awareness of conversation, goals, tasks and results (detailed discussion is presented in Chapter 4).

### 2.5.2.2 Vertegaal et al.’s framework of awareness

Vertegaal et al. [245] adapted and extended Gutwin and Greenberg’s framework of workspace awareness to present different elements of group awareness. Adding to previous analytical frameworks of group awareness, Vertegaal et al. defined the concept of “attentive states” to model different aspects of group awareness. An attentive state is defined by the researchers as “a description of someone’s focus of attention during an activity”. An attentive state includes temporal and spatial properties of users’ attention. For example, awareness information about how active a user is in the shared workspace involves the temporal aspect of attention, and information about where users are
viewing in the workspace relates to the spatial aspect of attention. The researchers argued that awareness elements can be defined in terms of conveying attentive states of users, as shown in Table 2.2.

<table>
<thead>
<tr>
<th>Attentive state</th>
<th>Element</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Workspace awareness</td>
</tr>
<tr>
<td><strong>Locus of Attention</strong></td>
<td>Location</td>
<td>Where are they working?</td>
</tr>
<tr>
<td>(Spatial)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attention Span</strong></td>
<td>Presence</td>
<td>Who is participating?</td>
</tr>
<tr>
<td>(Temporal)</td>
<td>Activity</td>
<td>How actively are they working?</td>
</tr>
<tr>
<td><strong>Attention to Objects</strong></td>
<td>Objects</td>
<td>What object are they using or referring to?</td>
</tr>
<tr>
<td><strong>Attention to People</strong></td>
<td>People</td>
<td>Whom do they work or communicate with?</td>
</tr>
<tr>
<td><strong>Attention to Actions</strong></td>
<td>Actions</td>
<td>What action are they performing or referring to?</td>
</tr>
<tr>
<td><strong>Attention to Range</strong></td>
<td>Extents</td>
<td>What can they see?</td>
</tr>
<tr>
<td></td>
<td>Abilities</td>
<td>What can they do?</td>
</tr>
<tr>
<td></td>
<td>Influence</td>
<td>Where can they make changes?</td>
</tr>
<tr>
<td><strong>Future Attention</strong></td>
<td>Intention</td>
<td>What will they do next?</td>
</tr>
<tr>
<td>(them)</td>
<td>Expectations</td>
<td>What do they need me to do next?</td>
</tr>
<tr>
<td>(me)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.2: Presenting elements of workspace awareness and conversational awareness using attentive states [246].

By adding the notion of attentive states, Vertegaal et al. extended Gutwin and Greenberg’s framework of workspace awareness to cover some aspects of conversational awareness. Vertegaal et al. applied their model to the design of the GAZE Groupware System [243], which is used to support communication and collaboration of a distributed group. However, this framework does not address users’ awareness of context in the shared workspace.

2.5.3 Notification-based models of awareness

ENI [99] and NESSIE [185] are generic awareness environments, which utilise sensors and configurable indicators to capture and present event notifications in the shared workspace. Sensors are associated with users and shared artefacts to capture events
occurring in both the physical world and the shared electronic workspace. As examples of this, some sensors are used to detect the presence of users and other sensors are used to detect changes in a shared document. Configurable indicators allow users to specify the presentation of awareness information that is captured by sensors. Examples of indicators include awareness widgets [110] and 3D graphical presentation [184].

NESSIE and ENI are used to support awareness of context, which involves the context of origin of events, the context of users’ work, and users’ preferences of information display. Figure 2.12 illustrates the process of providing awareness context:

(1) the first step is to capture the context of events,

(2) the second step is to capture the context of users’ work, and

(3) the third step is to map users’ preferences.

To map events and context, Gross and Prinz [99] defined a list of attributes of context, such as context-name, context-location, context-artefact, and so on. These context attributes are stored in a semi-structured database using XML (eXtensible Markup Language) format [256]. Gross and Prinz integrated the model with the Basic Support for Cooperative Work (BSCW) system [7] to support awareness. As the researchers pointed out in their study of applying the model to BSCW, whilst this model can be used to support awareness in context, it is currently limited in accommodating changes of context and in dealing with consistency in mapping between events and context.
AREA [81] is another model using event notifications to support awareness. AREA defines awareness as a profile of a set of objects, operations performed upon objects, and the user’s working situation. AREA also allows users to specify the working situation and awareness preferences for which they want to be notified about a particular event. However, the main drawbacks of AREA include inflexibility in adapting to changes and insufficient mapping between events and situations [100].

2.6 Group awareness matrix

Up to this point, this chapter has presented four basic characteristics of group awareness (Section 2.2), discussed the beneficial effects and problems of supporting awareness in groupware systems (Section 2.3), reviewed common mechanisms used to support group awareness (Section 2.4) and analysed models and frameworks that are developed to conceptualise the concept of group awareness (Section 2.5).

![Group awareness matrix](image-url)
The author develops a matrix of group awareness, which is used to provide an overall picture of various aspects of group awareness discussed in this chapter (Figure 2.13). The purpose of this matrix is to summarise different fundamental facets of group awareness without engaging in detailed descriptions of these aspects. The matrix presents eight major aspects of group awareness and attributes associated to each of these aspects.

2.7 Summary
This chapter examines group awareness and its relationships with cooperative activity. To begin with, this chapter explores the concept of group awareness by reviewing different definitions that researchers of CSCW have formulated to describe group awareness. The review shows diverse opinions from the literature in realising group awareness. This diversity indicates the complexity of the concept and also demands a need for further investigation in order to understand and support group awareness.

Progressing from the discussion of group awareness’ definitions, this chapter investigates relationships between group awareness and cooperative work. More specifically, Section 2.3 presents three major benefits and two prominent issues of providing group awareness. The literature pinpoints an important role of group awareness in enhancing communication among members of a group, aiding coordination of interdependent tasks and supporting formation and management of conventions in a group. The literature also raises concerns over privacy violation and disruption if awareness information is provided inappropriately.

Section 2.4 reviews popular mechanisms that have been developed to support group awareness. Different mechanisms, ranging from text-based chat tools, graphical widgets, audio- and video-mediated systems, to virtual environments, have been examined, according to their support for communication, coordination and conventions.

Section 2.5 examines existing models and frameworks that CSCW researchers have developed to conceptualise the concept of group awareness and to provide generic guidelines of how to support group awareness. Three major models of group awareness are identified, including spatial-based models, descriptive models and notification-based models. These models are important for this research because they are highly useful in laying a solid, fundamental background of the concept of group awareness. Based on
this background, this thesis aims to develop a comprehensive framework that attends to some issues that seem to be addressed insufficient by exiting models (as discussed in Chapter 4).

Section 2.6 closes this chapter by presenting the author’s matrix of group awareness. The matrix summarises fundamental aspects of group awareness and specific attributes associated to these aspects. The matrix serves as an overview of what is involved in research on group awareness.

Based on the understanding of group awareness built up in this chapter, this thesis will further previous research on group awareness. In particular, Chapter 3 presents several empirical studies of distributed group collaboration. These studies are used to extend the already available understanding of group awareness and how people use it in collaboration. Chapter 4 presents a framework of group awareness, which is developed based in part on existing models and in part on the findings from the empirical studies. In addition, resulting mechanisms to support group awareness are discussed in Chapters 5, 6 and 7.
Chapter 3

Empirical Studies of Synchronous Distributed Groupware Systems

3.1 Introduction
The previous chapter has reviewed related research on group awareness in synchronous distributed groupware. In the literature, support for group awareness has been mostly examined based on studies of face-to-face collaboration. An analysis of how people collaborate in face-to-face conditions has been used to advance an understanding of group awareness in distributed settings. However, group collaboration conducted in a face-to-face environment, to a certain extent, differs from distributed collaboration due to substantial differences between the two environments (for example, physical settings, communication links, individual and shared views, and so on).

This chapter presents three empirical user studies of synchronous distributed groupware systems, including systems for Instant Messaging, synchronous collaborative authoring, and multi-player computer games. These groupware systems were selected for three main reasons.

- First, the study of Instant Messaging was to understand how people maintain awareness in distributed conversations. As communication is a critical aspect of any collaboration, understanding this aspect of awareness is important.

- Second, the study of collaborative authoring was to understand how people maintain awareness in distributed collaboration. Collaborative authoring tasks require people to communicate and coordinate with each other. Hence, different aspects of group awareness could be investigated and learnt from this study.

- Third, the study of computer games was to understand group awareness in a collaboration situation where people need to move back and forth frequently
between tight coupling and loose coupling. In addition, selecting computer games allows the laboratory setting of the study could be closely similar to a real-world setting where participants normally play the games, as further discussed Section 3.4.1. Additionally, the study aims to explore the potential of dual monitors in supporting group awareness (reported in Chapter 6 and Appendix C).

The results drawn from these studies are used to form a framework of group awareness, which will be introduced in the next chapter. It is important to note that, many aspects of group awareness reported in this chapter are also further discussed in Chapter 5 and Chapter 6.

3.2 Empirical study of Instant Messaging

The first empirical study presented in this chapter is a study of Instant Messaging (IM)\(^2\). This study was conducted to gain an understanding of how people maintain group awareness and identify user needs for supporting group awareness in distributed conversations.

In brief, IM supports nearly synchronous communication between people over networks. Common examples of groupware systems for IM include Microsoft MSN Messenger [161], Yahoo Messenger [259], AOL Messenger [6] and Jabber [127]. IM supports various forms of communication ranging from text-based to multimedia contents, such as graphic, audio and video. IM tools have become one of the most popular Internet-based applications that are being used by hundreds of millions of people worldwide [180, 263]. People use IM daily for both social and business purposes.

3.2.1 The method

The empirical study of IM was composed of an online survey and face-to-face interviews. The purpose of the survey was to gain an understanding of current support for group awareness in IM, whilst the interviews were used to gain a deeper understanding of how people maintain awareness in IM and identify other types of awareness support that people need from IM applications.

\(^2\) The results of this study have been published in [231-233].
3.2.1.1 Online survey
The online survey consisted of demographic multiple-choice questions, a 7-point Likert scale (disagree/agree) questionnaire and open-ended questions. Examples of questions in the survey included:

- demographic questions (for example, “When do you often appear online i.e., set your status available or visible”),
- Likert scale questions (for example, “I want to set different statuses for different people in my buddy list.” and “I don’t want to know geographical locations of other people.”), and
- open-ended questions (for example, “What other features do you want when using Instant Messengers?”).

173 participants took part in the survey, of whom 149 completed and 24 partly completed the survey. The participants comprised of 56 females (32%) and 117 males (68%). The study did not aim to analyse the different uses of IM between males and females, hence no gender balance target was set. Furthermore, according to the nature of an online survey, responses were accepted as they came. The participants were students from several universities in Australia and most of them (around 75%) were in their early twenties. One requirement of the survey was that participants must have used IM for at least three months. Most of the participants used IM regularly: 35.1% used IM many times per day, 32.4% used it a few times per day, 23.1% used it a few times per week, and 9.4% used it a few times per month (n = 173).

3.2.1.2 Face-to-face interviews
After the survey was completed, further informal, face-to-face, individual interviews with six participants were conducted in order to know more about participants’ use of IM. The participants included two females and four males, who were selected from the 173 participants of the online survey. Like the online survey, no gender balance target was set for the interviews. Thus, the six participants were selected based on a first-come, first-served approach.

The interviews were semi-structured, using open-ended questions and follow-up questions based on participants’ responses. The open-ended questions were chosen

---

3 The complete list of questions used in the survey is included in Appendix A.1.
based on the results of the online survey. For example, the participants’ responses to the need for multiple statuses and avatars (as reported in Section 3.2.2.5) came out strongly in the survey. Hence, the interviews focused on asking questions about participants’ current use of statuses and avatars, and what support they wanted from IM in relation to the two issues. Examples of questions used in the interviews are “Can you tell me about your use of avatars?”, “Why do you use avatars?” and “Why do you want to use different avatars?”. The follow-up questions were used to probe participants so that they talked more about their use of IM. Listening to participants’ stories of how they had used IM helped gain valuable insights this research would not have gained otherwise.

### 3.2.2 Findings

The study yielded much data overall. Here, the results are reported in terms of:

1. awareness of presence,
2. awareness of turn-taking,
3. awareness of conversational context,
4. awareness of emotions,
5. awareness of identities, and
6. awareness of multiple concurrent conversations.

#### 3.2.2.1 Awareness of presence

In IM, awareness of presence refers to users’ knowledge of other users’ availability. Participants in the study indicated that support for presence awareness in IM should be enhanced in two aspects: tailored setting of presence status and presence awareness in group chat.

First, participants reported that they want to be able to set different presence messages for different buddies in their contact lists. Current IM applications, such as Yahoo Messenger and Trillian [40], allow users to compose their own presence messages (for example, “On the phone”, “In a meeting”, etc.) to convey specific information about their presence. However, the applications do not support the use of multiple presence messages simultaneously. If user A sets a presence message as “In a meeting”, everyone in user A’s contact list sees this message. The study showed that nearly 75% of the respondents want IM applications to support use of different statuses with different people ($mean = 5.4; std. dev. = 1.61; n = 149$).
Another aspect of tailoring presence awareness relates to the issue of managing personal boundary control in IM. Around 75% of the respondents wanted to control who can see them online \((mean = 2.6; \text{ std. dev.} = 1.93; n = 149)\)^4. One respondent commented on the desire to go online but appear offline to some buddies. MSN Messenger provides the “Block” feature that allows a user to prevent others from knowing if the user is online. Although this technique allows users to control who can see them online, it was not well received by the participants. One participant commented, “I don’t like the Block feature, the name itself sounds so negative to me”. More interestingly, on the one hand, participants wanted to control their online visibility that is seen by others, and on the other hand, they did not like to be blocked by other users [232]. After the study was completed, Yahoo released a new version of Yahoo Messenger with the “Stealth Setting” feature that allows users to appear online to some buddies while offline to others [259].

Second, the study revealed that there is a need for improving presence awareness in a group conversation. To date, IM applications only support presence awareness of people who are currently participating in a group conversation. They do not provide any presence awareness of past participants—who were in the conversation and had already left—and future participants—who are going to join the conversation. From the study, around 65% of the respondents believed that it is important to know who were in a group conversation before \((mean = 4.5; \text{ std. dev.} = 1.71; n = 149)\), and around 55% of the respondents thought that it is useful to know who might join a group conversation \((mean = 4.6; \text{ std. dev.} = 1.63; n = 149)\) [233].

3.2.2.2 Awareness of turn-taking

Awareness of turn-taking refers to users’ perception of fundamental information about a conversation (for example, who is talking, who is listening, whose message is this, etc.). Current IM applications provide support for turn-taking mainly in a one-to-one conversation. The support for turn-taking drops significantly in a group conversation. Many awareness cues provided in a one-to-one conversation are either missing or become significantly less effective in a group conversation. For example, a “who is typing” cue is missing in a group conversation especially when more than one person is

---

^4 The value of the mean is small because this is a negative question, “I don’t want to control who can see me online”. The small mean indicates that the majority of the respondents disagree with the idea of not being able to control who can see them online.
typing at the same time. This leads to many problems in maintaining turn-taking and resolving floor control conflicts. “I rarely use group conference but I once chat with four friends and it was very difficult . . . because they were talking about many things at the same time. It was hard to follow”, commented one respondent [233].

One aspect of supporting turn-taking is to provide users with information about the visibility of viewing- and listening-in-progress in conversations. Current IM applications do not provide users with moment-by-moment information about people who are viewing text messages and/or listening to auditory messages. For example, when users are talking in voice chat, they do not know who can and who cannot hear them. The study showed that nearly 80% of the respondents said that they want to know who can hear them speaking (mean = 5.7; std. dev. = 1.72; n = 149).

In the case of a video conference, systems like Yahoo Messenger informs users of how many people are viewing their Webcam, but users do not know exactly who are those viewers. Around 80% of the respondents wanted to know which users are currently viewing their Webcam video (mean = 5.9; std. dev. = 1.74; n = 149). In addition, the respondents suggested that IM applications should provide information about whether other users can have an audio or video chat. “It is quite useful to know if my friends are capable of audio and video conversations. For example, do they have microphone, speakers or Webcam installed?”, said one participant [233].

3.2.2.3 Awareness of conversational context

The study showed that IM applications should provide better support for awareness of conversational context in IM. More than 60% of the survey participants responded that they often want to refer to earlier messages of the same conversation (mean = 4.9; std. dev. = 1.58; n = 149) but current applications provide no support for this. Thus, users have to copy the messages to which they want to refer from the quasi-shared window and paste them to a new message. This issue is similar to deictic reference, a problem of referring to objects using gestures and eye-gazes and requiring people to choose one particular instance from a collection of instances [64].

Another issue related to the support for awareness of conversational context is the ability to link emoticons to the exact message that was posted by other people. More than half of the survey respondents said that they want to link emoticons to
corresponding messages posted by other people. Although the participants found this matter rather manageable in a one-to-one conversation, it becomes significantly problematic in a group conversation where branching turns often occur [233].

3.2.2.4 Awareness of emotions

Emotions are a social need and highly important in human communication. A person’s own affective state and perception of the states of other people affect the process and outcome of a conversation [56]. This study indicated the importance of supporting the expression of emotions in IM, and showed a strong demand for providing expressive representation of emotions.

The representation of emotions allows people to express their affective states and perceive those of other conversants. As stated by the interviewed participants:

“More icons to express my mind”,

“I believe emoticons have become an important part of chatting or expressing oneself. So improving on this aspect will be very appreciative”,

“Information about a person’s mood is helpful too. If a person is in a bad mood, then you may not want to talk to that person”, and

“Many more cute and funny emoticons”.

More than 70% of the survey respondents wanted to customise emoticons so their emoticons can be different from those of other users, and customised emoticons need to be flexible enough to reflect their affective states ($mean = 4.9; std. dev. = 1.86; n = 149$). In addition to emoticons, the avatar is another graphical representation that is also commonly used by users to portray their emotional state as responded by one participant, “I use a funny avatar when I am happy” [232]. The study showed that participants also expressed their emotions through their online status. An online status is a text-based description composed by a user and can be seen by the user’s buddies. One participant stated that “I often use status to tell my friends if I am sad or happy and also edit my status to tell them what I am doing like studying, cooking, and stuff” [232].
3.2.2.5 Awareness of identities

The results of the study showed a need for supporting multiple identities by allowing users to project themselves differently to different users (referred to as awareness of identities). The necessity of providing multiple identities is shown in participants’ high response to the need for multiple avatars and multiple online statuses.

Respondents of the online survey found avatars interesting and useful. All respondents liked avatars and used them for various purposes ranging from a simple purpose, such as “I like avatar, it gives me something to look at when chatting” to a more meaningful purpose, such as expressing their mood to online friends. Around 80% of the respondents wanted to display different avatars when they chat with different people ($mean = 5.8; std. dev. = 1.6; n = 149$). One participant responded, “I often use my real photo [as avatar] when chatting with Mom, Dad and my brother. But I’d use someone else’s photos or cartoons when chatting with friends”. On a practical level, IM supports communication, but for many respondents IM is a place for social interactions and sharing interests. The study showed that avatars are effective in allowing IM to serve those purposes. As reported by one respondent “If I have an interesting picture, I used to email to my friends. But now I often set the picture as my avatar so my friends can see it. But some pictures, I don’t want all people [in the buddy list] see it”. Respondents’ comments reflected an overwhelming interest in supporting multiple avatars in IM.

While an avatar is a graphical expression of users, online status is a text-based method commonly used by IM users to project themselves to other online friends. The text-based representation makes status very flexible in supporting users to describe themselves to their online friends. “Whenever my computer is switched on, I am in Yahoo. I don’t turn off my Yahoo even when I am busy doing assignments. I don’t want to use Yahoo default status like ‘busy’ I often use something like ‘working on law assignment’ instead.” Furthermore, as reported in Section 3.2.1, there is a need for supporting multiple statuses in IM because users want to use different statuses for different people.

In addition, users frequently chat one-to-one with more than one person simultaneously, as discussed 3.2.2.6. Topics of concurrent conversations tend to be different from one
another, and users often compose their status to reflect the topic of a conversation. One participant responded that after watching the “Shrek 2” movie, she often used the “donkey + dragon = so cute” status when chatting with friends about the movie. Therefore, supporting multiple statuses allows users to customise their conversations with different people at the same time [231].

### 3.2.2.6 Awareness of multiple concurrent conversations

Awareness of multiple concurrent conversations refers to users’ perception of the presence of more than one conversation in which users are engaged. The study uncovered that it is a common behaviour of a user to interact with many people simultaneously. The majority of the respondents (around 92%) had chatted one-to-one with two or more people simultaneously (that is, multiple, concurrent, one-to-one conversations). Furthermore, more than half of them had chatted with more than three people at the same time (mean = 3.2, std. dev. = 0.09, n = 173).

The study called attention to the fact that support for awareness of multiple concurrent one-to-one conversations is an important issue, but it has not been supported sufficiently by current IM applications. As an example of this, five out of the six interviewed participants responded that at one time or another they had typed into a window that was not the one intended, especially when they were engaged in multiple conversations at the same time. One respondent commented that “My biggest problem when chatting with more than one person is maintaining a presence in each conversation, so conversations should be arranged easily. For example, I would really appreciate if it [conversation] could lock into a corner and then perhaps another chat window could be stacked beneath it or beside it so that I can understand what's going on”.

### 3.3 Empirical study of synchronous collaborative authoring

Synchronous collaborative authoring (SCA) systems allow people, who are geographically dispersed, to work together on shared documents at the same time. This section reports an empirical study of SCA, using REDUCE⁵ [213] as a testbed, referred to as “the REDUCE study”. A snapshot of REDUCE is shown in Figure 3.1. The REDUCE study aims to understand how group awareness needs to be supported in

---

⁵ The REDUCE (REal-time Distributed Unconstrained Cooperative Editing) system is a real-time collaborative editor that allows more than one user to interact synchronously upon the same document from geographically distributed sites without constraints.
Chapter 3 Empirical Studies of Synchronous Distributed Groupware Systems

SCA. The study itself was carried out as part of the author’s Master research project and has been reported in [224, 226].

Figure 3.1: A snapshot of REDUCE (two users used different text colours).

This thesis performs an additional interpretation of data collected from the REDUCE study. Initial results of the REDUCE study reported in [226] consider different awareness elements discretely, while this thesis examines those elements from four major awareness perspectives—conversational awareness, workspace awareness, contextual awareness and self-awareness. In other words, though the statistical data reported here was collected prior to this thesis, the interpretation of the data is new.

Supporting group awareness is highly important in SCA. Group awareness helps team members collaborate naturally and effectively by providing them with valuable knowledge of the status of a shared document, the presence of other team members, their activities performed upon the document, etc. Research on group awareness in SCA has drawn much interest from the CSCW and HCI communities, such as studies of [10, 152, 182].

3.3.1 The method

The REDUCE study was conducted prior to this thesis, hence, only a brief description of the experiment is presented here. The reader is referred to [224] for more detail. The study took place in Swinburne Usability Laboratory at Swinburne University of Technology. The study involved twenty participants (six females and fourteen males), who were lecturers and Ph.D. students in the School of Information Technology at the
university. Participants were allocated into ten pairs. Each pair took part in a two-and-a-half hour test session and worked on two collaborative writing tasks.

Three categories of writing tasks were selected for the REDUCE study. They were brainstorming, creative writing, and technical document preparation tasks. In the brainstorming tasks, participants were required to generate a variety of ideas and solutions to solve a particular problem. An example of the brainstorming tasks was “How do movies or television influence people’s behaviours?”. In the creative writing tasks, participants were asked to write short, argumentative essays together from scratch. For example, one of creative writing tasks was “Some people think that children should begin their formal education at a very early age and should spend most of their time on school studies. Others believe that young children should spend most of their time playing. Compare these two views. Which view do you agree with?”. In the technical document tasks, participants were free to choose a topic, with which they were both familiar and on which they felt confident to work. All technical document preparation tasks chosen by participants were to write research papers.

Each test session of the REDUCE study included training (thirty minutes), actual test (one hour), and questionnaire and interview (one hour).

- First, participants were fully trained in using REDUCE to ensure that they were familiar with the system and confident in collaboration.

- Second, pairs of participants performed two collaborative writing tasks, each for thirty minutes. One task was carried with audio communication, while another task was without audio communication.

- Third, after completing the tasks, participants filled in the questionnaire and took part in interviews to discuss awareness information and awareness mechanisms that they needed when performing the writing tasks.

3.3.2 Findings
This section discusses the results of the REDUCE study in four aspects: conversational awareness, workspace awareness, contextual awareness, and self-awareness.
3.3.2.1 Conversational awareness

Conversational awareness refers to participants’ ability to stay aware of a conversation. In the REDUCE study, participants were able to communicate with one another using audio and text-based chat tools. An observation of participants’ use of these two communication means showed that while audio conversation is faster than a text-based counterpart, there was no significant difference between these two means in facilitating group performance. More interestingly, when audio was available, it provided opportunity for informal conversations. Participants tended to talk about other matters that did not directly contribute to the completion of their tasks; instead informal talks enhanced participants’ enjoyment of collaboration, as commented by one participant “I prefer audio because it is more convenient for our team communication. Also, we seem to collaborate more when audio is used”.

In both audio and text-based communication, participants had difficulty in coherently referencing particular objects of a shared document (that is, the “deictic reference” problem [64]). Below is the dialogue that was extracted from one experiment to illustrate how inconvenient it could be when user A asked user B to look at something in the shared document.

User A: How to spell word “undeniably”? Can you correct this for me?
User B: Where is it?
User A: The last paragraph ...<user A circles the mouse around the word> ... last sentence.
User B: You are talking about the one on Richard Branson, right?

When people collaborate on a shared document, there is a contextual gap between people’s practices of gesturing (for example, pointing at an object, etc.) and their verbal communication (for example, saying “this” or “that”). Verbal communication is limited in conveying people’s gestures. To address the problem of deictic reference, the Advanced Chat mechanism [230] was proposed. Advanced Chat allows users to drag and drop objects from a shared document into a text message.

3.3.2.2 Workspace awareness

In SCA, workspace awareness refers to participants’ knowledge of the presence of other team members, their actions and the status of artefacts in a shared document.
First, knowing who collaborates on a shared document (that is, presence awareness of team members) is fundamental and important knowledge of SCA. Presence awareness involves information about people’s virtual presence as well as their physical presence. The study showed that information about virtual presence is considered more important than that of physical presence, \( p\)-value < 0.05 [230]. The majority of the participants (85%) of the REDUCE study responded that knowing “who is in workspace” is important (mean = 4.15; std. dev. = 0.81; \( n = 20 \))\(^6\). Whereas only 5% of the participants considered that it is of some importance to be informed of other people’s physical locations (mean = 1.75; std. dev. = 0.97; \( n = 20 \)). However, this result might be biased by the setting of the study in which each pair of the participants performed their tasks from two adjacent laboratory rooms. That is, each member of the pair already knew where the other member was located physically.

Second, participants responded that it is important for them to stay aware of other people’s actions and the locations in a shared document where others are working and looking. The study showed overwhelming support for the need to be aware of other people’s current actions: 95% of the participants considered it important (mean = 4.50; std. dev. = 0.61; \( n = 20 \)).

In addition, participants’ responses indicated that it is useful to provide two separate views that illustrate people’s working and viewing regions in a document. Although these two views are usually the same most of the time in an authoring session, in certain cases they differ from one another. The study uncovered that knowing other people’s working areas is considered more important than knowing their viewing areas, the values of mean are 4.50 (std. dev. = 0.61) and 3.95 (std. dev. = 0.83), respectively: \( p\)-value < 0.05 [230]. Participants proposed the Split Window View mechanism [227] to support these dual views.

Third, participants also wanted to be informed of the status of artefacts, such as text, tables and images, in a shared document. In particular, they are concerned with changes that other people made to their artefacts. The Modification Director mechanism [227] was proposed by participants to address this issue.

---

\(^6\) The original REDUCE study used a five-point Likert scale questionnaire, as shown in Appendix A.2.
Modification Director:

- notifies users immediately whenever other people modified their artefacts,
- allows users to find out quickly which part of their work was altered, and
- indicates who made the modification.

### 3.3.2.3 Contextual awareness

In addition to information about workspace awareness that involves knowledge of actions, participants were also concerned about the tasks and progress of people in a group, such as how is everyone going with their tasks and so on. Possessing these forms of knowledge is considered as *activity contextual awareness* (referred to as *contextual awareness* for short).

Almost all participants found that it is important to “know tasks for which other users are responsible” ($mean = 4.35; \text{ std. dev.} = 0.75; \ n = 20$). In the REDUCE study, participants spent a certain amount of time discussing their strategy of how to work on co-authoring tasks, and more importantly assigning responsibilities to each person.

Furthermore, participants wanted to know to what extent other people had completed their tasks. One finding of the study is the Dynamic Task List mechanism [226] that was proposed by participants. Dynamic Task List presents an active and frequently-updated list of all co-authors’ tasks. The display of Dynamic Task List is immediately updated whenever there are changes, such as when a new task is assigned, a task is modified, or a task is removed from the list. Dynamic Task List also allows users to click on a particular task to view the corresponding section of the document.

### 3.3.2.4 Self-awareness

Besides maintaining other people’s activities and progress, participants also wanted to know if other users were aware of what they had been doing with the document. This helped them to be reassured that other users were interested in or keeping track of what they had been doing. If user $A$ knows other users are keeping track of what they are doing, one can infer that user $A$ would feel much less need to communicate to other users about what they are doing, which would improve the efficiency of collaboration [229].
Along with this aspect of awareness, user A also wishes to know that other users are satisfied with the current state of user A’s contribution to the document (mean = 4.10; std. dev. = 0.64; n = 20). Furthermore, participants were concerned about to what extent they had completed their own work in comparison to the extent that other people had completed their work (mean = 3.50; std. dev. = 0.76; n = 20). This would suggest that the user derives fulfilment and confidence in making their contribution to the document. This is understandable given that co-authoring is a collaborative exercise that synergy is required in working together and that team members rely on one another’s contribution to co-author the document successfully.

3.4 Empirical study of multi-player computer games

Multi-player computer games (MCGs) have become one of the most popular groupware systems [25]. This is attested by the fact that millions of people worldwide have played MCGs on LANs (Local Area Network) and the Internet. To a certain extent, MCGs can be considered as a type of collaboration, because in many MCGs players need to coordinate their games in order to achieve particular game objectives. In addition, players of MCGs often move back and forth between their own individual games and collaboration with other players. The empirical study\(^7\) presented here was conducted to examine the need for supporting group awareness in MCGs.

One challenge in supporting group awareness is how to manage and present awareness information in such a way that it effectively and efficiently supports both a user’s own individual tasks or work components and the group’s overall collaborative task or mission. There is a fundamental trade-off between providing awareness information and managing information overload. On the one hand, the more information is conveyed about other users and the shared workspace, the greater is the potential for shared awareness based on that information. On the other hand, the more information is delivered to users, the greater is the potential for them to be overloaded by that information. This thesis argues that using an appropriate method of workspace partition can contribute to the maintenance of group awareness while keeping information load at a manageable level. Another goal of this empirical study was, therefore, to explore the usefulness of dual monitors in partitioning a shared workspace and supporting group awareness in synchronous interaction.

\(^7\) This study has been presented in [225].
The MCG study was a laboratory-based experiment, carried out in Swinburne Usability Laboratory at Swinburne University of Technology. The study involved participants playing MCGs with and without a secondary monitor provided for each participant. That is to say that under some experimental conditions, shared awareness information was displayed on this secondary monitor and in other conditions it was not.

3.4.1 The method
The study was conducted using two commercial MCGs of different genres: Counter-Strike and WarCraft III. Both games are real-time team-oriented MCGs, but they are very different in nature. The two games represented two different tasks that participants were required to complete.

Counter-Strike is a first-person-shooter game that features one team playing the role of terrorist and the other team playing the role of counter-terrorist [52]. A player of Counter-Strike controls a single virtual character from a first-person perspective. This virtual character is able to interact with other virtual characters in a game.

WarCraft III is a strategy game that requires players to lead their civilisation and defeat their opponent in a military campaign [24]. Unlike Counter-Strike, a player of WarCraft III needs to control a substantial number of different units (the number can be 100 units in many cases).

MCGs were chosen for this study for four main reasons.

- First, MCGs are a typical form of collaborative work that requires players to use their individual and teamwork skills to accomplish their goals in the game.

- Second, for assessing individual and team performance, MCGs offer a substantial advantage compared to other types of collaboration such as collaborative writing, collaborative drawing, etc. MCGs inherently provide a range of performance variables, such as individual scores, team scores, time to complete a mission, level of acquired resources, etc.

- Third, Counter-Strike and WarCraft III are two of the most played computer games on the PC platform having millions of people playing on LANs and the Internet. This would make it easy to find experienced players who had been
regularly playing those games. Having experienced players as participants was
highly useful because: (1) no training is required, and (2) participants’
familiarity with playing games in a single-monitor fashion could provide
invaluable feedback of how dual monitors affected their game experience.

- Finally, although the study was conducted in a laboratory and laboratory studies
are often criticised for being “unrealistic”, by selecting MCGs, the experiment—
including networks, PCs and auditory equipment—could be set up similar to the
real-world conditions under which participants normally play these games.

3.4.1.1 Experimental design
Two pilot tests with six participants were conducted, followed by ten full experimental
sessions with a further forty participants. The two pilot tests were conducted to establish
and verify satisfactory operation of the laboratory set up (for example, network, video
and audio recording, session timing, etc.). The results reported here are from the ten full
sessions.

Each session involved four participants and lasted for three hours including two hours
of game playing and a one-hour focus group interview. The four participants were
divided into two teams, each of two members. Each team was located in its own test
room. Two teams competed against each other in either Counter-Strike or WarCraft III.
In each test room, a large screen divider was used to isolate the two team members
visually so they could not see one another nor directly view the other person’s monitor
(Figure 3.2). However, participants in the same test room could communicate verbally
through an audio channel supported by the game. Each participant was provided with
one PC workstation connected to a LAN, and two identical 17-inch display monitors:

- a primary monitor that participants used to play the game in the usual fashion,
  and

- a secondary monitor on which appeared a cloned view of their team mate’s
  primary display.
In effect, a participant could see both their own game screen and their team player’s
screen, as shown in Figure 3.2. In some test conditions the secondary display was active
and in others it was turned off.

Sound effects are important elements in modern computer games. To avoid the
confusion of having multiple soundtracks within the same room and to facilitate
communication between team members, participants were given headphone-and-
microphone sets connected to the soundcard of their PCs.

A head-and-shoulders view of each participant was recorded using a small USB
(Universal Serial Bus) Webcam mounted on top of each participant’s primary monitor.
A laptop PC in each test room was used to capture the two Webcam signals in that
room, encoding each signal into a separate Windows Media Video (.wmv) file. The
image captured from each Webcam was 320x240 pixels at a frame rate of 15 fps.

Pilot laboratory tests indicated that this image size and rate, although small, were
sufficient to determine whether a participant was viewing the primary or the secondary
displays. The head-and-shoulders video allowed recording participants’ pattern of use of
their second monitor. The author was interested in when, during a game, participants
used the second screen and the gaze and the glance statistics of their screen use. The
overall floor plan of the laboratory is showed in Figure 3.3.
Chapter 3  Empirical Studies of Synchronous Distributed Groupware Systems

<table>
<thead>
<tr>
<th>Code</th>
<th>Item</th>
<th>Comments / explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1 – 4</td>
<td>Participant</td>
<td>P1 &amp; P2 located in test room 1. P3 &amp; P4 located in test room 2</td>
</tr>
<tr>
<td>M1 – 4</td>
<td>Primary monitor</td>
<td>Main display used by each participant</td>
</tr>
<tr>
<td>M1b – 4b</td>
<td>Secondary monitor</td>
<td>Secondary display shows a cloned view of team player’s screen</td>
</tr>
<tr>
<td>WC1 – 4</td>
<td>USB Webcam</td>
<td>Provides head-and-shoulders view of each participant.</td>
</tr>
<tr>
<td>LT1 – 2</td>
<td>Laptop PC</td>
<td>Laptop records .wmv files of Webcams and user verbal interaction</td>
</tr>
<tr>
<td>ROC1 – 2</td>
<td>Room Overview Camera</td>
<td>General view of participants</td>
</tr>
<tr>
<td>CCM1 – 2</td>
<td>Camcorder microphone</td>
<td>Sound pickup for each room giving soundtrack for recording of ROC1 – 2 and W1 – 4</td>
</tr>
</tbody>
</table>

Figure 3.3: Floor plan of the laboratory setting.

3.4.1.2 Participants

The participants were recruited by four approaches, including email messages to Swinburne University email lists, posters placed on the University campus and at Internet cafes, electronic messages posted in Internet game forums, and word of mouth.

In this study, participants are people who were more than 18 years old and had played one of the two games for at least six months. All participants were experienced game players having played the two games on LANs and the Internet, that is, 34 participants (85%) have played the games for more than one year. In addition, the majority of participants (70%) regularly played the games at least a few times per week. For this study, no gender balance target was set, and thus volunteers were accepted as they came. Consequently, the final figure of two females and thirty-eight males is quite unbalanced, though it may reflect the proportions found in the playing population for these games.

3.4.1.3 Experimental procedure

The experiments were observed by the author from the observation room. Participants’ interactions were recorded by a camera in each room. Ten experimental sessions were
run—five for Counter-Strike and five for WarCraft III. In each session, two teams competed in games under four different test conditions, called “Scenes”. As the length of each round of the WarCraft III game was often much longer than that of the Counter-Strike game (that is, thirty minutes and two minutes, respectively), different experimental procedures were designed for the two games, as shown in Table 3.1.

<table>
<thead>
<tr>
<th>Scene 1</th>
<th>Session\textsubscript{1-5} (playing CS)</th>
<th>Session\textsubscript{6-10} (playing WC3)</th>
<th>CS: Counter-strike</th>
<th>WC3: WarCraft III</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>00</td>
<td>00</td>
<td>00: none</td>
<td>11: both teams</td>
</tr>
<tr>
<td>Scene 2</td>
<td>11</td>
<td>11</td>
<td>11: both teams</td>
<td>01: only team 2</td>
</tr>
<tr>
<td>Scene 3</td>
<td>01</td>
<td>11</td>
<td>01: only team 2</td>
<td>10: only team 1</td>
</tr>
<tr>
<td>Scene 4</td>
<td>10</td>
<td>11</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: Experimental conditions.

**Procedure of Counter-strike experiment**

In the first scene, two teams played a game in the one-monitor condition with which they were already familiar. In the second scene, both teams had dual monitors. Both teams were asked to start using dual monitors at the same time so as to minimise order effects. In the third and fourth scenes, one team had dual monitors while the other did not.

In Counter-Strike, a “map” consists of a specific military task that is to be carried out in a fixed set of scenery. The same map was used throughout the experiment. During their session, the teams played each other several times, each time constituting a single “round” of the game. A round ended when the allocated military task was completed, typically when all the enemies had been killed or when a bomb had been successfully planted and activated.

The complete “scene” ended when one team won ten rounds of the game. On average, participants played sixteen rounds in each scene. That means that in one Counter-Strike experiment, each participant played sixty-four rounds of the Counter-Strike game.

**Procedure of WarCraft III experiment**

Since each round of the WarCraft III game was much longer than that of the Counter-Strike game, the study could not conduct as many rounds of WarCraft III in one scene as in the case of Counter-Strike. In fact, for each scene, the teams completed just one
round of WarCraft III. In the first scene, both teams played with single monitors and in the following three scenes, both teams played with dual monitors. Noting that each participant played only four rounds of WarCraft III and taking into account participant comments and observations during the pilot tests, for WarCraft III no mixed, dual/single conditions (that is, the “01” and “10” conditions in Table 3.1) were tested. The pilot test participants commented that they felt they needed more completed rounds of the game to adapt themselves to having dual monitors.

**Focus group interviews**

After completing four scenes, participants took part in an interview\(^8\), which was structured as a form of focus group. Participants were encouraged to share their experience of playing games with the support of dual monitors and discuss how they took advantage of dual monitors.

Focus group interviews were selected because not only do focus groups allow direct interaction with participants as in individual interviews, but they also allow participants to react to and build upon the responses of other participants [72, 160]. Since participants played games as a team, many events and actions that occurred during the experiments involved more than one participant. Thus, in many cases it was feasible to engage all the participants together in a discussion about the same incident. Overall, focus groups were particularly relevant to the purpose of the study, as they really provided an opportunity to gain insights of participants’ responses.

### 3.4.2 Findings

Here, the results of the MCG study are reported in terms of conversational awareness, workspace awareness, and self-awareness. As mentioned earlier, one objective of the MCG study was to explore the usefulness of dual monitors in facilitating group awareness. The findings relating to the dual-monitor effects are included in Appendix C and also referred to in Chapter 6, where another experiment with dual monitors is described.

#### 3.4.2.1 Conversational awareness

Participants were able to communicate with one another using text-based and audio chat. As stated in Section 3.3.2.1, conversational awareness in MCGs refers to

---

\(^8\) Questions used in interviews are included in Appendix A.3.
participants’ ability to stay aware of what is going on in a conversation. At the beginning of a game, participants often discussed their game strategies and appointed their roles, such as who would take which position in a game, when the team would do what, etc. During an actual game, participants often asked one another about their statuses and progress in a game, such as how healthy they were, which weapon they possessed, where they were located in a game, etc. It was observed from the study that participants’ discussion constantly referred to particular objects and locations in a game (for example, a box in a tunnel or a gold mine). It was found that participants’ conversation was sometimes quite confusing when participants had trouble with visualising and understanding to which object and/or location another person tried to refer. Like the case of IM conversation, deictic reference is troublesome in MCGs.

The study showed that dual monitors were found useful in supporting conversational awareness in three aspects: addressing deictic reference, simplifying verbal communication, and encouraging informal conversation.

First, dual monitors were found useful in dealing with deictic reference. For example, when the secondary monitors were available, participants spent less time exchanging and verifying information about each other’s position. They often used the image appearing on the secondary monitor to clarify the location of their fellow team players. That was when verbal communication and dual monitors worked very well together (for example, participants often viewed the secondary display while talking to the other player). A view shown on the second monitor saved participants’ time in assigning a position to each player. As a result, dual monitors were used effectively by participants to coordinate attack, especially when they were located virtually in the game close to one another, as shown in a dialog below:

\[
\begin{align*}
A: & \text{ Keep going} \\
B: & \text{ Clear up that area first . . . a sniper behind you} \\
A: & \text{ And jump out} \\
B: & \text{ He had a scout, he didn’t have a chance} \\
A: & \text{ That worked well. Do you want to do it again?} \\
B: & \text{ Yeah}
\end{align*}
\]
Second, dual monitors helped simplify verbal communication between participants. One thing observed from both Counter-Strike and WarCraft III experiments is that with the presence of dual monitors, participants were more aware of the context in which the other player was situated. At this point, it is useful to relate to Dourish and Bellotti’s explanation of content and character [67] to explain what dual monitors were able to provide. The views shown on dual monitors were able to provide participants with awareness of both content (that is, the precise location of a player and surrounding artefacts in a game) and character (that is, the direction of a player’s view in the game) aspects. This allowed participants to make sense of the other player’s activity, and as a result, only short verbal expressions would be required when exchanging messages.

Based on the observation notes and recorded videos, it was noticed that when dual monitors were available, participants often used shorter verbal expressions to convey their ideas or commands compared to when dual monitors were not available. Table 3.2 shows two representative examples of how verbal expressions differ in the two circumstances.

<table>
<thead>
<tr>
<th>Without dual monitors</th>
<th>With dual monitors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example 1</td>
<td></td>
</tr>
<tr>
<td>They’re coming from the other side near your</td>
<td>They’re coming from the other side</td>
</tr>
<tr>
<td>towers at the top</td>
<td></td>
</tr>
<tr>
<td>Example 2</td>
<td></td>
</tr>
<tr>
<td>There is one guy behind the box in bomb site A</td>
<td>There is one guy behind the box</td>
</tr>
</tbody>
</table>

Table 3.2: Examples of verbal expressions when dual monitors were or were not available.

Moreover, the study also showed that when dual monitors were available, participants asked fewer clarifying questions than when they were unavailable. Typical clarifying questions were “Are you still behind me?”, “Where are you?”, “Did you see me?”, and “Did you get the bomb?”.

Finally, dual monitors encouraged informal conversations between participants. In the study, when dual monitors were available, the content of verbal communication between participants of the same team was different, in fact richer than when dual monitors were not used. This was due to the fact that participants knew more about the other person’s activities in a game when dual monitors were used. In addition, since dual monitors were able to simplify group verbal communication, participants could talk quickly about various aspects of the game like their health, weapons, money or a
specific incident that had happened to one of the players. Such factors were only discussed briefly when necessary in the single monitor experiments. Not only are these factors important performance-related aspects of the game, but they also play a vital role in stimulating informal communication in a game. For many years, informal communication has been known clearly by researchers in several disciplines to be an extremely important factor in group collaboration, including complex coordination and problem solving activities [124, 254].

3.4.2.2 Workspace awareness

Like the case of SCA, maintaining information about workspace awareness is important in MCGs. As defined in Section 3.3.2.2, workspace awareness in MCGs involves participants’ knowledge of the presence of other team members, their actions and the status of surrounding artefacts in a game. This form of knowledge is essential to game players and affects their immediate actions as well as an ongoing game strategy. For example, in the WarCraft III game, if participants did not find enough resources, they often asked their team members for support or discussed changing game units. In the Counter-Strike game, if participants knew that their opponents possessed more powerful weapons, they often tried to hold their position and waited for back up from team members rather than making a solo attack.

First, providing information about the presence of players in a game is fundamental and essential because it helps distinguish between players of the same or different teams. Furthermore, other aspects of presence awareness such as information about the locations of players in a game are important for team coordination in a game. In both Counter-Strike and WarCraft III, presence awareness is supported by several mechanisms such as a player list, different colours, different game characters and the radar view. For example, the Counter-Strike participants frequently viewed a player list to see who still “survived” in a game. When participants’ viewports intercepted, they could recognise one another from different colours and different game characters (for example, avatars). When participants were situated at different locations in a game, the radar view allowed them to spot each other’s location quickly.

---

9 The radar view is a well-known mechanism that is used to support workspace awareness in groupware systems [110]. Both Counter-Strike and WarCraft III feature the radar view. The radar view renders the entire game map within a single miniature view on which each player’s location is superimposed (and activities in the case of WarCraft III). In Counter-Strike, a radar is a blue round window floating at top left of the screen. In WarCraft III, a radar is a square window located at the bottom left of the screen.
Second, having a knowledge of other players’ actions in a game is useful to a player’s performance. In both the Counter-Strike and WarCraft III experiments, participants frequently observed other people’s actions in a game, and adjusted their actions accordingly. For example, in one scenario of the Counter-Strike game, as soon as a participant saw his team member planning a bomb at a target, the participant immediately went to cover a door leading to the bomb site. In another scenario, when a participant realised that an opponent was sniping from the roof of a building, the participant tried to take another entry to the building.

Third, participants of both Counter-Strike and WarCraft III games were required to interact intensively with various types of artefacts in a game, such as weapons, buildings, boxes, resources, etc. The study showed that staying aware of the locations and status of game artefacts is highly important for participants. Perceiving and understanding the status of artefacts helps participants play their game confidently and effectively. For example, in one scenario of the WarCraft III game, when a participant realised that his game units were heavily outnumbered by those of opponents, the participant asked his team member for help and then quickly retreated his game units back to the team member’s military base.

It was observed from the MCG study that dual monitors were used by participants to maintain workspace awareness in two ways: navigating one another in a game and supporting the radar view.

First, secondary monitors were often used for navigation purposes. That is, participants often looked at the secondary monitors while talking to the other player. In those cases, the second monitor was used as a navigator. Participants looked at the second monitor to acquire real-time understanding of other players’ actions in a game. At that point, participants of the same team saw exactly the same view and were virtually sited in the same (game) context. Team communication was, as a result, accomplished smoothly and accurately. In fact, in all experiments, participants treated the second monitor as a navigator when they wanted to inform the other player about specific game objects and/or guide the other player about what to do and where to go in a game. The occurrence of participants watching the same view while communicating was very
similar to what often happens in face-to-face collaboration. In a normal workplace, two people often look at the same document side-by-side while discussing it.

Second, dual monitors complemented the radar view. In the interviews, participants were asked to compare the uses of the radar view and the secondary monitor. Participants indicated that the radar view was very useful, especially in WarCraft III. The radar allowed them to spot their team player’s location and a hot zone quickly—the area on the game where many activities such as shooting occurred. However, the low-resolution representation of the radar view concealed details of other players’ actions in the game. Thus, to know exactly what was happening to the other player, participants had to click on the radar view.

The radar view in WarCraft III is an active view allowing players to click on the radar to shift back and forth between the radar and the main view. The participants of WarCraft III generally responded that the radar was a very useful feature and that they were accustomed to the transition between the radar and main views in WarCraft III. Dual monitors were not found as useful as they had initially expected. Participants mostly used the radar and only occasionally watched the secondary monitors. Participants reported in the interviews that they probably needed more time to get used to the dual monitor setting in order to really get the most out of it.

On the other hand, the radar in Counter-Strike was purposely designed as a read-only or passive view. That is, a player is not allowed to click on the radar view to have a close view of what is happening with other players. As a result, dual monitors were found really helpful in the Counter-Strike experiments, because by looking at dual monitors participants could conveniently recognise both the other player’s location and what was going on with the player.

3.4.2.3 Self-awareness
The study showed that it is highly important for participants to stay aware of their own actions and their own states in a game. This type of knowledge is referred to as self-awareness. For example, in the Counter-Strike game, participants constantly checked their position in a game, their health level and what type of weapons that they possessed. Similarly, in the WarCraft III game, participants frequently checked the status of game units that they controlled (for instance, whether a building was
completely built). In an evolving shared workspace like games, participants’ ability to perceive their own actions and understand the reflection of those actions allowed them to play and coordinate games naturally and effectively.

3.5 Summary
This chapter presents three empirical user studies of distributed synchronous groupware systems, including systems for Instant Messaging, collaborative authoring and multiplayer computer games. The purposes of these studies are to understand existing support for group awareness as well as identifying user needs for further improvement.

First, the study of Instant Messaging, which comprised an online survey and face-to-face interviews, uncovered a need for improving awareness support in six aspects—presence, turn-taking, conversational context, emotions, identities and multiple concurrent conversations. Second, the analysis of an existing study of synchronous collaborative authoring that involved a laboratory-based experiment of REDUCE yielded a need for supporting several aspects of awareness, including conversational awareness, workspace awareness, contextual awareness and self-awareness. Third, the study of multi-player computer games that was conducted with two games—Counterstrike and WarCraft III—showed that it is important to enhance conversational awareness, workspace awareness and self-awareness in a game.

The next chapter of the thesis will present the F@ framework of group awareness. The formation of F@ is derived partly from the results of these three empirical studies.
PART II:
THE F@ FRAMEWORK OF
GROUP AWARENESS
Chapter 4

The F@ Framework of Group Awareness

4.1 Introduction

The past two chapters have examined related research on group awareness, and reported three empirical studies of distributed synchronous groupware. In particular, Chapter 2 analyses the advantages and limitations of existing frameworks and mechanisms of group awareness. The analysis of frameworks and mechanisms shows that there is a need for developing a more comprehensive framework of group awareness that helps improve the design process of awareness mechanisms. Chapter 3 presents three empirical studies of synchronous distributed groupware systems—Instant Messaging, collaborative authoring and multi-player computer games. These empirical studies show that many aspects of group awareness have not yet been addressed or have been supported but at an insufficient level in current groupware systems. This chapter presents the F@ framework of group awareness for synchronous distributed groupware systems\(^\text{10}\), which was developed with a twofold objective:

1. providing a better understanding of group awareness, and
2. facilitating the design of awareness mechanisms.

F@ was developed using a mixture of informal methods and formal methods to present different aspects of group awareness. These two methods have often been seen as alternatives and as competing with one another [257]. On the one hand, informal methods are often used to describe the concept and its context of use. On the other hand, formal methods aim to reduce complexity by using abstraction and formal proofs for correctness. Combining informal and formal methods is useful in developing F@ because these two methods can complement one another to fulfil the two abovementioned objectives of F@. First, informal methods are highly expressive and useful in describing the concept of group awareness. Second, formal methods are

\(^{10}\) The F@ framework has been published in [235, 236].
powerful in specifying precise requirements and properties of group awareness that assists developers in designing user interfaces for group awareness.

F@ consists of two parts: the abstract level and the concrete level. The abstract level presents in-depth descriptions of different sub-types of group awareness. The concrete level exploits temporal logic to formalise fundamental time-related aspects of group awareness. It is not a goal of the concrete level to formalise all elements of group awareness which are addressed at the abstract level. Instead, the formulas presented at the concrete level aim to demonstrate the feasibility of formalising the concept of group awareness as an approach of defining precise requirements of designing supporting mechanisms. Having said that, within the scope of this thesis, the concrete level focuses on formalising time-related aspects as they are fundamental to synchronous group collaboration (for example, information about users’ past, current and future presence and activities). The remaining aspects, which are not covered by the concrete level presented here, are left as future work.

The operational process of developing F@ is summarised in Figure 4.1. The formation of F@ is derived from existing theories of coordination and from empirical studies of distributed collaboration. The principles presented in F@ can be applied to design of awareness mechanisms. It is important to note that F@ is developed as a high-level guideline rather than a specific set of design solutions. Therefore, detailed implementations using F@ are dependent specifically upon the nature of collaboration and the culture of a group.

![Figure 4.1: Operational structure of F@.](image)

This chapter is structured as follows. To begin with, an analysis of group components is presented. This analysis is used to determine different sub-types of group awareness. Then, the details of the abstract level and the concrete levels are described in Section
4.3 and Section 4.4, respectively. Section 4.5 presents a comparative analysis of the similarities and differences between F@ and existing frameworks of group awareness.

4.2 Analysis of group components
The abstract level is a starting point of this thesis’ investigation of group awareness. The aim of the abstract level is to identify different sub-types of group awareness that are involved in supporting group collaboration. The first step of identifying sub-types of group awareness is to analyse components of a collaborating group. Knowing the set of group components helps understand the corresponding sub-types of group awareness that need to be supported. The analysis of group components has evolved in part from the model of virtual teams [140], and in part from coordination theory [144]. Although neither the model nor the theory covers awareness aspects, they provide a thorough knowledge of group components. The following sections examine these two underlying theories and define a common set of group components that are used to study group awareness.

4.2.1 Coordination theory
Malone and Crowston [144] defined coordination theory as a set of principles explaining how a group of actors can carry out their work by coordinating activities in a harmonious manner\(^\text{11}\). According to coordination theory, coordination is “the act of managing interdependencies between activities performed to achieve a goal”, and therefore it involves four components: *Actors, Activities, Goals* and *Interdependencies*. Actors are the core component who perform activities directed toward goals. Interdependencies are goal-relevant connections between activities, as depicted in Figure 4.2.

---

\(^{11}\) Malone and Crowston emphasise that working together harmoniously includes both conflicts and cooperation. A group that contains strong conflicts of interest may still produce good results.
4.2.2 Model of virtual teams

Virtual teams are groups of people who are spread out across different time and physical locations, share a common goal, and are required to perform interdependent tasks with the support of technology. Lipnack and Stamps [140] proposed the model of virtual teams based on three basic components of virtual teams: People, Purpose, and Links.

People are the core component of a virtual team, and they need to know how to be “me” (that is, independent) while simultaneously holding on to being “we” (that is, shared and integrated) in order to execute cooperative tasks smoothly. Purpose is what drives the need for forming a group. Purpose includes common goals, interdependent tasks and results. “Teams exist to produce results. Without cooperative goals, the project will never get started. And if the tasks are all independent, then a team is unnecessary” [139: p18]. Links refer to both social relations and technical connection between people of a group. Links are not limited to technology. Links include many factors such as communication channels, group interaction and trust between people.

These three components are decomposed into nine principles. Lipnack and Stamps used an input-output processing model to show the relationships between three components and nine principles, as seen in Table 4.1.

<table>
<thead>
<tr>
<th>Component</th>
<th>Input principle</th>
<th>Process principle</th>
<th>Output principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>People</td>
<td>Independent members</td>
<td>Shared responsibility</td>
<td>Integrated group</td>
</tr>
<tr>
<td>Purpose</td>
<td>Common goals</td>
<td>Interdependent tasks</td>
<td>Results</td>
</tr>
<tr>
<td>Link</td>
<td>Communication mediums</td>
<td>Interactions</td>
<td>Trustworthy relationships</td>
</tr>
</tbody>
</table>

Table 4.1: Components and principles of a virtual team [140].

4.2.3 Group components for the study of group awareness

Based on the two sets of group components addressed in coordination theory and in the model of virtual teams, the author analyses the relationships between group components from these two sets. A goal of the analysis is to examine the correlation and consistency between the two sets. The analysis allows the determination of a relevant set of group components that can be used to study group awareness. The analysis begins with something straightforward and obvious, and then moves to more complicated and less obvious relationships.
First, there is a clear one-to-one mapping between *Actors* of coordination theory and *People* of the virtual team model. They both refer to members of a collaborating group.

Second, *Goals* of coordination theory is covered as an input principle of *Purpose* in the virtual team model (that is, “Common goals”), as seen in Table 4.1.

Third, *Activities* and *Interdependencies* of coordination theory correspond to a process principle of *Purpose* in the virtual team model (that is, “Independent tasks”).

Finally, *Links* of the virtual team model is referred to as a process component underlying coordination (that is, the *Interdependencies* component) [144].

This analysis shows that two sets of group components presented in coordination theory and the model of virtual teams, although named differently, share commonalities as analysed above.

For the purpose of studying group awareness, *People* are considered as a set of many *Person* components. This separation is important as it allows modelling both “me” (i.e., *Person*) and “we” (i.e., an interaction between different *Person* and *Person* components) aspects in a collaborating group. According to the virtual team model, *Purpose* is a composition of *Goals*, *Tasks* and *Results*. *Goals* include personal and group goals. *Tasks* include independent and dependent tasks. *Results* are a product of team actions.

In addition to two components of *People* and *Purpose*, another group component, called *Shared Workspace* [107], is used. A shared workspace is a virtual environment in which people collaborate. It contains things like shared artefacts and actions that are performed upon the artefacts. The motivation of including the shared workspace component is drawn from important roles that it plays in supporting articulation work [107, 175].

In summary, *Purpose*, *People* and *Shared Workspace* are three components that are used by this research to identify categories of group awareness. The descriptions of these components and their involved factors are summarised in Table 4.2.
Abstract level of F@  

Given the three group components discussed above, the abstract level identifies four sub-types of group awareness. Each sub-type is referred to as an “awareness school” henceforth. The four awareness schools include conversational awareness [107], workspace awareness [107], contextual awareness [100] and self-awareness [191]. Figure 4.3 illustrates the relationships between the three group components and these four awareness schools.

- Conversational awareness involves interaction between Person and Person.
- Workspace awareness involves interaction between Person and Shared Workspace.
- Contextual awareness involves interaction between Person and Purpose.
- Self-awareness involves perception of Person itself.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>Purpose is what drives the need for forming a collaborating group. It includes cooperative goals toward which the group strives, interdependent tasks which the group needs to perform, and results which are a phenomenon caused by the tasks.</td>
<td>Goals, Tasks, Results</td>
</tr>
<tr>
<td>People</td>
<td>People are actors in a group who perform tasks in order to achieve the group's cooperative goals.</td>
<td>Person</td>
</tr>
<tr>
<td>Shared Workspace</td>
<td>A shared workspace is a virtual space where people of a group perform tasks directed toward the group's goals.</td>
<td>Artefacts, Activities</td>
</tr>
</tbody>
</table>

Table 4.2: Summary of group components.
4.3.1 Conversational awareness

Facilitating communication between people of a group is one of the most fundamental and vital aspects of collaboration. As discussed in Section 3.3.2.1 and Section 3.4.2.1, conversational awareness refers to people’s knowledge of a conversation. This type of knowledge includes answers to questions like, “Who is talking?”, “Who is listening?”, “Can they hear me?”, “Do they pay attention?” and “Who is talking next?”. By answering such mechanical questions, conversational awareness helps people maintain a sense of awareness of what is happening in a conversation [92, 107].

In face-to-face communication, people retain conversational awareness naturally via various verbal (for instance, intonation and expression of words) and non-verbal (for instance, eye contact and facial expressions) cues that are picked up from conversational partners. Maintaining conversational awareness allows people to conduct a conversation fluidly by adjusting and adapting their communicational behaviours in a natural and appropriate manner [105]. Support for conversational awareness has been studied comprehensively by linguistic researchers. Researchers have examined how verbal and non-verbal cues affect the way in which a conversation is conducted. For example, Clark and Brennan [44] analysed the use of verbal cues like “uh huh” and “yeah”, and non-verbal responses like head nods, for acknowledging people’s understanding of an utterance. Sacks and Schegloff [195] studied how various signals are used to manage turn-taking of a conversation.

Unfortunately, when a group is no longer co-located in the same room, the rich set of verbal and non-verbal cues, which is often naturally available in a face-to-face conversation, becomes difficult to find over distance. To address this issue, there has been a large body of CSCW research committed to investigating the benefits of mediaspaces (discussed in 2.4.1.2) for facilitating distributed conversations. For example, significant effort has been devoted to studying how synthetic audio and video links can support conversational awareness [26, 68, 78, 219, 242]. Although computer-integrated audio-video medias are useful in providing rich context of a conversation, these technologies are faced with many problems such as turn-taking control, eye contact, gesture, privacy, and so on [83, 89, 125, 131]. In addition, a physical setting largely reliant on video is often not flexible enough to deliver appropriate images for the
context of a conversation. For instance, video at times conveys incorrect eye contact [121].

In many cases, text-based links are preferred over rich medias like audio and video for several reasons, such as their low-bandwidth requirement and relaxed styles [77]. Text-based communication tools are able to support conversational awareness in their own way such as allowing copying-and-pasting contents of previous messages, colouring and highlighting emphasised words, providing textual cues of who is typing, and so on. Commonly, text-based communication tools, such as chat and Instant Messaging, often organise messages in chronological sequence. This sequential method of presenting conversation has been found inefficient in supporting conversational awareness in group discussion for several reasons, such as lack of mapping between people and their messages, no listening-in-progress, poor turn-taking support, etc. [207, 249].

In order to understand and support conversational awareness, it is useful to determine a set of components that are involved in the conduct of a conversation. According to coordination theory, conversational awareness needs to be supported in the communication process, which involves senders, receivers, messages and languages. Within the scope of this research, it is assumed that senders and receivers use the same language. Thus, conversational awareness depends on the relationships between the three components of senders, receivers and messages\(^{12}\), as shown in Table 4.3.

<table>
<thead>
<tr>
<th>Conversational awareness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group process: communication</td>
</tr>
<tr>
<td>Group components: senders, receivers, messages</td>
</tr>
</tbody>
</table>

**Table 4.3: Associated process and components of conversational awareness.**

The empirical study of Instant Messaging, which is reported in Section 3.2, showed a need for improving six major aspects of conversational awareness:

1. awareness of presence,
2. awareness of turn-taking,
3. awareness of conversational context,
4. awareness of emotions,

\(^{12}\) A message can be conveyed in various forms including, but not limited to, text, graphic, audio and video.
(5) awareness of identities, and
(6) awareness of multiple concurrent conversations.

These six aspects of conversational awareness can be interpreted from the perspective of
the three group components presented in Table 4.3. For example, awareness of presence
involves a sender’s knowledge of the availability of a receiver, and vice-versa.
Awareness of turn-taking involves a receiver’s perception of a sender who sent a
message as well as a sender’s perception of a receiver who would receive a message.

In addition to the six aspects of conversational awareness and the three group
components, it is valuable to consider a specific set of information that needs to be
provided to support conversational awareness. Adopting the approach used in Gutwin
and Greenberg’s framework of workspace awareness [107], this thesis presents
elements of knowledge that relate to conversational awareness as a set of the 5WH
dimensions—“Who”, “What”, “Where”, “When”, “Why”, and “How”—as shown in
Table 4.4.

These six dimensions articulate and categorise information that should be provided to
help people stay aware of a conversation. For example, the “Who” dimension involves
two essential awareness elements of presence and identity. These two elements refer to
people’s fundamental knowledge of anyone else’s participation in a conversation, who
they are, and a clear ownership of messages (for example, a mapping between a
message and the person sending the message). The “What” dimension concerns the
context of a conversation such as a coherent connection between messages. The “Why”
dimension describes motives behind people’s behaviour in a conversation. Knowing
why other people behave as they do in a conversation is a valuable cue to understand
and adjust conversational behaviours. However, it is extremely difficult for IM
applications to interpret people’s motives and to represent them in a conversation,
especially in an automated manner. Therefore, an acquisition of information related to
the “Why” dimension is often managed explicitly by conversants (for example, sending
messages to explain their thought and motives of their behaviour). The “How”
dimension is about a person’s actions and reaction to others’ conversational behaviour
(for example, if a person understands, agree or disagree, etc.).
Table 4.4 lists fundamental, commonsense elements and questions related to typical conversational interaction and behaviour. By no means, this list is exhaustive. It is important to note that each of these elements and questions varies or may not even exist at all depending upon the group situation. The specific information that people might require in order to stay aware of a conversation can vary from situation to situation. It depends upon the nature of conversation that is being carried out, the conversants who participate in the conversation, as well as the surrounding environment.

The 5W1H dimensions are used as probes to determine the type of information that members of a group need to maintain awareness and whether the information is supported adequately by the groupware. In addition, the questions cover the six aspects of conversational awareness listed in the previous paragraph.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Awareness element</th>
<th>Specific question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who</td>
<td>Presence (current)</td>
<td>Who participates in a conversation?</td>
</tr>
<tr>
<td></td>
<td>Presence (past)</td>
<td>Who was in a conversation?</td>
</tr>
<tr>
<td></td>
<td>Presence (future)</td>
<td>Who is going to join a conversation?</td>
</tr>
<tr>
<td></td>
<td>Identity (current)</td>
<td>Who is this person?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Who is talking?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Who is listening?</td>
</tr>
<tr>
<td></td>
<td>Identity (past)</td>
<td>Who was this person?</td>
</tr>
<tr>
<td>What</td>
<td>Context (current)</td>
<td>What does this message respond to?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>What are responses to this message?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>What is a question of this response?</td>
</tr>
<tr>
<td></td>
<td>Context (future)</td>
<td>What does a person refer to?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Who is talking next?</td>
</tr>
<tr>
<td>Where</td>
<td>Location (current)</td>
<td>Where is the latest message?</td>
</tr>
<tr>
<td>When</td>
<td>Event (past)</td>
<td>When did a person join a conversation?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When was this message sent?</td>
</tr>
<tr>
<td></td>
<td>Event (future)</td>
<td>When will a person leave a conversation?</td>
</tr>
<tr>
<td>Why</td>
<td>Motivation (current)</td>
<td>Why is a person saying that?</td>
</tr>
<tr>
<td></td>
<td>Motivation (past)</td>
<td>Why was this message sent?</td>
</tr>
<tr>
<td></td>
<td>Behaviour (current)</td>
<td>Why does a person stop talking?</td>
</tr>
<tr>
<td>How</td>
<td>Reaction (current)</td>
<td>How do other people react?</td>
</tr>
<tr>
<td></td>
<td>Reaction (past)</td>
<td>How have others changed their behaviour?</td>
</tr>
<tr>
<td></td>
<td>Reaction (future)</td>
<td>How will others react after I send this message?</td>
</tr>
</tbody>
</table>

Table 4.4: Dimensions of conversational awareness.

Conversational awareness also involves information about past and future behaviour occurring in a conversation, such as who is talking next, who is going to join a
conversation, how have people changed their behaviour, and so on. Supporting conversational awareness of the past and of the future is useful in maintaining a natural and smooth conversation. While information that supports conversational awareness of the past can be supported by the system, it is much more difficult to collect information of the future and anticipate the future.

Different facets of conversational awareness with regard to the 5W1H dimensions are shown in Table 4.4. It is not a goal of this research to present the entire complete set of possible pieces of knowledge related to conversational awareness. Instead, the table covers fundamental awareness elements, mostly common sense, in order to provide designers with a starting point in considering what needs to be supported. Furthermore, in different situations, the specific information required to make up a person’s conversational awareness varies depending on the nature of a conversation and the surrounding environment.

### 4.3.2 Workspace awareness

As reported in the empirical studies of synchronous collaborative authoring (SCA) and multi-player computer games (MCGs) (Section 3.3.2.2 and Section 3.4.2.2, respectively), workspace awareness is the state of knowing the presence of team members, the status of artefacts and the actions performed upon the artefacts in a shared workspace. According to Gutwin and Greenberg’s analysis [107], workspace awareness is considered as a continuous process of acquiring information from the workspace and integrating it with existing knowledge to support collaboration. Information available in and through the workspace helps a person maintain awareness of other people’s activities, and as a result, assists the person in performing cooperative tasks.

Gutwin and Greenberg’s framework of workspace awareness lays out a comprehensive set of elements of knowledge that needs to be provided by groupware systems. Examples of such elements include information about people’s presence (for example, is anyone in the workspace?), their identities (for example, who are participating?), actions (for example, what are they doing?), working locations (for example, where are they working?), and so forth. Furthermore, the framework also addresses the process and mechanisms by which workspace awareness is supported, for instance, the techniques of body embodiment and feedthrough.
Robertson [191] analysed the concept of public availability of actions and shared artefacts in a virtual workspace, and how the perception of availability affects workspace awareness. Robertson argued that things in a virtual workspace (for example, shared artefacts and actions) that are to be used as resources for maintaining awareness and coordinating tasks need to be made available publicly to the participants. Along the same lines, coordination theory [144] stresses that the analysis of actions and interdependencies between actions can be based on common artefacts upon which the actions are performed. In other words, shared artefacts constrain the execution of actions in a virtual workspace.

The empirical studies of SCA and MCGs yielded insights into types of information that should be provided to maintain workspace awareness. For example, the SCA study showed that users should be informed of other users’ working and viewing areas as well as being notified of modifications that others made to their artefacts. The MCG study indicated that users need to be aware of other users’ positions in a game, their actions, their game states and surrounding artefacts. Grounded on previous influential research on workspace awareness and the empirical studies of SCA and MCGs, an association between workspace awareness, group process, and group components is drawn, as seen in Table 4.5.

<table>
<thead>
<tr>
<th>Workspace awareness</th>
<th>Group process: perception of artefacts and actions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Group components: people, artefacts, actions</td>
</tr>
</tbody>
</table>

Table 4.5: Associated process and components of workspace awareness.

This thesis adopts and enhances Gutwin and Greenberg’s framework to define a fundamental set of awareness elements that can help people comprehend things happening in a shared workspace. Gutwin and Greenberg’s framework addresses five dimensions of “Who”, “What”, “Where”, “When” and “How”. To their framework, this thesis adds the “Why” dimension, as seen in Table 4.6. Similar to the case of conversational awareness, the list of awareness elements presented in Table 4.6 does not mean to be exhaustive. Instead, the list addresses fundamental and common awareness elements. Not all aspects of the 5W1H dimensions are applied to every groupware system. Rather, they vary depending on the group context.
The “Why” dimension is added to address motives of people’s actions in the workspace (for example, why is this action being executed? why is this artefact being used? etc.). Like the case of conversational awareness, gathering and presenting information about the “Why” dimension is difficult and relies substantially on the support of verbal communication.

Furthermore, the awareness elements listed in Table 4.6 are also extended to address the past and the future aspects of workspace awareness. Whilst workspace awareness of the future is mostly concerned with information about possible newcomers joining the workspace, workspace awareness of the past deals with a wide range of issues such as history of events and history of actions.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Awareness element</th>
<th>Specific question</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Who</strong></td>
<td>Presence (current)</td>
<td>Who is in the workspace?</td>
</tr>
<tr>
<td></td>
<td>Presence (past)</td>
<td>Who was in the workspace?</td>
</tr>
<tr>
<td></td>
<td>Presence (future)</td>
<td>Who is going to join the workspace?</td>
</tr>
<tr>
<td></td>
<td>Identity (current)</td>
<td>Who is that person?</td>
</tr>
<tr>
<td></td>
<td>Authorship</td>
<td>Who is performing this action?</td>
</tr>
<tr>
<td><strong>What</strong></td>
<td>Action (current)</td>
<td>What are they doing?</td>
</tr>
<tr>
<td></td>
<td>Intention (current)</td>
<td>What is the goal of performing that action?</td>
</tr>
<tr>
<td></td>
<td>Artefact (current)</td>
<td>What artefacts are they working on?</td>
</tr>
<tr>
<td><strong>Where</strong></td>
<td>Location (current)</td>
<td>Where are they working?</td>
</tr>
<tr>
<td></td>
<td>Gaze (current)</td>
<td>Where are they viewing?</td>
</tr>
<tr>
<td></td>
<td>View (current)</td>
<td>Where can they see?</td>
</tr>
<tr>
<td></td>
<td>Reach (current)</td>
<td>Where can they reach?</td>
</tr>
<tr>
<td><strong>When</strong></td>
<td>Event (past)</td>
<td>When did that event happen?</td>
</tr>
<tr>
<td></td>
<td>Action (past)</td>
<td>What did a person do?</td>
</tr>
<tr>
<td></td>
<td>Location (past)</td>
<td>Where was this artefact located before?</td>
</tr>
<tr>
<td><strong>Why</strong></td>
<td>Motivation (current)</td>
<td>Why perform this action?</td>
</tr>
<tr>
<td></td>
<td>Motivation (past)</td>
<td>Why use this artefact?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Why was this artefact changed?</td>
</tr>
<tr>
<td><strong>How</strong></td>
<td>Action (past)</td>
<td>How was that action carried out?</td>
</tr>
</tbody>
</table>

Table 4.6: Dimensions of workspace awareness, adapted from [107] and added with “Why” dimension.

### 4.3.3 Contextual awareness

In group collaboration, the need for supporting group awareness varies and changes quickly depending on work situation. Therefore, it is essential that people understand awareness information that is presented to them, as well as the context of perceived
information [117]. Broadly speaking, people’s knowledge of the context of things in a shared workspace is referred to as *contextual awareness* [100, 145].

The term “context” is elastic and has different meanings [100]. For example, the domain of collaborative authoring (Section 3.3.2.3), contextual awareness involves a knowledge of users’ tasks and progress (for example, how are other team members performing their tasks, etc.). Therefore, it would be easier to understand the term “context” when specifying it explicitly by asking the question: “Context of what?”. Previous research on group awareness has investigated contextual awareness from the perspective of:

- **Context of a physical workspace**: This type of contextual awareness involves the state of knowledge of geographical locations and physical environments in which each individual is situated such as buildings, offices and doors. Information about physical context is often captured by sensors and/or live video channels [68, 100]. For example, sensors can be attached to people and physical objects to capture phenomena occurring in the physical space, such as presence and movement [129, 185].

- **Context of social relationships**: This type of contextual awareness refers to personal relationships between members of a collaborating ensemble [12, 55]. In the literature, this type of awareness is also known as “informal awareness” [107]. Informal awareness often leads to informal and spontaneous interaction between people, and hence helps people recognise and take up opportunities for collaboration [68, 85]. Previous studies have shown that the benefits of maintaining awareness of social context are more materialised in long-term articulation work. For instance, in the field of management where the formation of coalition is beneficial, or in the area of education where socialisation is important [200].

- **Context of an organisational structure**: This form of contextual awareness involves people’s knowledge of the structure and culture of a group (for example, the understanding of rules, roles, authorships and the patterns of interactions of a group [147]). This type of awareness is also referred to as
“structural awareness” in other research [107]. Structural awareness helps a group operate smoothly by resolving possible conflicts in coordination. It often plays an important role in formal groups and/or large groups because the structure of a small group is often evolving, dynamic and less clearly defined.

- **Context of technological support**: This form of contextual awareness refers to people’s knowledge of the types of technology, such as telephone, chat tools, video and peer-to-peer architecture, that are used to support cooperative work. These days, the design of groupware systems aims to be technologically transparent. That is, end users do not need to have knowledge of what technologies are used; instead, they are more concerned about what services are available to support their tasks.

- **Context of cooperative activity**: Awareness of activity context is a broad concept that concerns people’s knowledge of tasks, activities and results in a shared workspace [50, 167]. To gain an appropriate understanding of activity contextual awareness, it is useful to consider how activity contextual awareness relates to workspace awareness. On the one hand, workspace awareness involves awareness of actions (for example, what is happening?). On the other hand, activity contextual awareness involves awareness of activities (for example, how are things going?) in a shared workspace [36]. Therefore, it could be argued that activity contextual awareness refers to people’s knowledge of an overall picture of their ongoing collaboration.

The scope of this research is focused on collaboration of a small group, thus, interest is particularly in activity contextual awareness of these five types of contextual awareness. Henceforth, whenever the term contextual awareness is referred to, it means activity contextual awareness.

Contextual awareness involves three components, including goals, tasks and results (for example, what is the overall goal, what has been done, what needs to be done, etc). In order to support activity contextual awareness, it is therefore important to provide users with information about goals, tasks and results. In addition, information about a shared workspace can be used to support sharing of context. This is because a shared
workspace specifies activities that people perform in the workspace, and this information reflects people’s tasks [100].

The relationship between goals, tasks and results is illustrated in Figure 4.4. A goal refers to a state that an individual aims to achieve. In a collaborating group, goals consist of common goals (or shared goals) and individual goals. Common goals are a set of goals that are shared by at least two members of a group, whilst an individual goal belongs to an individual member, who is responsible for achieving that goal.

Figure 4.4: Relations between goals, tasks and results, adapted from [140].

The literature shows that goals held by a group can be considered as a goal hierarchy [140, 144]. At the top of the hierarchy are common goals that are shared by all members of a group. Common goals are decomposed into many sub-goals. The process of decomposing goals can include various levels depending on the complexity of shared goals. The bottom of the goal hierarchy consists of individual goals (sometimes referred to as operational goals) that can be achieved by individual actions.

For example, a group is formed with a common goal of authoring a research paper. This common goal is divided into many sub-goals such as creating a structure of the paper, assigning tasks to team members, writing and editing different sections of the paper, etc. Tasks are bridges between a goal and a result, as seen in Figure 4.4. In other words, tasks are steps taken to accomplish a goal and to achieve results. Some tasks are sequential and dependent of one another, and some are parallel and independent. It is often that a cooperating group operates both models of tasks.
According to coordination theory, contextual awareness corresponds to the two processes of coordination and decision making, and is associated with several group components, as illustrated in Table 4.7.

<table>
<thead>
<tr>
<th>Contextual awareness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group process: coordination, decision making</td>
</tr>
<tr>
<td>Group components: people, goals, actions, tasks, results, artefacts</td>
</tr>
</tbody>
</table>

Table 4.7: Associated process and components of contextual awareness.

The empirical study of synchronous collaborative authoring (SCA) reported in Section 3.3 showed some useful insights into support for contextual awareness in SCA. For example, users need to be aware of other team members’ tasks and their progress (for example, to what extent they have completed their tasks). Users are also concerned about whether other team members are satisfied with their progress. Using the 5W1H dimensions, Table 4.8 shows a set of basic awareness elements that should be taken into account in designing user interfaces for contextual awareness.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Awareness element</th>
<th>Specific question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who</td>
<td>Role (current)</td>
<td>Who is responsible for doing this task?</td>
</tr>
<tr>
<td></td>
<td>Authorship (current)</td>
<td>Who produced this result?</td>
</tr>
<tr>
<td>What</td>
<td>Goal (current)</td>
<td>What does a group need to achieve?</td>
</tr>
<tr>
<td></td>
<td>Task (current)</td>
<td>What are a person’s tasks?</td>
</tr>
<tr>
<td></td>
<td>Result (current)</td>
<td>What has a person done?</td>
</tr>
<tr>
<td></td>
<td>Artefact (current)</td>
<td>What artefacts does a person need for this task?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>What artefacts are available?</td>
</tr>
<tr>
<td>Where</td>
<td>Result (current)</td>
<td>Where are results of this task?</td>
</tr>
<tr>
<td></td>
<td>Task (current)</td>
<td>Where is this task performed in the workspace?</td>
</tr>
<tr>
<td>When</td>
<td>Task (past)</td>
<td>When was this task completed?</td>
</tr>
<tr>
<td>Why</td>
<td>Motivation (current)</td>
<td>Why do we need to do this task?</td>
</tr>
<tr>
<td></td>
<td>Motivation (past)</td>
<td>Why did a person perform that action?</td>
</tr>
<tr>
<td>How</td>
<td>Activity (current)</td>
<td>How are things progressing in the workspace?</td>
</tr>
<tr>
<td></td>
<td>Process (current)</td>
<td>How do people progress with their tasks?</td>
</tr>
<tr>
<td></td>
<td>Result (current)</td>
<td>How effective is the result?</td>
</tr>
</tbody>
</table>

Table 4.8: Dimensions of activity contextual awareness.

The awareness elements and questions shown in Table 4.8 cover fundamental and important awareness information related to people's goals, tasks and results. Similar to the cases of conversational awareness and workspace awareness, it is important to note
that the 5W1H questions are used as probes to see whether a particular element of awareness is useful for group collaboration and whether the specific information is supported adequately by the groupware system.

4.3.4 Self-awareness
Robertson’s analysis of phenomenological perception of artefacts and actions [190, 191] indicates that “people are sentient bodies at the same time as they are available to the perceptions of themselves and others; perception is always self-perception”. In a study of group awareness, it is essential to consider both an individual’s perception of other people’s actions as well as that of their own activities.

As reported in Section 3.3.2.4 and Section 3.4.2.3, self-awareness was found an important aspect of collaboration. For example, participants of the SCA study were concerned about whether other team members were aware of what and how they had performed their tasks. In addition, participants wanted to have a knowledge of to what extent they had completed their tasks in comparison to other team members. In the case of MCGs, maintaining information about one’s own actions and state in a game was found essential. Providing feedback of participants’ own actions in a game and the effects of those actions on game artefacts helped participants adjust their gaming behaviour and how they coordinate with other team members.

Knowledge of self-awareness helps users to be aware of their own actions and locations in the shared workspace [159], especially in relation to those of other team members (for example, users know how much work they have done relatively in comparison to others), as seen in Table 4.9.

As emphasised in other three awareness schools, not every aspect of the 5W1H dimensions needs to be provided in order to support self awareness. Furthermore, the dimensions are required differently across varied group situations. Table 4.9 shows fundamental aspects of self awareness based on the other three awareness schools, addressed in Tables 4.4, 4.6 and 4.8, respectively.
An argument here is that self-awareness can be regarded as an aggregation of the other three awareness schools. Based on information that makes up conversational awareness, workspace awareness and contextual awareness, people can acquire knowledge of the relationships between their own tasks, goals, activities, results, etc. and those of other people.

From the design perspective, it may not be necessary to develop separate mechanisms that are used to support self-awareness. Instead, mechanisms that are designed to support other aspects of group awareness should consider both a local user and remote users when showing awareness information.

### 4.4 Concrete level of F@

The abstract level presents in-depth descriptions of four awareness schools: conversational awareness, workspace awareness, contextual awareness and self-awareness. Each of these four awareness schools involves the “When” dimension,
which relates to fundamental information about the past, current and future aspects of group awareness (for example, users’ actions, locations, tasks, etc).

Drawing on the importance of the “When” dimension, the concrete level aims to model fundamental time-related properties of the four awareness schools, using temporal logic as a vehicle. As mentioned earlier (Section 4.1), it is not the intention of the concrete level to formalise every aspect of group awareness that is addressed at the abstract level.

Temporal logic (TL) is an extended version of first-order logic [58] by adding temporal aspects to it. There are two types of TL: linear-time logic and branching-time logic (Figure 4.5). The former considers time as a sequence, whilst the latter views time as a sequence only up to “now”, and after that time can be divided into many branches. Linear-time logic can be seen as a special case of branching-time logic when there is only one potential branch in the future (that is, after “now”). This thesis adopts the model of linear-time logic because it is simpler and still able to reflect potential future events.

![Figure 4.5: Two types of temporal logic.](image)

In addition, time can be considered either as continuous or discrete. Continuous-time logic is complex but more accurate, and allows the representation of independent events simultaneously, whereas discrete-time logic is simpler and more efficient to implement. To reduce the complexity of the continuous-time model, discrete-time logic can be viewed as a conservative approximation of continuous-time logic [46, 176]. Therefore, this study adopts the linear discrete-time model of TL for the study of group awareness.

Let $\mathcal{J}$ be a linear-tree of time and $s$ be a node in $\mathcal{J}$. Let $\varphi$ be a proposition which can hold at some nodes in $\mathcal{J}$. A notion of proposition $\varphi$ being satisfied (that is, $\models$) at node $s$ in $\mathcal{J}$ is defined as:

\[
M(s, \mathcal{J}) \models \varphi
\]
Definitions
To begin with, this thesis presents definitions of a group, components of a group, and relationships between the components.

Definition 4.1: Group
As described at the abstract level, a collaborating group includes three components—People, Purpose, and Shared workspace. Hence, a group can be defined as a set of four elements:

\[ \text{Group} = \{ \text{People}, \text{Purpose}, \text{Workspace}, \text{Relation} \} \]

where Relation is a set of relationships between People, Purpose, Workspace and Group.

Definition 4.2: People
People denotes a set of many people, and each person is a set of many properties like name, telephone, address, qualification, etc. In group collaboration, not all properties of a person are relevant to group tasks. A person only needs to know relevant information about other people. People is a subset of Group. Hence:

\[ \text{People} \subseteq \text{Group} \]

Definition 4.3: Purpose
Purpose includes goals, tasks and results, and therefore can be described as a set of three elements:

\[ \text{Purpose} = \{ G, T, R \} \]

where \( G \) is a set of goals, \( T \) is a set of tasks, and \( R \) is a set of results.

Definition 4.4: Workspace
Workspace includes shared artefacts and actions that people perform upon the artefacts. Thus, Workspace is a pair:

\[ \text{Workspace} = \{ \text{Artefact}, \text{Action} \} \]

where Artefact is a set of artefacts, and Action is a set of actions that are performed by people People. Hence, Action \( \subseteq \text{People} \times \text{Artefact} \) (that is, actions can be regarded as relations between people and artefacts).
Definition 4.5: part_of(p, Group)
A person is said to be part of a group iff (if and only if) the person is in group Group at node s. This relation can be expressed as follows:

\[ M(s, \mathcal{I}) = \text{part_of}(p, \text{Group}) \iff M(s, \mathcal{I}) = (\exists x \in \text{People}: \text{People} \subseteq \text{Group}) (p = x) \]

In this thesis, a person is part of a group, i.e., part_of(p, Group) means that a person is actually in a shared workspace. However, the predicate part_of(p, Group) is only useful to indicate whether or not a person is in a group, and is unable to specify the temporal points at which a person joins or leaves a group. Thus, two additional relations between a person and a group are defined: join(p, Group) and leave(p, Group).

Definition 4.6: join(p, Group)
Person p joins a group at node s_i, iff the person was not part of the group at node s_{i-1} (that is, just before s_i), and becomes part of the group at s_i.

\[ M(s, \mathcal{I}) = \text{join}(p, \text{Group}) \iff M(s_{i-1}, \mathcal{I}) = \neg \text{part_of}(p, \text{Group}) \land \\
M(s, \mathcal{I}) = \text{part_of}(p, \text{Group}) \]

Definition 4.7: leave(p, Group)
A person p leaves a group at node s_i iff the person was part of the group at node s_{i-1} (that is, just before s_i), and is not part of the group at node s_i.

\[ M(s, \mathcal{I}) = \text{leave}(p, \text{Group}) \iff M(s_{i-1}, \mathcal{I}) = \text{part_of}(p, \text{Group}) \land \\
M(s, \mathcal{I}) = \neg \text{part_of}(p, \text{Group}) \]

In group collaboration, a person can join and leave a group many times during the collaboration. In this thesis, when join(p, Group) and leave(p, Group) are referred to, it is assumed that there are no other join and leave events occurring between them. The following two conditions hold. First, a person is always part of a group since the person joined a group, the operator since\textsuperscript{13} is denoted as S:

\[ M(s, \mathcal{I}) = \text{part_of}(p, \text{Group}) S \text{join}(p, \text{Group}) \]

\textsuperscript{13} \alpha S \beta \text{ holds iff } \beta \text{ had held at some state in the past and } \alpha \text{ has always held from that state until now.}
Second, a person is always part of a group until the person leaves a group, the operator until\(^\text{14}\) is denoted as \(U\):

\[
M(s, \mathcal{J}) = \text{part_of}(p, \text{Group}) U \text{leave}(p, \text{Group})
\]

From these two conditions, it is clear that a person cannot leave a group, if the person never joins the group.

**Definition 4.8: has(p, goal)**

As shown at the abstract level, in set \(\text{Purpose} = \{G, T, R\}\), tasks are bridges between goals and results, thus \(T \subseteq G \times R\). That is, for any task \(t \in T\) carried out, and directed by goal \(g \in G\), and achieves result \(r \in R\), it is considered that task \(t\) is a function that takes one variable goal \(g\) and returns result \(r\):

\[
t: g \rightarrow r
\]

Person \(p\) has a goal at node \(s_i\) iff there are task \(t\) and result \(r\), and task is a function taking a goal \(g\) and returning value \(r\).

\[
M(s, \mathcal{I}) = \text{has}(p, \text{goal}) \Leftrightarrow M(s, \mathcal{I}) = (t_p: g \rightarrow r_p),
\]

where \(j > i, p \in \text{People}, t_p \in T, r_p \in R, g_p \in G\).

**Definition 4.9: common-goal(p1, p2, goal)**

From the relation between a person and a goal defined above, a common goal is defined as follows:

\[
M(s, \mathcal{J}) = \text{common-goal}(p_1, p_2, \text{goal}) \Leftrightarrow
M(s, \mathcal{J}) = \text{has}(p_1, \text{goal}) \land
M(s, \mathcal{J}) = \text{has}(p_2, \text{goal})
\]

A goal is considered as a common goal between two people, iff each person has that goal. This formula can be recursively represented to show that there is a common goal between all people of a group.

Having presented the definitions of \(\text{Group, People, Purpose, Workspace}\) and \(\text{Relation}\), the section now describes the properties of four awareness schools including conversational awareness, workspace awareness, contextual awareness, and self-awareness.

\(^{14}\) \(\sigma U \phi\) holds iff \(\sigma\) holds continuously until \(\phi\) becomes true.
4.4.1 Conversational awareness

The abstract level examines different aspects of conversational awareness. Within the scope of this thesis, the formalisation of two aspects of conversational awareness—presence awareness and awareness of turn-taking—are presented at the concrete level. This is to illustrate the viability of formalising conversational awareness, whilst the formalisation of other aspects is left as future work.

As addressed at the abstract level, conversational awareness refers to people’s knowledge of a conversation and involves three components of senders, receivers, and messages. According to speech act theory [201], conversations can be considered as actions, because conversations change people’s knowledge of a workspace in one way or another. Speech act theory identifies five speech acts, based on the meanings they try to convey. Declarative speech act is used to tell a fact; interrogative speech act is used to ask questions; imperative speech act is used to ask a listener to do something; exclamatory speech act is used to express emotions; and performative speech act is used to achieve some conditions in the world. Despite the differences in these five speech acts, they all involve senders, receivers and messages.

Therefore, a conversation can be considered as a set of people, including senders and receivers, and messages:

\[ \text{Conversation} = \{ \text{People}, M_g \} \]

where People denotes people and \( M_g \) denotes messages.

Similar to Definition 4.6 and Definition 4.7, person \( p \) joining a conversation, and leaving a conversation can be formulated as follows:

\[
M(s_i, \mathcal{I}) \models join(p, \text{Conversation}) \iff \\
M(s_{i-1}, \mathcal{I}) \models \neg part_of(p, \text{Conversation}) \land \\
M(s_i, \mathcal{I}) \models part_of(p, \text{Conversation})
\]

\[
M(s_i, \mathcal{I}) \models leave(p, \text{Conversation}) \iff \\
M(s_{i-1}, \mathcal{I}) \models part_of(p, \text{Conversation}) \land \\
M(s_i, \mathcal{I}) \models \neg part_of(p, \text{Conversation})
\]
Presence awareness of past/ current/ future conversants

As shown in the empirical study of IM (Section 3.2.2.1) and the abstract level, one aspect of conversational awareness is to provide information about past, present and future conversants. That is, in addition to showing “current conversants” who are currently participating in a conversation, it is useful to provide information about the presence of “past conversants” who were in the conversation and had already left, and “future conversants” who might join the conversation.

Generally speaking, groupware systems can use a history of events to track the presence of past conversants and current conversants. Regarding future conversants, systems can use an invite-and-accept protocol to identify who are going to join a conversation. This section shows how TL is used to formulate the presence aspect of conversational awareness.

(a) Past conversants
Past conversants are those who were once in a conversation. Operator once is denoted as ‘♦’. Person p is considered once in conversation Conversation, iff at state $s_j$ in the past, p joined a conversation, and at another state $s_k$ after $s_j$ also in the past p left the conversation.

\[
M(s_j, \mathcal{I}) = ♦ \text{part_of}(p, \text{Conversation}) \iff \\
M(s_j, \mathcal{I}) = \text{join}(p, \text{Conversation}) \land \\
M(s_k, \mathcal{I}) = \text{leave}(p, \text{Conversation}), \text{ where } j < k < i
\]

(b) Current conversants
Current conversants are those who joined conversation Conversation at a state in the past and have not left the conversation.

\[
M(s_b, \mathcal{I}) = \text{part_of}(p, \text{Conversation}) \iff \\
M(s_j, \mathcal{I}) = \text{join}(p, \text{Conversation}) \land \\
M(s_k, \mathcal{I}) = (\forall k: j \leq k \leq i) \rightarrow \text{leave}(p, \text{Conversation})
\]

(c) Future conversants
Future conversants are those who will eventually join a conversation and are not currently in the conversation. Operator eventually is denoted as ‘◊’. Person p is

---

15 It is assumed that if a user accepts to join a conversation, the user is a future conversant.
considered eventually in conversation Conversation, iff \( p \) is not currently in the conversation now \((s_i)\), and will join the conversation later \((s_j; j > i)\).

\[
M(s_i, \mathcal{I}) = \Diamond \text{part_of}(p, \text{Conversation}) \Leftrightarrow \\
M(s_i, \mathcal{I}) = \neg \text{part_of}(p, \text{Conversation}) \land \\
M(s_j, \mathcal{I}) = \text{join}(p, \text{Conversation}), \text{ where } j > i
\]

Based on the formulas of past/ current/ future conversants above, the phenomenon by which a user is aware of the presence of conversants is defined as:

\[
M(s, \mathcal{I}) = \text{aware(user, presence-conversants)} \Leftrightarrow \\
(M(s, \mathcal{I}) = (\forall p_i; p \neq \text{user}) \Diamond \text{part_of}(p_i, \text{Conversation}) \land \text{know(user, } p_i)) \land \\
(M(s, \mathcal{I}) = (\forall p_j; p \neq \text{user}) \text{part_of}(p_j, \text{Conversation}) \land \text{know(user, } p_j)) \land \\
(M(s, \mathcal{I}) = (\forall p_k; p \neq \text{user}) \Diamond \text{part_of}(p_k, \text{Conversation}) \land \text{know(user, } p_k)),
\]

where \( p_i, p_j, p_k, \text{user} \in \text{Conversation} \).

**Awareness of turn-taking**

Awareness of turn-taking involves people’s fundamental knowledge of who is sending a message and who is receiving a message, as described at the abstract level and Section 3.2.2.2.

At node \( s_n \) during a conversation, an action of sending a message from a sender to a receiver can be modelled as:

\[
M(s_n, \mathcal{I}) = \text{send(sender, receiver, message)},
\]

where \( \text{sender, receiver, message} \in \text{Conversation} \).

Similarly, at node \( s_m \) during a conversation, an action of confirming that a receiver receives a message can be modelled as:

\[
M(s_m, \mathcal{I}) = \text{receive(receiver, message)}
\]

To support a person’s conversational awareness, the following conditions need to hold when a sender sends a message to a receiver.

First, a sender needs to be aware of the fact that the sender sends a message to a receiver:

\[
M(s, \mathcal{I}) = \text{aware(sender, send(sender, receiver, message)} \Leftrightarrow \\
M(s, \mathcal{I}) = \text{send(sender, receiver, message)}
\]
Second, a sender needs to know that the receiver actually receives a message. In the following formula, it is assumed that a transmission is real-time:

\[ M(s_j, \mathcal{J}) = \text{aware}(\text{sender}, \text{receive}(\text{receiver}, \text{message})) \iff \]
\[ M(s_i, \mathcal{J}) = \text{send}(\text{sender}, \text{receiver}, \text{message}) \land \]
\[ M(s_j, \mathcal{J}) = \text{receive}(\text{receiver}, \text{message}), \text{where } j > i \]

Third, a receiver needs to know that a sender sent the receiver a message:

\[ M(s_j, \mathcal{J}) = \text{aware}(\text{receiver}, \text{send}(\text{sender}, \text{receiver}, \text{message})) \iff \]
\[ M(s_i, \mathcal{J}) = \text{send}(\text{sender}, \text{receiver}, \text{message}) \land \]
\[ M(s_j, \mathcal{J}) = \text{receive}(\text{receiver}, \text{message}), \text{where } j > i \]

### 4.4.2 Workspace awareness
As shown at the abstract level, workspace awareness corresponds to the process level of maintaining a person’s perception of shared artefacts, other people in the workspace and actions that other people perform upon the artefacts. In this section, these three aspects of workspace awareness—presence awareness, awareness of artefacts, and awareness of actions—are formulated using TL. Like the case of conversational awareness, not all awareness elements of workspace awareness are formulated at the concrete level. Instead, fundamental awareness elements relating to temporal factors are addressed.

#### Presence awareness of group members
In the formalisation of conversational awareness presented in Section 4.4.1, presence awareness of conversants was introduced. Although the current section also addresses the presence aspect, the scope of presence awareness covered here deals with the presence of group members, who might or might not be engaged in a conversation. Having stated that, formulas of presence awareness of group members are quite akin to formulas of presence awareness of conversants.

Presence awareness of group members consists of three cases: awareness of past members (that is, who were in a group and had already left the group), current members (that is, who are currently in a group), and future members (that is, who are going to join a group).
(a) Past group members
Past members include those who were once in a group. Person p is considered once in group Group, iff at state \( s_j \) in the past, p joined a group, and at another state \( s_k \) after \( s_j \) also in the past p left the group:

\[
M(s_i, \mathcal{I}) = ♦ \text{part_of}(p, \text{Group}) \iff
M(s_j, \mathcal{I}) = \text{join}(p, \text{Group}) ∧
M(s_k, \mathcal{I}) = \text{leave}(p, \text{Group}), \text{where } j < k < i
\]

(b) Current group members
Current members are those who joined group Group and have not yet left the group.

\[
M(s_i, \mathcal{I}) = \text{part_of}(p, \text{Group}) \iff
M(s_j, \mathcal{I}) = \text{join}(p, \text{Group}) ∧
M(s_k, \mathcal{I}) = (∀k : j ≤ k ≤ i)¬\text{leave}(p, \text{Group})
\]

(c) Future group members
Future members are those who will eventually become part of a group. A person is eventually in a group iff the person is not currently in the group now and will join the group:

\[
M(s_i, \mathcal{I}) = ◇\text{part_of}(p, \text{Group}) \iff
M(s_j, \mathcal{I}) = ¬\text{part_of}(p, \text{Group}) ∧
M(s_k, \mathcal{I}) = \text{join}(p, \text{Group}), \text{where } j > i
\]

Given the formulas of past/ current/ future group members above, the phenomenon by which a person is aware of the presence of other group members is defined as:

\[
M(s_i, \mathcal{I}) = \text{aware(user, presence_of_members)} \iff
(M(s, \mathcal{I}) = (∀p_j : p ≠ user)\text{part_of}(p_j, \text{Group}) ∧ \text{know(user, p_j))} ∧
(M(s, \mathcal{I}) = (∀p_j : p ≠ user)\text{part_of}(p_j, \text{Group}) ∧ \text{know(user, p_j))} ∧
(M(s, \mathcal{I}) = (∀p_k : p ≠ user)◇\text{part_of}(p_k, \text{Group}) ∧ \text{know(user, p_k))},
\]

where \( p_i, p_j, p_k, \text{user} \in \text{Group} \).

**Awareness of artefacts**
Awareness of artefacts involves people’s understanding of states of the artefacts such as positions, values, and so on. For example, as shown in the empirical study of collaborative authoring (Section 3.3.2.2), the states of artefacts in a shared document include information about modification such as what has been changed, who made a
change, where a change occurred, and so on. In multi-player computer games (Section 3.4.2.2), awareness of artefacts involves information about the positions and status of artefacts in a game. Given there are many aspects contributing to awareness of artefacts, the focus of this section is particularly on formulating the position aspect of artefacts.

To show that an artefact is located at position \( x \) at node \( s \), the following expression is used:

\[
M(s, \mathcal{A}) \models \text{position(artefact, } x) \]

A user is said to be aware of the position of an artefact in the workspace at state \( s \), if there is location \( x \) of the object that makes \( \text{position(artefact, } x) \) true at state \( s \), and the user knows \( x \).

\[
M(s, \mathcal{A}) \models \text{aware(user, position of artefact)} \iff M(s, \mathcal{A}) \models \text{position(artefact, } x) \land \text{know(user, } x) \text{, where artefact } \in \text{Artefact}
\]

It is important to model the previous and next positions of an artefact in the workspace. Operators previous and next are denoted as ‘\( \bullet \)’ and ‘\( \circ \)’, respectively:

\[
M(s_i, \mathcal{A}) \models \bullet \text{position(artefact, } x) \iff M(s_{i-1}, \mathcal{A}) \models \text{position(artefact, } x)
\]

\[
M(s_i, \mathcal{A}) \models \circ \text{position(artefact, } y) \iff M(s_{i+1}, \mathcal{A}) \models \text{position(artefact, } y)
\]

From the previous and next operators, the position of an artefact before \( n \) time units \( \bullet_{\leq n} \) and after \( n \) time units \( \circ_{\geq n} \) can be obtained recursively. For example, proposition \( \varphi \) is true at 2 time units before the current state:

\[
\bullet_{\leq 2} \varphi \equiv \bullet(\bullet \varphi)
\]

Generally speaking, it is difficult to represent the future position of an artefact, as it is difficult to anticipate a user’s future actions. In most cases, maintaining awareness of the past and current positions of an artefact is more appropriate and important, which can be formulated as follows:

\[
M(s_i, \mathcal{A}) \models \text{aware(user, position of artefact)} \iff
M(s_i, \mathcal{A}) \models \text{position(artefact, } x) \land
M(s_i, \mathcal{A}) \models \bullet_{\leq j} \text{position(artefact, } x) \land
know(user, x) \land \text{know(user, } x),
\]

where \( 0 \leq j \leq i \), artefact \( \in \text{Artefact} \)
Awareness of actions
As described in Definition 4.4, people’s actions, Action, are relations between people People and artefacts Artefact in a shared workspace:

\[ \text{Action} \subseteq \text{People} \times \text{Artefact} \]

At state \( s \), action \( \text{action} \in \text{Action} \) performed by person \( p \in \text{People} \) upon \( \text{artefact} \in \text{Artefact} \) giving result \( \Omega \), can be formulated as:

\[
M(s, \exists) \models A^\text{artefact}_{\text{action}}(p, \text{artefact}) \rightarrow \Omega,
\]

where action denotes a type of action such as viewing, working, gazing and reaching:

\[ \text{action} \in \langle \text{viewing, working, gazing, reaching} \rangle \]

Each person’s action \( A_\alpha \) is a function that takes values of person \( p \) and artefact \( a \), and returns value \( \Omega \). A person is said to be aware of other people’s actions, if for all people of the group, the person knows \( \Omega \).

\[
M(s, \exists) \models \text{aware(user, action)} \leftarrow \quad M(s, \exists) \models (\forall p \in \text{People} : \alpha \in \text{Artefact}) A^\alpha_{\text{action}}(p, \alpha) = \Omega \land \text{know(user, } \Omega) \]

4.4.3 Contextual awareness
Contextual awareness deals with awareness information related to goals, tasks and results of people in a group. Contextual awareness provides members of a group with information about what other people are responsible for, what are their tasks, what tasks they have done, are doing or going to do, etc. As shown at the abstract level, purpose Purpose is a set of goals \( G \), tasks \( T \) and results \( R \):

\[ \text{Purpose} = \{G, T, R\}, \]

and \( T \) is a mapping between goals and results, that is, \( T \subseteq G \times R \).

\[ t : g \rightarrow r, \text{ where } t \in T, \ g \in G, \text{ and } r \in R \]

This research assumes that when a user wants to achieve a goal, the user is aware that he/she has a goal:

\[
M(s, \exists) \models \text{has(user, } g) \Rightarrow M(s, \exists) \models \text{aware(user, } \text{has(user, } g)) \]

A user is contextually aware if the user is aware of other people’s tasks, and the corresponding results produced.
\[ M(s, 3) \models aware(user, contextual\_awareness) \iff\]
\[ M(s, 3) \models (\forall p \in People) know(user, t_p : g_p \rightarrow r_p) \]

### 4.4.4 Self-awareness

Self-awareness refers to an individual’s perceptions of their own actions. As suggested by [190], this knowledge of self-perception should be considered similar to the perception of other people’s actions. All of the TL formulas that model awareness schools presented above can also be used to present people’s knowledge of their own presence and actions.

In addition, this research assumes that if a user is aware of fact \( \varphi \), then the user is also aware that s/he is aware of \( \varphi \):
\[ M(s, 3) \models aware(user, \varphi) \iff M(s, 3) \models aware(user, aware(user, \varphi)) \]

### 4.5 Comparison between F@ and other frameworks

To this point, this chapter has described both the abstract level and the concrete level of F@. Before ending this chapter, this section presents a comparative analysis between F@ and existing frameworks of group awareness which are examined in Section 2.5. The aim of this analysis is to comprehend the similarities (Table 4.10) and differences (Table 4.11) between F@ and other frameworks. The analysis compares F@ and other three major forms of group awareness frameworks, including: spatial-based models of awareness (e.g., Rodden [193] and Sandor et al. [198]), descriptive frameworks (e.g., Gutwin and Greenberg [107] and Vertegaal et al. [245]), and notification-based models (e.g., Gross and Prinz [99] and Fuchs [81]).

### Similarities

- Emphasise the importance of supporting group awareness.
- Address the need for providing awareness information and the context of perceived information.
- Indicate that supporting group awareness is a secondary goal of group collaboration, and needs to be maintained at all times.
- Confirm the necessity of maintaining awareness of other team members as well as perception of their own awareness.

| Table 4.10: Similarities between F@ and other existing frameworks of group awareness. |
Differences

The comparison of the differences between F@ and other frameworks is conducted in terms of three aspects:

1. *method* of constructing the framework,
2. *scope* of the framework, and
3. *illustrative applications* of the framework.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
</table>
| **F@**          | - F@ is developed based on existing theories of group and coordination, and on the results of empirical studies of distributed groupware.  
                 - F@ combines both informal and formal methods to present concepts and properties of group awareness. For example, the abstract level includes informal in-depth descriptions of different aspects of group awareness. Motivated by Gutwin and Greenberg's framework, the abstract level analyses group awareness in terms of awareness elements. The concrete level of F@, on the other hand, adopts the formal method of temporal logic to formalise several aspects of group awareness, in particular, those related to time (for example, past, current and future information). |
| **Spatial-based models** | - Inspired by the Spatial model of interaction, these types of models using formal notations of aura, nimbus and focus to construct and compute awareness levels. |
| **Descriptive frameworks** | - Gutwin and Greenberg's framework is developed based on the theory of situation awareness, and observational studies of face-to-face collaboration.  
                             - Gutwin and Greenberg's framework and Vertegaal's framework study workspace awareness and conversational awareness, respectively, in terms of awareness elements.  
                             - Gutwin and Greenberg's framework considers mechanisms to maintain workspace awareness (for example, capturing and understanding awareness information).  
                             - Vertegaal's framework goes further by defining two levels of awareness: macro level and micro level. The macro level involves background information that is generated prior to a meeting. The micro level deals with information inside the temporal frame of a meeting itself.  
                             - Vertegaal's framework includes the “attentive state” factor as a description of people's focus of attention during an activity, and defines awareness elements in terms of the attentive stages. |
| **Notification-based models** | - Notification-based models utilise sensors and configurable indicators to support awareness of context in a shared workspace. Sensors are used to capture events occurring in both the physical workspaces and the shared electronic workspace. Configurable indicators allow users to specify the presentation of awareness information captured by sensors.  
                             - Gross and Prinz's model uses XML to store attributes of contexts and map the attributes with users' preferences. |
### Summary

This chapter presents the F@ framework of group awareness in synchronous distributed groupware. F@ is developed based on existing theories of group and coordination, and the results of empirical studies of distributed groupware reported in Chapter 3. Two objectives of F@ include extending the current understanding of group awareness and facilitating the design of supporting mechanisms. F@ is structured into the abstract level and the concrete level.
The abstract level identifies and presents in-depth descriptions of four major sub-types of group awareness (referred to as “awareness schools”), including conversational awareness, workspace awareness, contextual awareness and self-awareness that need to be supported by groupware systems. The identification of these awareness schools is grounded primarily on two existing theories—the model of virtual teams and coordination theory. Based on these two influential studies, three components of People, Purpose and Shared Workspace are identified as common components to study group awareness. People denotes members of a collaborating group. Purpose includes shared goals that the group strives to accomplish, interdependent tasks that the group needs to perform, and results that are phenomena generated by the tasks. Shared Workspace refers to a virtual working environment in which the group collaborates.

Conversational awareness involves people’s knowledge of a conversation. Workspace awareness refers to knowledge of what things are happening in a shared workspace. Contextual awareness concerns people’s knowledge of how things are going in the workspace. Self-awareness refers to a person’s perception of their own activities. At the abstract level, descriptions of the four awareness schools also involves mapping elements of these awareness schools into the 5W1H (“Who”, “What”, “Where”, “When”, “Why” and “How”) dimensions.

Given rich descriptions of the four awareness schools presented at the abstract level, the primary purpose of the concrete level is to formalise fundamental time-related aspects of these awareness schools using temporal logic. That is, not all awareness elements that are covered at the abstract level are formulated; instead, the concrete level focuses particularly on awareness factors that are related to past, current and future information.

The formalisation of awareness aspects is useful in capturing and presenting precise requirements of what needs to be supported by awareness mechanisms. Combined with in-depth descriptions of group awareness at the abstract level, those formulas can be useful in designing relevant awareness mechanisms.

In addition, this chapter compares the similarities and differences between F@ and three other major types of frameworks of group awareness, including (1) spatial-based awareness models such as Rodden’s model and the Aether model; (2) descriptive frameworks of awareness such as Gutwin and Greenberg’s framework and Vertegaal’s framework; and (3) notification-based awareness models such as Gross and Prinz’s
model and Fuchs’ model. The comparison shows that supporting group awareness is commonly recognised as an important factor of groupware design. The comparison also realises how F@ is motivated by and distinctive from other existing frameworks in terms of the research method, scope and applications.

In the next part of the thesis, the focus turns to the application of F@ to mechanism design practice. In particular, Chapter 5 and Chapter 6 present two case studies of designing awareness support in Instant Messaging and synchronous collaborative authoring, respectively. These two chapters describes the design, implementation and evaluation of various awareness mechanisms, and discusses how F@ is reflected in the designs of resulting mechanisms.
PART III:
APPLYING F@ TO THE DESIGN OF
AWARENESS MECHANISMS
Chapter 5

Designing Awareness Support for Instant Messaging

5.1 Introduction

Chapter 4 presents the F@ framework of group awareness. F@ aims to help developers gain a better understanding of group awareness and thereby implement relevant mechanisms to support awareness. The purpose of this chapter is to present a case study of designing and building awareness mechanisms in Instant Messaging (IM)\(^\text{16}\). The design of mechanisms presented in this chapter has evolved in part from the knowledge presented in F@, and in part from the design of existing awareness mechanisms.

IM is an application that supports quasi-synchronous\(^\text{17}\) communication between people over networks. In the early days of IM, it was largely used as a text-based messaging service for teenagers’ chitchat. Many features such as voice chat, Webcam, emoticons, avatars, etc. have been added to IM over the years. Today, both text-based and multi-media messages, such as audio, video and graphic contents, are exchanged over IM networks; and IM users are no longer limited to teenagers. The service has quickly moved to the mainstream as many home users and business users find IM an easy, fast and convenient way of communication with family members, friends and colleagues [124, 165]. The population of IM home users increased 28% from 42 million in September 2000 to 53.8 million in September 2001 [180], and enterprise IM is growing at approximately 20% annually [203]. As reported in February 2003, MSN Messenger [161] alone has more than 75 million registered users [263]. Research showed that 77% of IBM employees responded that IM had enhanced their communication by reducing the time they often spent on e-mail, telephone and face-to-face communication.

\(^{16}\) The work presented in this chapter has been published in [236, 237].

\(^{17}\) The term “quasi-synchronous” was coined by Garcia and Jacobs [86] to refer to a type of synchronous transmission in which messages are composed before being transmitted, unlike the case of synchronous transmission that transmits each character as soon as it is typed.
Additionally, more than 75% of IBM’s clients, who have IM, acknowledged that IM enhanced their productivity [203].

The phrase “instant messaging” does not reflect accurately the multi-purpose use of IM. Apart from being used to send short and quick messages [248], IM applications have also been used as chat tools that allow conversations within a small group of people [98]. Popular IM applications such as MSN Messenger, AOL’s Instant Messenger (AIM) [6] and Yahoo Messenger [259] support both one-to-one conversations and group conversations. Therefore, the limitation of IM support for group discussion discussed in this chapter is also true for conventional chat applications. The results presented here can be applied to the design of chat applications.

IM is somewhat akin to face-to-face spoken conversations in terms of exchanging relatively short, quick and even incomplete sentences [64, 149, 207]. However, IM differs from face-to-face conversations in the sense that conversants of IM are distributed over distance. Thus, conversants are able to pick up only a limited set of visual and non-visual cues of their conventional partners. As a result, there is a strong need for maintaining awareness between conversants of IM conversations. For example, presence awareness of other people assists a person in deciding if they should move into conversations; and awareness of other’s turns in a conversation helps coordinate the conversation naturally and effectively.

Research on awareness in IM has gained much attention within the CSCW and HCI communities. Many studies, such as those of Cech and Condon [38], Isaacs et al. [123], Segerstad and Ljungstrand [202], Tang et al. [220], Viegas and Donath [246], and Vronay [249] have been conducted. These studies have researched various techniques to support different aspects of awareness, including awareness of users’ presence, awareness visualisation, turn-taking convention, and so on. For example, Cech and Condon’ study of turn-taking in CMC uncovered the relationship between users’ turn sizes, response times and window sizes (for example, turn sizes are often longer in the case of larger windows, while response times are faster in the case of separate windows). Isaacs et al. explored the potential of using sounds to provide awareness cues in mobile IM. Segerstad and Ljungstrand’s study of WebWho showed that both virtual and physical presence awareness affect the content of messages. Tang et al. explored the potential of awareness information in facilitating communication, especially between
mobile users. Although various useful awareness features have been included in IM applications, they are mainly used to support awareness in a single one-to-one conversation. Limited support for awareness is provided for a group conversation and multiple concurrent one-to-one conversations. The objective of this chapter is, therefore, to address these two issues of awareness support in IM.

In Chapter 4, F@ specifies four awareness schools, including conversational awareness, workspace awareness, contextual awareness and self-awareness. However, IM is used primarily to support conversations, thus conversational awareness is the most important and relevant awareness school out of these four awareness schools. To design conversational awareness support in IM, the research adopted a user-centred approach. The investigation was carried out following a three-part process, as seen in Figure 5.1:

1) understanding user needs,
2) applying an analysis of the user needs to the mechanism design of IM prototypes, and
3) evaluating the usefulness of resulting prototypes.

First, an analysis of user needs is grounded on the empirical study of IM (Section 3.2) and the concept of conversational awareness addressed in F@ (Section 4.3.1 and Section 4.4.2). Second, based on the results of the empirical study and the principles of F@, two novel prototypes—Relaxed Instant Messenger (RIM) and Conversation Dock (ConDock)—have been developed. Third, a laboratory-based test and a field trial were conducted to evaluate RIM and ConDock, respectively. The evaluations are used to confirm whether RIM and ConDock meet their design objectives, and also to reflect the use of F@ in designing of these two awareness mechanisms.
The remainder of the chapter is organised as follows. The next section reviews existing support for conversational awareness as well as identifying shortcomings on this front. Section 5.3 discusses current design metaphors of IM applications. Section 5.4 describes the mechanism design of RIM and ConDock, followed by the implementation and evaluation of these two prototypes in Section 5.5 and Section 5.6, respectively. Section 5.7 considers the possible design of other awareness mechanisms and discusses an experience in building RIM and ConDock. Section 5.8 reflects on the use of F@ in designing RIM and ConDock and identifies potential enhancements of F@.

5.2 Support for conversational awareness in IM

Conversational awareness involves people’s knowledge of a conversation. There has been great effort devoted to understanding and supporting various aspects of conversational awareness for IM. Corresponding to Section 3.2.2, this section examines support for conversational awareness in relation to six aspects: awareness of presence, awareness of turn-taking, awareness of conversational context, awareness of emotions, awareness of identity, and awareness of multiple concurrent conversations.

5.2.1 Awareness of presence

Presence awareness refers to knowledge of the availability of other people in a user’s contact list, who are often referred to as “buddies”. Supporting presence awareness is one of the most fundamental and important features in IM. It helps users decide if and when to move into conversations [165]. At a rudimentary level, presence awareness informs users if their buddies are online or offline, as shown in Figure 5.2. This feature has been implemented in all popular IM applications such as AIM, MSN Messenger, Yahoo Messenger, Gaim [84], Trillian [40] and Jabber [127]. At a higher level, IM applications incorporate many other features such as sound alerts and live video to inform when buddies come online and go offline. Hubbub [123] uses auditory cues to support presence awareness; whenever buddies go online, their “sound IDs” are played at the user’s client. IMVis [168] and Chat Circles [246] explore alternative metaphors to represent presence awareness. IMVis develops a 3D tunnel to show available buddies around the outside edge of the tunnel, and relatively less available buddies closer to the vanishing point of the tunnel. Chat Circles represents users as coloured circles. The circles expand as a new message arrives, and become blurry after a period of idleness.
Furthermore, IM users can set presence messages such as “On the phone” and “Stepped out” to inform buddies of the status of their availability (that is, they are on the phone or not in the office). In addition to showing users’ availability, some IM clients even provide the level of their activities. For example, Activity Meter [123] shows users’ level of activities within the last fifteen seconds. Chat Circles changes colours of users’ circles to indicate how active they are.

When IM becomes part of an integrated communication platform, more sophisticated support for presence awareness is required. For example, as the mobility factor is added to IM, a new degree of presence awareness is introduced. Hubbub shows if users are online and also indicates whether they logged in from their PCs or their PDAs. MOST [41] and WebWho [141] provide awareness of both virtual and physical presence. The systems display notifications if users are present in a network (i.e., virtual presence) as well as their geographical locations (i.e., physical presence). A study of WebWho showed that both virtual and physical presence awareness affect the content of messages.

However, there is limited support in current IM applications with regard to the customisation of presence settings for different buddies in a contact list, as reported in Section 3.2.2.1.

5.2.2 Awareness of turn-taking

Turn-taking is one of the fundamental processes of human conversations [64]. In face-to-face communication, turn-taking is supported by a suite of fine-grained back
channels such as body language, eye-contact, voice intonation, facial expression, and so on. Unfortunately, those fine-grained back channels are difficult to find in IM due to the distributed nature of IM conversations.

Various solutions to support awareness of turn-taking in IM have been developed. The simplest solution is that conversants explicitly offer the floor to other people by asking direct questions such as “What do you think, Bob?” However, this solution is limited as it does not suit the conversational style of IM in which exchanged messages are short and instant [64]. As a result, other alternatives have been studied. For example,

- IM applications provide awareness cues such as the textual “who is typing” indicator (Figure 5.3) in messengers of MSN, Yahoo, Gaim, Trillian, etc.;
- the visual “focusing” and “not-focusing” cues used in Hubbub [123]; and
- the auditory typing cues used in Babble [77].

![Figure 5.3: “Who is typing” cue supports turn-taking (snapshot of Trillian).](image)

Yet, effective support for organising turn-taking rules and resolving floor control conflicts is still very limited [38].

Threaded Chat [207] adopts a threaded metaphor that has been widely implemented in discussion boards to support turns and replies in chat conversations. However, the threaded metaphor suffers from several usability problems such as difficulty in navigation or difficulty in providing the focal point of a conversation, as discussed in [207, 238].

When an auditory element is added in IM, supporting turn-taking becomes even more challenging in comparison to text-based conversations. IM applications, such as MSN Messenger and Yahoo Messenger, have introduced a visual indicator that signals when a person is talking. Additionally, Woodruff and Aoki [255] examined how the push-to-talk mechanism affects users’ turn-taking conventions.
However, current IM applications provide limited awareness support for listening- and viewing-in-progress, as discussed in Section 3.2.2.2.

### 5.2.3 Awareness of conversational context

The richness of face-to-face conversations is obtained naturally by people’s implicit understanding of situational information, or context. However, this knowledge about a conversation is not easy to acquire when people communicate via IM.

Commonly, current IM applications help users stay aware of conversational context by displaying a *quasi-shared* window—a window containing messages sent by all participants in a conversation. The term “quasi-shared” is defined to refer to that window because currently local messages appear on that window instantly, but remote messages are displayed in order of their arrivals at a central server. Consequently, the order of messages shown on conversants’ screens can be different from one to another. In addition, the number of messages displayed across users can be different. This is because a user can close conversation windows at any time, and after a window is closed, past messages are cleared.

Figure 5.4 illustrates a scenario in which conversation windows of two users—Minh and exitoebay—are different from one another. Despite the possible difference in message order, the quasi-shared window is able to provide users with some degree of a common understanding of the flow of messages.

![Figure 5.4: Quasi-shared windows: Minh and exitoebay’s conversation windows are different (snapshot of Gaim).](image)

In addition, conveying information about other conversants’ activities, such as if they are typing, talking, focusing or not focusing on a chat window, helps maintain contextual awareness in IM. Chat Circles uses the cadence of size of coloured circles on a user’s screen to show the flow of conversations. Babble uses a graphical
representation called “social proxies” to show the activity that people carry out with the application. This also provides users with an intuitive sense of context in conversations. Some other IM applications, such as Gaim and Trillian, even notify users when their buddies close chat windows, and display a timeout flag if a conversation is inactive for too long. However, there is limited support in IM applications for referencing earlier messages of the conversation, as reported earlier in Section 3.2.2.3.

5.2.4 Awareness of emotions

Emotions are a social need and play an important role in human communication. Both a person’s own affective state and perception of the states of others influence the process and outcome of a conversation [56]. There has been a growing interest in providing expressive representation of emotions in IM [87].

At the most basic (but very popular and effective) level, IM users convey their emotional state like happiness, anger or sadness by using punctuations and acronyms, for example, “:-)” stands for a smiling face, “;-)” is for a winking face, “LOL” is short for laughing out loud [64]. Advancing from that, IM applications have integrated those punctuations with animated graphical emoticons [189], as seen in Figure 5.5.

![Figure 5.5: Emoticons support emotional awareness (snapshot of Trillian).](image)

Recently, Yahoo Messenger developed animated utterances called “Audibles”, and MSN Messenger introduced animated actions called “Winks”. Audibles and Winks consist of animated images and auditory tracks used to deliver messages (for example, hello, goodbye and flirt) and also to reflect the affective state of a sender and the illocutionary force of the messages. Furthermore, Conductive Chat [63] explores a new way to convey emotions by incorporating people’s skin conductivity levels into a conversation. However, current IM applications provide limited support for personalising the presentation of emotions such as customising emoticons, avatars and online status, as discussed in Section 3.2.2.4.
5.2.5 Awareness of identities

Enabling users to develop and sustain their own identities is one of the key issues in online communities [183]. In the case of IM, an identity can be developed in many ways, ranging from a rudimentary form of using different text colours to more sophisticated forms such as unique nicknames, customised avatars, unique Sound IDs, and so on. Figure 5.6 illustrates the use of avatars to portray users’ identities in Yahoo Messenger. As IM becomes popular, it is likely that one person will use IM in different domains for many purposes, such as personal and business, under different identities. Previous research, such as that of Handel and Herbsleb [113], also showed that people in the workplace often participate in many different groups. Hence, there is a need for supporting multiple identities in IM.

In the current design of IM, one username carries one virtual identity. This one-to-one model is used by the four most popular IM networks including AOL, ICQ, MSN and Yahoo. Users carry multiple identities by registering usernames with one or more networks. A user can take on many identities simultaneously by logging into different networks with different usernames. For example, a user can have two Yahoo usernames and three MSN usernames. The user can log into both Yahoo and MSN networks simultaneously using two usernames—one Yahoo and one MSN. In addition, current IM applications do not support different settings for different buddies in a user’s contact list. Thus, the user’s setting appears the same to all people in the buddy list. For example, if a Yahoo user composes a status as “Boo, I miss you”, then everyone in the user’s contact list sees the same status [231].

Figure 5.6: Customised avatars (snapshot of Yahoo Messenger).
As Turkle argued, the self in cyberspace is fragmented and multiple, but “multiplicity is not viable if it means shifting among personalities that cannot communicate... How can we be multiple and coherent at the same time?” [240: p.258]. In the case of current IM, users can create many identities by registering different usernames, but there is weak coherence between those identities, as reported earlier in Section 3.2.2.5.

### 5.2.6 Awareness of multiple concurrent conversations

In the literature, very little research examines the support for awareness of multiple concurrent one-to-one conversations. To the best of the author’s knowledge, using “tabs” to organise multiple concurrent one-to-one conversations is the only support that some IM applications such as Gaim and Trillian provide to assist users in managing multiple conversations (Figure 5.7). This technique is referred to as “tabbed conversations”.

![Figure 5.7: Tabbed conversations (snapshot of Gaim).](image)

Grouping conversations into tabs within a single frame helps save screen estate: instead of displaying many windows, tabbed conversations only require as much screen estate as one single window alone. In addition, tabbed conversations provide a visual indicator (for example, flashing colour) to inform users of the arrival of new messages at a particular conversation. The disadvantage of tabbed conversations is that the technique requires users to switch between tabs in order to read new messages at an inactive tab. Although switching tabs is a simple task, the problem lies in the fact that it forces users to leave a conversation in which they are currently engaged. The need for supporting awareness of multiple concurrent conversations was identified from the empirical study of IM, as reported in Section 3.2.2.6.
5.3 Existing design metaphors
There are three main design metaphors—sequential text-based, threaded text-based, and avatar-based—which have been used by current IM applications to present both one-to-one and group conversations.

5.3.1 Sequential text-based metaphor
The sequential text-based design has been the most dominant user interface adopted by popular IM applications such as MSN Messenger, Yahoo Messenger, etc. The design presents an electronic conversation in a list of messages which is often sorted in chronological order with the latest message at the bottom (Figure 5.8). This sequential text-based interface is simple, easy-to-use, and suitable for the fast pace of online conversations.

![Figure 5.8: Sequential text-based metaphor (snapshot of Gaim).](image)

However, the sequential metaphor is limited in many ways, including lack of mapping between people and their messages, no listening-in-progress, poor turn-taking support, obscure visual cues for facilitating communication, etc. [86, 207, 249]. For example, the display of messages in linear order is highly limited in creating a structural and logical layout of a conversation, especially in the case of group conversations. Thus, it is difficult to connect coherently between questions and answers, and between two
consecutive messages of the same person. This issue becomes more problematic when the number of users participating in a discussion increases.

In addition, given the nature of a linear sequence of messages, it is impossible for users to respond explicitly to a particular message that was previously posted by another person. Hence, users often copy-and-paste the contents of the previous message into the message they are currently composing. Furthermore, the conversational context of a group conversation is poorly supported by the sequential structure. Conventional IM applications only provide limited visual cues for turn-taking (for example, a textual “who is typing” cue). Additionally, the sequential text-based IM applications fail to indicate the point in the conversation into which a new message fits.

5.3.2 Threaded text-based metaphor

To tackle the limitations of the sequential interface, a new metaphor of a threaded layout has been studied. For example, Threaded Chat [207] organises messages into threads, and presents them in a tree-based layout (Figure 5.9). The tree-based design was reported as useful in helping users sustain their discussion, supporting turn-taking and producing a more balanced level of participation in a group.

![Figure 5.9: Threaded-based metaphor (snapshot of Threaded Chat [207]).](image)

However, Threaded Chat was found difficult to use, especially in navigating between threads. Smith et al’s evaluation of Threaded Chat showed two major usability challenges to thread-based design.
First, the focal point of a conversation is not supported by a message tree. It is difficult to show a node in the tree at which a new message is located. Unlike conventional IM applications in which a new message is always pushed to the bottom of the linear stack, the tree layout allows new messages to be displayed anywhere on the tree. As a result, it becomes difficult to keep track of new messages, who posted the latest messages, etc. The problem of missing the focal point forces the users to scroll up and down frequently in order to track the arrival of new messages. As the size of the tree (that is, the number of branches and levels) grows, this becomes conspicuously problematic.

Second, conventional IM applications do not require users to think about where they should place their messages, as new messages automatically appear at the bottom of the message stack by default. However, when messages are grouped into threads, users carry an extra cognitive load of justifying where to post their messages. This can be problematic when topics of a conversation are not clearly defined.

### 5.3.3 Avatar-based metaphor

In recent years, as personal computers have become more powerful and the connectivity of networks has increased, it has become feasible to include graphical content into IM. Several studies such as The Palace [222], Comic Chat [132], Chat Circles [246], CrystalChat [221], and Coterie [209] have adopted the avatar-based interface as an alternative metaphor to present a conversation. An appealing look-and-feel of the avatar-based interface enriches the context of online conversations. For instance, a person is now represented by an avatar instead of merely a text-based username and background images can be included to reflect the social culture of a conversation, as seen in Figure 5.10.

The Palace is a graphical, icon-based chat tool that provides a unique way of allowing users to portray their social identity and personality. In The Palace, users are represented by cartoon characters. Users can pick their favourite avatars that are meaningful to them as well as conveying their personalities to others. A conversation between users is visualised by text-based bubbles that appear next to users’ avatars as they speak. The Palace is a fun tool for social chat, but provides little support for maintaining the structure of a conversation. As a result, it is not practical for group discussion.
Figure 5.10: Avatar-based metaphor (a snapshot of The Palace [209]).

Comic Chat is another attractive graphical chat tool that creates a stylish visualisation of an IRC (Internet Relay Chat) conversation. Comic Chat adopts a comic book metaphor to lay out a chat conversation in the form of a story, which is very entertaining. Unfortunately, due to space constraints in Comic Chat, only a small portion of a conversation is shown. In addition, comic avatars and background images used in Comic Chat are loosely related to the conversation itself, which could be misleading.

Chat Circles represents users as coloured circles. A circle expands when a new message is posted, and becomes blurry after a period of idleness. Chat Circles is able to convey the level of turns in a conversation. In Chat Circles, messages are conversant-oriented in structure, thus there is no support for grouping related messages.

Coterie is another graphical interface of an IRC conversation that focuses on visualising users’ presence, conversational activity and the structure of a conversation. In Coterie, users are represented as coloured ovals which bounce and become brighter when they post messages. Coterie defines heuristics to analyse the relationships between messages, and then groups them into threads. However, messages in each thread still appear in sequential order. Therefore, Coterie has similar limitations to the sequential text-based model discussed in Section 5.3.1.
5.4 Design of awareness mechanisms

Section 5.2 analyses six major aspects of conversational awareness support in IM, including awareness of presence, awareness of turn-taking, awareness of conversational context, awareness of emotions, awareness of identities, and awareness of multiple, concurrent, one-to-one conversations. The analysis shows that there has been significant effort devoted to the study of awareness in IM, and many aspects of awareness need to be supported more adequately. Certainly, in order to support these diverse aspects of conversational awareness, many mechanisms need to be designed, implemented and evaluated.

This section describes two innovative prototypes, including RIM and ConDock. RIM facilitates a group conversation of a small group by enhancing presence awareness, turn-taking and conversational context. ConDock helps users manage awareness of multiple concurrent one-to-one conversations. Within the scope of this research, some major aspects of conversational awareness are addressed in the design of RIM and ConDock, while other aspects can be either found in existing IM applications or addressed as future work (see Section 5.7). Here, RIM and ConDock are introduced in terms of their design rationale and actual user interfaces.

5.4.1 Design of RIM

5.4.1.1 Design rationale

The design of RIM has evolved in part from F@ and in part from existing design metaphors.

First, F@ addresses various issues regarding support for conversational awareness. Awareness of presence, awareness of conversational context and awareness of turn-taking are three of the most important aspects of conversational awareness. The design objectives of RIM are, therefore, to enhance support for these three aspects. In particular,

1. RIM enhances awareness of presence by showing both current conversants and past conversants (discussed in Section 4.4.2);

2. RIM improves support for conversational context by providing a coherent layout of messages; and
(3) RIM enhances support for turn-taking by showing multiple textual “who is typing” cues, and detecting threads to which new messages are posted.

Second, the user interfaces of RIM are designed based on two design metaphors—sequential text-based and threaded text-based—which are described in Section 5.3.1 and Section 5.3.2, respectively. Venolia and Neustaedter [241] proposed mixed-model visualisation, which combines the sequential metaphor and the threaded metaphor to design a conversation-based email tool. Combining these two orthogonal metaphors is useful in addressing two different key issues. The sequential metaphor supports tracking of a chronological order of messages (for example, “Which of these two messages was sent first?” and “Which message was sent before this one?”). The threaded metaphor, on the other hand, helps users follow the discussion by providing the structure of the conversation (for example, “Which message replies to this one?” and “To which message does this one reply?”). Venolia and Neustaedter’s user test showed that this mixed-model visualisation is useful in assisting users in understanding both the sequence of messages and the tree of messages at a glance for an arbitrary conversation. The success of using mixed-model visualisation in designing interfaces for email has inspired the adoption of the mixed-model visualisation to design RIM.

5.4.1.2 User interfaces of RIM

The interface of RIM consists of five main panels: Tree Canvas, Message Canvas, Chat Area, Buddy List, and Topic List (Figure 5.11). Tree Canvas uses the threaded layout to form the structure of a group conversation. Message Canvas is designed to provide the focal point of a conversation using a linear sequence of messages. Chat Area includes a text field entry implementing a principle to locate automatically a new message in the tree. Buddy List shows information about conversants such as their names, avatars, activities, and how long they have been in, or had left, the conversation. Topic List shows a list of top-level topics of a conversation, and the number of child nodes under each topic.

Tree Canvas

Tree Canvas organises IM messages into a tree structure, similar to what was implemented in Threaded Chat [207]. Tree Canvas includes a list of top-level nodes (referred to as topics) and all sub-level nodes (referred to as replies) organised in a tree layout.
Figure 5.11: Relaxed Instant Messenger (RIM).

(A): Tree Canvas organises messages in a tree layout, and shows a textual “Who is typing” cue; (B): Message Canvas displays messages in a linear order; (C): Chat Area includes a text field which implements URP; (D): Buddy List displays multiple visual cues of “Who is typing”, and shows how long users have been logged onto the conversation; (E): Topic List shows a list of top-level threads of a conversation as well as the number of replies in each thread.

Tree Canvas allows users to post messages by interacting directly with a message tree. A user can compose a message by first clicking (that is, a double left-mouse click) on a tree node containing the message to which the user wants to respond, and then typing the content of the new message. The message is broadcast to other users once the ENTER key is pressed. The new message can be a new topic or a new reply. Tree Canvas also allows users to edit their own messages in the tree. The rationale inspiring
this design came from the results of the empirical study of IM users, from which it was found that many IM users wanted to be able to edit their messages after the messages were sent [232].

When a new message is being composed, or an existing message is being edited, a *textual* cue of “Who is typing” appears on the other users’ screen at the node where the new message is allocated (see Figure 5.11a). Tree Canvas also highlights the last message posted by the local user. This feature allows users to navigate through different parts of the tree, without the need to remember the location of a node at which they posted the last message.

To maintain the consistency of layout of a local user’s message tree, RIM does not expand the tree when there is a new message posted by a remote user. Although the tree does not expand, the user is still able to read a new message via Message Canvas, as described in the next section. In addition, Tree Canvas retains the content of the message so that whoever joins the group discussion late is still able to see the entire tree as other users see it.

**Message Canvas**

Message Canvas was developed to support the focal point which is missing from Tree Canvas. Message Canvas is a read-only view showing all messages in a linear order as seen in conventional IM and chat systems. Message Canvas keeps users informed of the arrival of new messages and who posted them. Message Canvas becomes significantly useful when a new message is out of a local user’s tree view, or a parent node containing the message is collapsed.

In addition, Message Canvas allows users to locate corresponding positions of messages in the tree. When users click on a particular message on Message Canvas, the message’s position in the tree is highlighted.

**Chat Area**

In conventional IM applications, the users mainly use a keyboard, and use the mouse infrequently, when posting messages. However, the current interaction with a tree (that is, clicking on a node to create a new message) is inconvenient and slow [207]. This is
because the procedure of clicking-and-typing requires users to switch continually between the mouse and keyboard.

Chat Area is an innovative way to speed up the pace of interaction caused by the clicking-and-typing procedure. Chat Area includes two main features:

1. a text field where users can compose messages as they normally do in conventional IM tools; and

2. the utterance rule-based principle (URP) that is used to detect automatically the node in the tree at which a message composed in the text field should be allocated.

URP is developed based on dialogue acts\(^{18}\) [61]. A conversation is a set of topics and turns. Each turn then consists of one or more utterances [195]. Knowledge about an utterance is, therefore, useful in detecting a topic to which the utterance belongs. According to the dialogue act study, an utterance can perform two functions: forward looking function (FLF) and backward looking function (BLF). FLF refers to an utterance that affects its following discourse. BLF refers to the link between an utterance and the previous discourse. Examples of utterances that have FLF include questions and comments; examples that have BLF are answers and agreements (for example, reject and accept). Take the following dialogue as an example:

\begin{quote}
User A: Should we use black for the icon?
User B: It is ok, but how about dark blue?
User C: I think either is fine.
\end{quote}

The first utterance has FLF as it affects the next two utterances. The second utterance has both BLF and FLF as it gives an answer, and also has an effect on the third utterance. The third utterance has BLF, as it responds to two previous utterances.

Based on the utterance analysis, and the author’s own experience of using RIM, it is recognised that a user’s next message would often be on the same topic as the user’s last message. When changing to a new topic, a user often posts more than one message on the new topic.

\(^{18}\) This thesis does not discuss different views on the definition of dialogue acts, and the analysis of the utterance hierarchy is complex and beyond the scope of this thesis.
Drawing on that analysis, URP is defined as:

“A user’s next message will be at the same level and under the same topic at the user’s last message”.

URP is simple but the impact that it has on RIM is significant. URP positions automatically a new message in a tree (that is, without clicking on a tree node), and consequently could increase the pace of IM conversations. In comparison to the method of clicking on a tree node to post a new message, by applying URP it is expected that conversations in RIM would be easier and more convenient to carry out. A hypothesis is that users will use Chat Area, which implements URP, to post their messages more often than interacting directly with Tree Canvas. This hypothesis is tested when the evaluation of RIM is reported.

**Buddy List**

Unlike conventional IM clients that provide only a list of active users in a group discussion, Buddy List of RIM shows both active users and inactive\(^\text{19}\) users. Buddy List also displays durations of how long active users have been in a discussion and the inactive users had left the discussion. In addition, Buddy List provides a visual cue of “Who are typing” to indicate that multiple users are composing or editing messages. As shown in Figure 5.11d, three users—Xeon, SkyLine and Overture—were composing new messages. Furthermore, Buddy List also allows users to display different avatars.

**Topic List**

In the development of RIM, it is realised that when the size of a conversation grows (for example, the number of branches and sub-levels increases), users could be unaware of what topics are about. To address this problem, Topic List is developed to show a list of main topics and the number of replies under each topic. Users can click on a topic in Topic List to jump to the corresponding location of that topic on Tree Canvas. This feature avoids scrolling up and down the entire tree to find a topic on Tree Canvas. In addition, Topic List also marks the topic which contains the last message posted by the local user.

\(^{19}\) The term “inactive” is used to refer to users who were in group chat but *had left* the conversation.
Coloured Messages

Colours have been widely used in designing user interfaces of many interactive systems, and can be very effective [65, 151]. This inspires the use of colours to code different information in an IM conversation, and to enhance display aesthetics. Colours are used in RIM for two main purposes:

1. coherently connecting messages on Message Canvas to those on Tree Canvas, and
2. facilitating turn-taking control.

First, colours are used in RIM to improve coherence between Tree Canvas and Message Canvas. RIM already supports a link between the two views by allowing users to click on a message on Message Canvas to see the message’s corresponding location on Tree Canvas. However, this interactive style could be limited as it requires the switch between the mouse and keyboard.

Therefore, this research works towards a solution of using colours as a mechanism to establish a coherent link between two views. As seen in Figure 5.11, each top-level topic on Tree Canvas is coloured in a unique colour\(^{20}\), and all child nodes of that topic inherit the same colour as the parent node. Each message on Message Canvas includes an icon that is coded with the same colour of the top-level topic into which the message is added. By doing so, this technique allows users to recognise promptly to which topic a new message belongs in the tree.

Second, when a user composes new messages via Chat Area, a coloured icon next to the text field informs the user of the destination of the message in the tree, even before the message is sent. RIM supports turn-taking by presenting a coloured visual “Who are typing” cue. Conventional IM clients, such as messengers of MSN and Yahoo, show only a visual cue of one user typing. However, the visual cue provided by RIM is able to show multiple users composing messages concurrently.

Moreover, RIM’s visual cue of “Who are typing” is also coloured in such a manner to inform remote users of the topic to which a currently composed message belongs. Thus, RIM’s visual cue shows both “Who are typing” and “Where new messages go”.

\(^{20}\) Currently, colours of topics are generated automatically and checked to ensure that colours are not repeated. Also, colours are only used to distinguish between different topics, and do not carry any meanings such as red denoting an active topic or grey meaning a less active topic.
5.4.2 Design of ConDock

5.4.2.1 Design rationale

Similar to the case of RIM, the design of ConDock has evolved from the principles of F@ and from existing awareness widgets.

First, current IM applications provide very limited support to help users manage the presence of multiple concurrent conversations. Consequently, when users are engaged in several conversations simultaneously, controlling multiple windows is problematic, as discussed in Section 3.2.2.6 and Section 5.2.6. Therefore, the main design objective of ConDock is to improve awareness support for multiple concurrent conversations. In particular, ConDock helps users manage and easily stay aware of arrival of new messages associated with different conversations.

Second, the design of the user interfaces of ConDock is inspired by the fisheye view awareness mechanism [82, 94]. A fisheye view is a focus+context visualisation technique that shows both global context and local details in a single display. Global context is shown in a miniature view, while local detail is magnified by a magnification lens. A fisheye view has been found useful supporting group awareness in cooperative work such as collaborative writing [93] and mediaspaces [53]. However, its potential has not been explored in the IM domain.

5.4.2.2 User interfaces of ConDock

ConDock adopts the fisheye technique to visualise the presence of multiple, concurrent one-to-one conversations. The interface of ConDock includes a single window that contains all conversations in which a user is currently engaged, in a miniature view, as seen in Figure 5.12a. Users can quickly read a conversation in ConDock by moving a mouse over the conversation. As a user moves the mouse over a particular conversation, the window of that conversation is enlarged while windows of other conversations in ConDock remain unchanged (Figure 5.12b). This creates a visual effect like fisheye views. If users want to type further in a particular conversation, they can drag the conversation out of ConDock and interact with the window as normal. When users minimise a conversation window, it is then placed back into ConDock instead of on the task bar.
ConDock also includes visual cues to provide awareness information about the arrival of new messages. When a new message arrives at a particular conversation in ConDock, the window containing that conversation is flashing, and a new message is highlighted in another colour. The window stops flashing and the colour changes to the default colour when a user attends to the conversation by moving a mouse over the window.

5.5 Implementation of awareness mechanisms

5.5.1 Implementation of RIM

RIM was written in the Java language, using the client-server architecture. The overall class diagram\(^\text{21}\) of RIM is illustrated in Figure 5.13.

A RIM server is non-graphical software that manages incoming connections from RIM clients, and receives messages from and passes information to the other clients. For example, when a user changes an avatar, the RIM client sends a message to the RIM client.

\(^{21}\)This class diagram only shows associations between classes that actually create a particular object in RIM.
server; the server then sends messages to the other RIM clients to inform the change of the user’s avatar.

A RIM client is a graphical application which is responsible for connecting to the RIM server, handling messages passed from the server, and displaying graphical interfaces (for example, Tree Canvas, Message Canvas, etc.) to end-users. Figure 5.14 presents sequence diagrams of two typical interactions between a RIM client and the RIM server, including “Send a message from Chat Area” (Figure 5.14a), and “Send a message from Tree Canvas” (Figure 5.14b).

Figure 5.13: Class diagram of RIM.
Chapter 5  Designing Awareness Support for Instant Messaging

(a) Sequence diagram of sending a message from Chat Area.

(b) Sequence diagram of sending a message from Tree Canvas.

Figure 5.14: Typical interactions between a RIM client and a RIM server.
5.5.2 Implementation of ConDock

ConDock was written in the C++ language and implemented as a plug-in of MSN Messenger using the transparent adaptation approach. The transparent adaptation approach utilises the MSN Messenger APIs (Application Programming Interface\textsuperscript{22}) to access and manipulate objects of MSN Messenger. This approach was proposed by Xia et al. [258] in the development of CoWord. This research adopts and enhances the approach specifically for the purpose of developing awareness mechanisms [237]. This research has applied the transparent adaptation approach to the implementation of ConDock (presented in this section) and the implementation of other awareness mechanisms for collaborative authoring (presented in Section 6.4).

The Messenger APIs provide a set of interfaces for objects and events (for example, Messenger, MessengerWindow, OnSignIn, OnSignOut, etc.), which conform to Component Object Model (COM) Automation. Figure 5.15 shows the system architecture of ConDock that involves two major processes: API Adaptation and Message Interception.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure5.15.png}
\caption{System architecture of ConDock.}
\end{figure}

- **API Adaptation**: ConDock interacts with MSN Messenger via its APIs. For the purpose of supporting awareness, it is important to know the evolving statuses of data objects as well as users’ interaction with the Messenger. Thus, the APIs are categorised into two groups: Object Interfaces (OIs) and Event Interfaces (EIs).

\textsuperscript{22} Application Programming Interface (API) consists of a set of commonly-used and well-defined interfaces that support an integration of different software components. An API is also known as a standard technique for developing software that hides module implementation details from consumers of those modules by separating interfaces from implementation [60].
OIs support ConDock in accessing and manipulating properties of data objects in MSN Messenger, whereas EIs allow ConDock to listen to events generated by users’ interaction with MSN Messenger.

- **Message Interception**: Apart from interacting with MSN Messenger, ConDock also intercepts messages (for example, data message, events, etc.) sent between the Operating System (OS) and MSN Messenger. This message interception is important as certainly not all events are supported by the Messenger EIs.

Figure 5.16 illustrates the author’s solution of adapting the MSN Messenger APIs for developing ConDock. The Messenger COM objects such as *Messenger, MessengerGroup, MessengerWindow*, etc. expose their interfaces that allow ConDock to access and manipulate the objects’ properties. The API Consumer (APIC) component of ConDock acts as an *ActiveX* client\(^{23}\) that accesses the Messenger COM objects using the *IDispatch* interface\(^{24}\) provided by the objects.

---

23 An *ActiveX* client is a software component that accesses other components’ features via the other components’ exposed interfaces.

24 The *IDispatch* interface includes functions that allows APIC to access the methods and properties of the MSN Messenger COM objects (for example, *Messenger, MessengerWindow*, etc.).
As shown in Figure 5.16, ConDock and MSN Messenger are two independent applications, and communicate with one another using the COM technology. The APIC component creates a list of references to object interfaces corresponding to a list of MSN Messenger COM objects. For example, the \texttt{MA\_Messenger} object in the APIC component maps onto the \texttt{Messenger} object in MSN Messenger. The term “\texttt{MA}” in front of each object’s name stands for “\textit{Messenger Adapted}”.

5.6 Evaluations of awareness mechanisms

5.6.1 Laboratory-based user evaluation of RIM

5.6.1.1 Goals of the evaluation

The design objective of RIM is to enhance conversational awareness in a conversation of a small group. A laboratory-based user test was conducted to evaluate the design of RIM. The evaluation focuses on four objectives:

- First, the evaluation provides information about how RIM is used by participants in supporting group conversation. This information is useful in broadening understanding of how conversational awareness needs to be supported.

- Second, the evaluation tests the design of RIM in terms of providing the focal point of a group conversation, and supporting posting messages to a message tree.

- Third, the evaluation is used to compare the effects of RIM and Gaim on the effectiveness of group discussion.

- Finally, the evaluation uncovers the strengths and weaknesses of the design of RIM. These results will be useful for further improvement of RIM.

5.6.1.2 Participants

Twenty-one participants, including twelve females and nine males, were recruited for the user test of RIM. The majority of them were in their early twenties. All of them had used IM applications, such as MSN Messenger, Yahoo Messenger and AOL’s Instant Messenger, for more than three months, and nearly 90% of them had used IM for more than two years.
5.6.1.3 Tasks
The participants were allocated to seven groups of three. Each group participated in a two-hour experimental session to perform two tasks. Each task was composed of two collaborative exercises.

- The first exercise was a “finding words” game. The rules of the game are that the next word must start with the letter which is the final letter of the previous word listed by another participant, and a word cannot be repeated. One scenario of this game was that each participant had to list 10 non-repeated nouns. For example, the first person started with the word “orange”, and the next person had to name a noun starting with “e” (for example, “electricity”). Three participants took turns to list proper nouns, and the exercise ended when each participant had listed 10 nouns.

- The second exercise was a “planning budget” project. One of the project scenarios was that three participants worked in a marketing department of a travel agency. They had to plan for a marketing campaign with the budget of $3000. The participants’ task was to chat with each other as a group and decide how they should spend the $3000, such as on the media of radio, television, local newspapers and magazines.

These two exercises were designed and chosen to ensure that participants needed to discuss matters as a group to complete the exercises. For example, in the first exercise, participants must know other participants’ messages and post corresponding valid answers. The second exercise required participants to discuss different alternatives (for example, pros and cons of each media) and make their final decision.

5.6.1.4 Experimental procedure
In each session, participants performed two tasks (that is, four exercises), using different IM software, different “finding words” games (for instance, finding nouns and finding verbs) and different “planning budget” scenarios. The procedure of each session is shown in Table 5.1.

---

25 Since RIM aims to support discussion of a small group, groups of three people were used in the evaluation.
Table 5.1: Experimental procedure of each session.

<table>
<thead>
<tr>
<th>Task</th>
<th>Exercise</th>
<th>IM tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each session</td>
<td>Exercise 1: Finding verbs</td>
<td>Gaim</td>
</tr>
<tr>
<td>Task 1</td>
<td>Exercise 2: Planning budget</td>
<td></td>
</tr>
<tr>
<td>Task 2</td>
<td>Exercise 1: Finding nouns</td>
<td>RIM</td>
</tr>
<tr>
<td></td>
<td>Exercise 2: Planning budget</td>
<td></td>
</tr>
</tbody>
</table>

In Task 1, Gaim was chosen as a conventional IM client. Gaim is an open source IM client implementing multiple messaging protocols of MSN, AOL, Yahoo, etc. The purpose of this task was to see how participants perform their exercises using a conventional IM client. In Task 2, participants used RIM to carry out their exercises (they were trained to use RIM before commencing the exercises). Three top-level nodes were pre-defined as a proposed structure of the discussion. The “Non-exercise related chat” node was used for social chat (that is, not related to their exercises) between participants, the “Finding words” node was for the first exercise, and the “Planning project” node was for the second exercise. Although the three topics were made available, participants could (and did) create more topics as appropriate for their discussion.

At the end of a session, participants filled in a 7-point Likert scale questionnaire\(^\text{26}\) (for example, “Coloured keyboard icons show multiple people typing” with the scores ranging from 1 as not useful to 7 as useful), and took part in a group interview. Group interviews were selected because not only do group interviews allow interacting directly with participants as in individual interviews, but they also allow the participants to react to and build upon the responses of other participants.

5.6.2 Results of RIM evaluation

This section reports the results of the evaluation of RIM. First, the structure of RIM conversations is examined and the usefulness of RIM in supporting awareness for group conversation is analysed. Then, a comparison of the usefulness of Gaim and RIM in supporting group discussion is reported.

\(^{26}\) The Likert scale questionnaire was developed based on Questionnaire for User Interface Satisfaction [42] and Purdue Usability Testing Questionnaire [137], and is included in Appendix B.1.
5.6.2.1 Structure of RIM conversations

Analysing the structure of RIM conversations is useful because it helps gain a deeper understanding of how a tree layout was used in organising messages. The analysis of a message tree examines three aspects:

1. the number of branches,
2. the centre, and
3. the depth of the tree.

First, the number of branches is measured based on the number of top-level topics. Second, the centre of a tree indicates at which level of the tree most messages are located. Hence, the centre of a tree is calculated based on the total number of messages at each level. Third, the depth of a tree shows how many levels are nested in the tree. The top-level node is defined as “level 0”. The level increases every time a child node is created. The results of analysing message trees from the test are summarised in Table 5.2.

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
<th>Std. dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of branches</td>
<td>3</td>
<td>8</td>
<td>5.33</td>
<td>1.63</td>
</tr>
<tr>
<td>Centre of a tree</td>
<td>0</td>
<td>4</td>
<td>1.23</td>
<td>0.02</td>
</tr>
<tr>
<td>Depth of a tree</td>
<td>1</td>
<td>3</td>
<td>2.17</td>
<td>0.75</td>
</tr>
</tbody>
</table>

Table 5.2: Structure of a message tree in RIM.

On average, participants created five topics in their discussion (mean = 5.33, std. dev. = 1.63). This number shows the diversity of matters discussed in group chat. The centre of a tree shows that most of the messages were posted at one level below the main topic (mean = 1.23, std. dev. = 0.02). This result indicates that in most cases messages are in a linear order, and in some cases participants created a sub-node to answer, question, and/or comment on another participant’s message. The depth of a message tree is around 2 (mean = 2.17, std. dev. = 0.75). That means, on average, participants used up to two levels below the top-level node in their discussion, which is very difficult to support in the conventional linear structure of messages.
5.6.2.2 Support for focal point

RIM was designed to provide the focal point of a conversation which is missing in standalone tree-based applications such as Threaded Chat in [207]. To support the focal point, RIM features Message Canvas that adopts a familiar layout of a linear sequence of messages to show the temporal order of coming messages. In addition, Message Canvas uses coloured icons to associate messages arriving at Message Canvas with topics to which messages belong.

Most participants found, to some extent, Message Canvas effective in helping them to be aware of the arrival of new messages. Table 5.3 shows participants’ feedback on the usefulness of Message Canvas.

<table>
<thead>
<tr>
<th>Questions</th>
<th>Mean (out of 7)</th>
<th>Std. dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keeping track of new messages on Message Canvas</td>
<td>5.00</td>
<td>1.87</td>
</tr>
<tr>
<td>Using coloured icons to associate messages on Message Canvas with topics</td>
<td>5.00</td>
<td>1.85</td>
</tr>
<tr>
<td>Mapping messages between Message Canvas and Tree Canvas</td>
<td>4.11</td>
<td>1.83</td>
</tr>
</tbody>
</table>

Table 5.3: Message Canvas supports a focal point.

A coloured icon in front of each message is designed to inform users of the topic containing that message. However, an observation from the user test shows that participants often discussed one topic at a time and only occasionally posted messages in another topic. As a result, participants mainly used a coloured icon as an indicator when there is a change in a conversation (that is, a new message does not belong to the currently discussed topic). One participant commented on how he used coloured icons: “With colours, I could tell that is where we were up to, without having to think about oh! It was not related. You could see the difference. But I wasn’t tracking the colour specifically to the topic, just a matter of the difference”.

5.6.2.3 Ease of message posting

URP is implemented in Chat Area to ease the procedure of posting messages in RIM by automatically allocating to Tree Canvas messages composed in Chat Area. The following hypothesis, as proposed earlier, is tested to evaluate the effectiveness of URP:

\[ H: \text{“Users use Chat Area to compose messages more than interacting directly with Tree Canvas”} \].
Figure 5.17 shows means of the number of times that participants composed messages using Tree Canvas and Chat Area in Task 1 and Task 2. To justify whether Chat Area was used significantly more often than Tree Canvas, the one-way repeated measures ANOVA test \(^{27}\) was conducted.

![Figure 5.17: Mean of the numbers of times that participants used Tree Canvas and Chat Area to compose messages.](image)

As seen in Table 5.4, the ANOVA test shows that \(p\)-values in both Task 1 and Task 2 are less than 0.05. Therefore, there is significant evidence to conclude that Chat Area was used more often than Tree Canvas.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Chat Area</th>
<th>Tree Canvas</th>
<th>Test of significance, with (\alpha = 0.05)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>14.05 (3.01)</td>
<td>1.76 (1.76)</td>
<td>(p)-value &lt; 0.05</td>
</tr>
<tr>
<td>Task 2</td>
<td>29.05 (13.68)</td>
<td>4.95 (5.26)</td>
<td>(p)-value &lt; 0.05</td>
</tr>
</tbody>
</table>

Table 5.4: ANOVA test of the number of messages posted via Chat Area and Tree Canvas.

The logged data shows that participants often clicked on Tree Canvas when they posted direct questions and answers to other participant’s messages, and when they changed to another topic. As Chat Area was used by participants as a primary method of posting messages, the evaluation clearly shows that URP has been useful in automatically detecting the position of a new message in the tree.

To verify the correctness of URP, this research measures the percentage of errors occurring in RIM conversations (referred to as “thread errors”). This research considers a thread error as an occurrence when a message is composed using Chat Area and placed at a tree node that was not the one intended (for example, a message was posted

\(^{27}\) The same sample across Task 1 and Task 2 was examined, thus, the repeated measures ANOVA procedure is used \([37]\). And, a standard level of significant used is 5% \((\alpha = 0.05)\).
in an incorrect topic or at a wrong level). Figure 5.18 shows error rates that occurred in RIM. On average, error rates are only around 4% and 5% in Task 1 and Task 2, respectively.

The logged data shows that these thread errors occurred when users wanted to change threads. In such situations, users were supposed to specify new threads explicitly (for example, by double clicking on Tree Canvas), however, they still used Chat Area to post messages. Consequently, new messages were located in their current threads instead of new threads that users intended. When thread errors occurred, users had either ignored them and posted the next message in the right threads, or actually corrected the errors by creating duplicates of the misplaced messages at the actual targeted threads.

5.6.2.4 Comparison of Gaim with RIM

Whilst Gaim is a conventional IM tool that displays messages of group discussion in a linear layout, RIM organises messages in both a tree layout and a linear layout. A comparison between Gaim and RIM is examined in two aspects: quality of discussion and user satisfaction.

Quality of discussion

The measure of the quality of discussion is based on the number of turns occurring in the discussion, and participants’ feedback. The logged data shows that on average, a Gaim conversation included 220.50 turns (std. dev. = 22.2), while a RIM conversation had 149.43 turns (std. dev. = 20.22), which was found to be significantly different: \( t(6) = 12.07, p\)-value < 0.05. Hence, it is concluded that a Gaim conversation took more turns than a RIM conversation. This finding is consistent with the study of a standalone
tree-based chat tool conducted by Smith et al. [207]. That means, participants were able
to complete the tasks, but fewer turns were required when RIM was used.

Participants acknowledged the usefulness of RIM in structuring their discussion (*mean*
\[= 5.86; \textit{std. dev.} = 1.17\]). RIM was found highly useful in organising the content of a
conversation in topics. RIM helps participants track different posts of the same topic
quickly and easily, as commented by participants:

“A very cool idea, topics are organised and can be followed easily”,

“Classify topics of discussion is a good idea”,

“Unique in its ability to create a real-time message board”,

“I like it. When I click on a topic, I can see all responses of other people. I don’t
have to look somewhere else because all messages are grouped there”, and

“I found RIM useful in structuring the discussion. It makes the discussion more
systematic and easier to keep track.”

In comparison to Gaim, RIM was found advantageous in allowing participants to post
answers to questions in a conversation conveniently (*mean* \[= 6.33; \textit{std. dev.} = 0.71\]).
Even in the case when a question had been posted a reasonably long time ago,
participants still could post a direct answer and/or comment to the question by creating
a child node under the question. Participants stated:

“In RIM, I can answer the specific question”, and

“Quick question and answer posting.”

Figure 5.19 illustrates the scenario of how a linear style is limited in conveying the
structural order of messages, and how a tree style can support that order. In Figure
5.19a, David’s message of “I agree with this” could be understood either as David agrees
with Melissa or David agrees with Lap. Whereas in Figure 5.19b, it is clear that David
agrees with Melissa as his message was posted as a child node that directly links to
Melissa’s message.
(a) Messages are structured in a linear layout

(b) Messages are structured in a tree layout

Figure 5.19: Structural order of messages.

Another useful feature of RIM, which is not supported by conventional IM clients such as Gaim, is visual indicators that indicate many people are typing at the same time and to which topic currently composed messages belong. Almost all of the participants found this feature useful (mean = 5.89; std. dev. = 1.17). As commented by one participant, “The icons [showing who are typing] are very handy. Often, if I saw the other people were typing in the same topic as mine, I stop typing and waited for them to finish first.” The comment indicates that such a visual cue showing “Who are typing” and “Where new messages go” facilitates turn-taking control in a conversation.

User satisfaction

User satisfaction was measured based on participants’ responses to post-experiment questions. The results show that on the one hand, RIM was more preferred than Gaim as a discussion tool; on the other hand, participants were less satisfied with the UIs of RIM than with those of Gaim. The main reason is because many basic features have not yet been implemented in RIM (for example, a word wrap feature is missing, no emoticons, fixed fonts and fixed text colours). This lack of features affected participants’ experience, as commented by one participant “Your program is quite handy for discussion. For example, the tree saves our group time when we reviewed our discussion. But personally I still find Gaim more fun because it has lot of smileys and nicer colours”. However, these UI features can be enhanced easily by further development of RIM as the primary focus of the current design of RIM is on new features.
5.6.3 Discussion of RIM evaluation

The evaluation of RIM indicates the potential of combining a tree structure and a linear structure in improving awareness support for group chat. Here, shortcomings of the current design of RIM as well as possible improvements are considered, and participants’ adaptation to the interaction style of RIM is examined.

5.6.3.1 Design issues

Drawing on the evaluation of RIM, several design issues need to be considered in order to improve the usability of RIM: multiple focal points for Tree Canvas, a tailorable threaded layout, and support for turn-taking.

Multiple focal points for Tree Canvas

In the current design of RIM, the focal point of conversation is supported by Message Canvas. However, the user study of RIM shows that it is difficult for those who paid great attention to Tree Canvas to follow the conversation, as commented by participants:

“I didn’t look at Message Canvas much because I often looked at the tree. It is hard to navigate two windows [Message Canvas and Tree Canvas] at the same time”, and

“It is uneasy to see the update in each topic even though there is Message Canvas window.”

Although this is related to the adaptation of using RIM, it seems useful to display multiple focal points within Tree Canvas. These focal points show the latest messages posted by users. Unfortunately, it is difficult to show the latest messages posted by all users in a user’s viewport because new messages can be posted at any node of a tree. For instance, one message could be at the first thread while another message could be at the last thread, and the two threads are not necessarily visible simultaneously in a user’s viewport.

Fish-eye views [82, 93] can be used to address the challenge of displaying multiple new messages in Tree Canvas. Fisheye views allow display of both the global view and detailed view on a single surface. Figure 5.20 shows the mock-up of fish-eye views integrated in Tree Canvas. Fish-eye views render the entire Tree Canvas in a single
window as well as showing both a local user—Minnie—and remote users’—MeoMap and Neo—focal points regardless of the locations of their foci. The focal points of remote users are adjusted automatically to show the latest messages posted by remote users. In addition, the users’ avatars and nicknames are also displayed to ease the identification of the focal points.

![Figure 5.20: Tree Canvas with fish-eye views (mock up).](image)

**Tailorable threaded layout**

As reported in Section 5.6.2.3, in some cases messages composed via Chat Area were misplaced in wrong threads. Observing participants’ behaviour in the study, the author realised that even when messages are located at an appropriate threads participants feel a need for being able to re-locate these messages to another thread if required. For example, in the budget planning tasks, participants often compared one media with another. Messages referring to these comparisons are spread over many threads. At the end of the task, participants wished to re-arrange these messages in such a way so that all posted comparisons are grouped under one single thread. By doing so, it would be easy for a group to review the discussion and thereby make their final decision.
To satisfy this need, Tree Canvas should allow users to tailor the layout of the message tree by supporting a drag-and-drop feature. That is, users are able to re-position nodes of the tree by dragging the nodes and dropping them at particular threads.

**Support for turn-taking**

RIM supports the control of turn-taking of a conversation by adding a visual cue showing “Who are typing” and “Where new messages go”. This feature was found effective and useful by participants. However, the current cue does not handle the case when users stop typing while they are composing messages. This is also a problem in conventional IM applications.

For example, when a user is typing a message, and for some reason the user stops typing before completing the message. The visual cue of “who are typing” is still on and showing that the user is typing. When other users see that visual cue appear for a long period of time, it gives them the impression that the user is composing a long message, which can be wrong. This phenomenon is different from idleness, as in this situation the user can still be active. Three participants believed that this problem can affect the sequence of turn-taking.

To address this problem, participants suggested that RIM could show a timeout if users stop typing for so long when composing a message. Figure 5.21 depicts the mock-up of a solution proposed by one participant showing a clock-like icon next to the keyboard icon. The clock icon is changing from full to blank after a period of time to indicate how long a user has stopped typing.

![Figure 5.21: A clock-like visual cue (mock-up).](image)

**5.6.3.2 Participants’ adaptation to RIM**

The evaluation also investigated how participants adapted to RIM by using 7-point Likert scale questions about learnability, as shown in Table 5.5. The results show that participants did not experience major issues in learning to use RIM. The design goal of
RIM is to apply familiar metaphors. For instance, Message Canvas is similar to a conventional chat window and Tree Canvas is similar to threads of a discussion forum application. However, the evaluation showed that participants’ adaptation to RIM is still fairly slow. In some cases, although there were pre-defined topics, some participants still used one topic for discussing two exercises at the same time.

<table>
<thead>
<tr>
<th>Questions</th>
<th>Mean (out of 7)</th>
<th>Std. dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning to use RIM</td>
<td>4.76</td>
<td>1.61</td>
</tr>
<tr>
<td>Remembering interaction styles of RIM</td>
<td>4.33</td>
<td>1.71</td>
</tr>
<tr>
<td>Remembering the meaning of different icons</td>
<td>4.70</td>
<td>1.13</td>
</tr>
<tr>
<td>Associating colours with topics of a conversation</td>
<td>3.76</td>
<td>1.38</td>
</tr>
</tbody>
</table>

Table 5.5: Participants’ learnability.

The study also found that some participants who had used IM for a long time do not necessarily adapt rapidly to RIM. What became important was participants’ previous experience with group conversation using conventional IM applications. The test shows that participants who had used conventional IM applications for group discussion greatly appreciate the structural conversation built by RIM.

### 5.6.4 Field trial of ConDock

#### 5.6.4.1 Goals of the evaluation

A field trial was conducted to evaluate the design of ConDock in improving users’ awareness of multiple concurrent conversations. ConDock is implemented as a plug-in for MSN Messenger, a very popular IM application that has been used by millions of real-world users. Thus, selecting a field trial to evaluate ConDock is appropriate and advantageous because it allows real-world users to use ConDock to support real-world conversations in a real-world setting.

The field trial of ConDock focuses on three objectives. The first objective of the evaluation is to test the usefulness of ConDock in helping users manage and stay aware of multiple concurrent one-to-one conversations. Second, the field trial is also used to extend understanding of users’ handling of multiple, concurrent, one-to-one conversations. Finally, through the trial, the strengths and weaknesses of the design and implementation of ConDock are uncovered. This becomes valuable for further improvement of ConDock.
5.6.4.2 Participants
Eight participants, including five males and three females, were recruited for the field trial of ConDock. The participants were university students in their twenties. All of them had used MSN Messenger for more than one year on a daily basis.

5.6.4.3 Experimental procedure
Participants used ConDock for a period of three weeks at home. They were asked to run ConDock whenever they engaged in more than one conversation with their buddies. At the end of each week, participants were asked to complete a 7-point Likert scale questionnaire. At the end of the third week, informal interviews were conducted to ask participants about their experience with ConDock as well as perceiving their comments on how ConDock could be improved.

During the three-week trial, each participant was asked to keep a digital diary which is a Microsoft Word document that recorded their positive and negative comments on ConDock. The diary also contains screenshots of ConDock to illustrate scenarios in which ConDock was found or not found useful in supporting participants’ conversations.

5.6.5 Results of ConDock evaluation
This section reports the results of a field trial of ConDock in relation to two aspects:

(1) the usefulness of ConDock in helping participants manage multiple conversations, and

(2) how ConDock was used by participants.

5.6.5.1 ConDock supports managing multiple conversations
The usefulness of ConDock in aiding users to manage multiple, concurrent conversations is evaluated based on participants’ responses to the 7-point Likert scale questionnaire and their comments in the interviews. Overall, ConDock is found useful in helping users manage their conversations \((mean = 4.87; std. dev. = 1.33; n = 24)\). In addition, it is useful to know how participants’ responses have changed over three weeks of trial. The results of participants’ weekly and overall responses are depicted by the boxplots in Figure 5.22. The boxplots illustrate a change of participants’ opinions on

---

28 The questionnaire is included in Appendix B.2.
the usefulness of ConDock over three weeks, with the highest rating belonging to the third week. The boxplots also indicate that the median rating (that is, the dark line in the middle of the box) increases slightly after each week.

![Boxplots showing weekly ratings of ConDock usefulness](image)

**Figure 5.22:** Weekly measure of participants’ responses to usefulness of ConDock in managing multiple conversations.

In order to justify if there is any difference among the mean rating of the three weeks, the *one-way repeated measures* \(^{29}\) ANOVA test [37] was performed. The null and alternative hypotheses would be:

\[
H_0: \text{mean}_1 = \text{mean}_2 = \text{mean}_3
\]

\[H_1: \text{there is difference among mean}_1, \text{mean}_2 \text{ and mean}_3\]

The repeated measures ANOVA test shows a \(p\)-value of 0.685 which is greater than 0.05 (Table 5.6). Therefore, the null hypothesis is not rejected and the test concludes that participants’ rating *change* significantly after three weeks [37].

<table>
<thead>
<tr>
<th>Within Subjects Effect</th>
<th>Mauchly's W</th>
<th>Approx. Chi-Square</th>
<th>df</th>
<th>Sig.</th>
<th>Epsilon(^{(a)})</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Greenhouse-Geisser</td>
</tr>
<tr>
<td>WEEK</td>
<td>.882</td>
<td>.756</td>
<td>2</td>
<td>.685</td>
<td>.894</td>
</tr>
</tbody>
</table>

**Table 5.6: Repeated measures ANOVA: Mauchly’s Test of Sphericity.**

\(^{29}\) Again, the *same* sample across three weeks was examined, thus, the repeated measures ANOVA procedure was used [37].
In general, participants’ feedback shows that placing all IM conversations within a single window frame and allowing an accessible way of reading new messages is the most valuable feature of ConDock. As commented by participants:

“I like how conversations are stacked at one place. And it is fairly easy to move from one conversation to another”, and

“It is convenient that I just need to move the mouse over ConDock to read messages ... without having to switch continually between windows.”

5.6.5.2 Interaction style in ConDock

In the 7-point Likert scale questionnaire, participants were also asked to rate the design and implementation of fisheye views and window drag in ConDock. First, regarding a fisheye view, there are contradictory streams of participants’ feedback on the adoption of a fisheye view and on the actual implementation of a fisheye view in ConDock. On the one hand, participants like the interesting concept of a fisheye view (mean = 4.96; std. dev. = 1.04; n = 24). On the other hand, they were not satisfied with the current implementation of fisheye views in ConDock (mean = 2.79; std. dev. = 1.10; n = 24). The main reason for these opposing responses is because the fisheye view currently implemented in ConDock is too sensitive to mouse movement and flickers as the focal point of the fisheye view changes, as addressed in the discussion in Section 5.6.6.3.

5.6.6 Discussion of ConDock evaluation

The field trial of ConDock shows that the design of ConDock has been well received by the participants, as they found ConDock useful in helping them manage conversations easily. However, the evaluation of ConDock also indicates that the current implementation of ConDock is limited in some ways, and ConDock still has room for further improvement. This section discusses limitations to the current implementation of ConDock, and examines how participants had used ConDock to deal with the issue of multiple concurrent one-to-one conversations.

5.6.6.1 Always-on-top feature

Two participants suggested that ConDock should include an “always-on-top” feature. This feature allows ConDock to float on top of other windows like browsers, media players, etc., and therefore helps users stay updated with what is shown in ConDock. At
the moment, users need to activate the ConDock window first in order to access windows containing their ongoing conversations.

5.6.6.2 Lack of support for graphical content
Current design of ConDock does not show graphical backgrounds of conversations, and it also does not display all emoticons available in MSN Messenger. This lack of support for graphical content also affects participants’ satisfaction with ConDock. Furthermore, one participant suggested that ConDock could display users’ avatars to ease the identification of conversations. The illustration in Figure 5.23 shows an example of incorporating avatars and user names to enhance conversation identification in ConDock.

![Figure 5.23: Avatars and user names enhance conversation identification of ConDock (mock up).](image)

5.6.6.3 Limitations of a fisheye view
As reported in Section 5.6.5.2, the concept of a fisheye view in ConDock is acknowledged positively by participants. Unfortunately, the actual implementation of the fisheye view in ConDock should be improved in two aspects.

First, five out of eight participants believed that the fisheye view in ConDock is too mouse-sensitive. Often, when the mouse is placed outside a currently active conversation, the fisheye view loses its focus immediately, and the conversation consequently returns to a miniature state. To address this issue, ConDock can include a
timer that can be used to set an offset duration before a zoom of the fisheye view is activated.

Second, when a focal point of the fisheye view changes, a window of ConDock flickers and therefore causes an unpleasant user experience. The current implementation of ConDock uses a simple method of redrawing a conversation window as the size of the window is changed. That is, a window is redrawn many times to create a visual zooming effect. To overcome the flickering problem, more sophisticated programming techniques using graphic libraries such as DirectX [154] or OpenGL [173] can be utilised.

5.6.6.4 Participants’ experience with multiple conversations
The main design objective of ConDock is to help users manage multiple conversations by avoiding manually aligning different conversation windows. The field trial of ConDock showed two trends of how participants dealt with multiple conversations.

First, if participants chatted with their buddies while they were doing other tasks such as surfing the Internet or doing their homework, then ConDock was found useful because ConDock organises their conversations at one place, and provides support for presence awareness of new messages.

However, in some specific situation when chatting with other people is the primary task of users, then ConDock was found less useful. This is because in such a situation participants often arranged all conversation windows over the entire screen; therefore, ConDock was used less frequently.

5.7 General discussion
RIM and ConDock were designed to support awareness for group conversation (for example, presence awareness, turn-taking and awareness of conversational context) and awareness of multiple conversations. The designs proposed here aim to support other aspects of awareness, including viewing- and listening-in-progress and multiple identities, which are discussed in Section 5.2.1. These designs have not been implemented. Thus, mock-ups are presented to illustrate the design concept and general functionality. These designs result partially from participants’ paper-based sketches that
were transferred to high-fidelity design, and partially from experience gained from developing RIM and ConDock.

5.7.1 Awareness widget for viewing- and listening-in-progress

5.7.1.1 Track View

Track View is an awareness widget that is designed to inform users of who are currently *listening* to their auditory conversation, and who are currently *viewing* their Webcam [233]. Figure 5.24 illustrates a scenario where three users—Neo, Oai, and Jane—are currently viewing a local user's Webcam and two users—Kate and Neo—are currently listening to the local user's auditory track. In addition, Track View allows the local user to stop any current viewer from listening to their voice or seeing them through Webcam, for example, by unselecting a check box in Figure 5.24.

![Figure 5.24: Track View supports awareness of viewing- and listening-in-progress.](image)

At the moment, information about listeners and viewers shown in Track View is based on the request-and-accept protocol. For example, whenever a request to view Webcam is granted, Track View adds a user whose request is accepted, into a viewer list. But, Track View is not able to track if that user actually sees the Webcam (for example, the Webcam window might be covered by another window on the user’s PC). Similarly, Track View is not able to detect if listeners’ speakers are muted.

5.7.1.2 Buddy List

Buddy List was designed to enhance the current buddy list by providing visual cues to inform users if their buddies are capable of joining audio and video conversations [233]. Those visual cues are icons displayed next to contacts on the buddy list. If a buddy can
join audio and video conferences, a headphone icon and a camera icon are shown accordingly (for example, in Figure 5.25, John, Neo and Jane can join audio chat; Neo and Jane can join video chat). If a buddy is currently participating in audio and video conversations, the corresponding icons become bright (for example, Neo is involved in both audio and video chats).

![Buddy List](image)

**Figure 5.25: Buddy List.**

### 5.7.2 Awareness widgets for multiple identities

To support multiple identities in IM, this thesis proposes a design model that can support the one-to-many relationship between a user’s *username* and their many identities [231]. As shown in the class diagram in Figure 5.26, the model of multiple identities supports one-to-many (1 – 1...*) and many-to-many (1...* – 1...*) associations between different IM components. For example,

- associations between *Username* and *Avatar*, and between *Username* and *Status* are one-to-many relationships. Thus, each username can use many avatars and edit many online statuses;

- associations between *Group* and *Avatar*, and between *Group* and *Status* are many-to-many relationships. This means that each group in a user’s buddy list can use many avatars and each avatar can be used by different groups in the buddy list;

- associations between *Avatar* and *Nickname*, and between *Status* and *Nickname* are one-to-many relationships. Thus, only one avatar and one status can be set
for each nickname (that is, a person appearing in a buddy list) at a time. However, any avatar and status can be shared among many nicknames.

Figure 5.26: Class diagram showing the framework of multiple identity support.

Drawing on the model above, the author worked together with participants of the interviews to design low-fidelity interfaces using a paper-prototyping technique. The purpose of a paper-based prototype is to sketch out ideas and possible user interfaces to support multiple identities. After the participants were happy with an overall layout in a paper prototype, the prototype was transformed into higher fidelity design.

The high-fidelity design shown in Figure 5.27 supports multiple avatars and multiple statuses by allowing a local user to set a different avatar and a different status for each conversation. For example, a local user—Minh—has options to change another avatar and another status. When a change is made, the new avatar and new status are merely applied to a current conversation (that is, conversation between Minh and DarkAngel) by default. However, if Minh wants, he can also set a new avatar and new status as global; and they are automatically applied to all conversations between Minh and people other than DarkAngel. In a situation when a global avatar and a global status are used, this is similar to what is being implemented in current IM applications. However, in more general situations, Minh can display more than one avatar and edit more than one status [231].
Figure 5.27: High-fidelity prototype supporting multiple identities.

The mock-up screenshot in Figure 5.27 is the result of very early work in designing support for multiple identities, thus it is not complete in any way. However, the mock-up does show essential features, including support for multiple statues and multiple avatars, which are lacking in current IM applications. And more importantly, it illustrates how the proposed model can be interpreted in user interface design.

5.8 Reflection on the use of F@

The past few sections have presented details of the design, implementation and evaluation of two novel IM prototypes—RIM and ConDock—as well as the proposed design of some other mechanisms. This section steps back from the detailed descriptions, and presents higher-level reflection that can be drawn from the development of the two prototypes and the design of proposed mechanisms. In particular, the reflection focuses on a comparative analysis of relations between the principles of F@ and the designs of RIM, ConDock and proposed mechanisms.
### 5.8.1 Reflection on the abstract level

Based on Table 4.4 shown at the abstract level, Table 5.7 shows a list of awareness elements that are addressed in F@ and supported by RIM, ConDock and/or proposed mechanisms. The abstract level shows a comprehensive set of awareness elements and specific questions to which each element corresponds. The designs of RIM, ConDock and proposed mechanisms have addressed some of those awareness elements.

As mentioned in Section 4.3.1, not every aspect of the 5W1H dimensions is required to be supported by every groupware system because the need of providing conversational awareness depends on the group situation and varies according to the group context. Thus, mechanisms presented in this chapter are not designed to support all aspects of conversational awareness. As seen in Table 5.7, many, though not all, awareness elements raised by F@ are supported by awareness mechanisms of this chapter. The evaluations of RIM and ConDock confirm that awareness elements provided by the two prototypes enhance conversational awareness. This confirmation indicates that awareness elements presented at the abstract level (Section 4.3.1) are useful and can be applied to designing other mechanisms to support conversational awareness.

<table>
<thead>
<tr>
<th>Awareness elements (past/ current)</th>
<th>Design of RIM, ConDock and proposed mechanisms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presence</td>
<td>- RIM shows active users who are currently participating in a conversation</td>
</tr>
<tr>
<td></td>
<td>- Buddy List is proposed to show if users are able to join audio and/or video conversations</td>
</tr>
<tr>
<td></td>
<td>- RIM shows inactive users who are no longer in a conversation</td>
</tr>
<tr>
<td>Identity (current)</td>
<td>- RIM and ConDock display username, colour and avatar</td>
</tr>
<tr>
<td></td>
<td>- RIM shows multiple “who is typing” cues</td>
</tr>
<tr>
<td></td>
<td>- The thesis proposes a design to support multiple concurrent identities</td>
</tr>
</tbody>
</table>

The “Who” dimension

<table>
<thead>
<tr>
<th>Awareness elements (current)</th>
<th>Design of RIM, ConDock and proposed mechanisms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context</td>
<td>- RIM uses a tree to create a structured layout of messages</td>
</tr>
<tr>
<td></td>
<td>- RIM allows users to post questions or answers explicitly to a particular message</td>
</tr>
<tr>
<td></td>
<td>- RIM uses the same colour for messages of the same topic</td>
</tr>
<tr>
<td></td>
<td>- Track View is proposed to support listening- and viewing-in-progress</td>
</tr>
<tr>
<td></td>
<td>- ConDock displays multiple conversations in a miniature view, and magnifies a particular conversation when the mouse moves the conversation</td>
</tr>
</tbody>
</table>

The “What” dimension
Table 5.7: F@ and mechanisms for conversational awareness.

<table>
<thead>
<tr>
<th>Awareness elements</th>
<th>Design of RIM, ConDock and proposed mechanisms</th>
</tr>
</thead>
</table>
| **Motivation**     | - Users send messages to ask questions explicitly  
| (past/ current)    | - RIM supports a logical and structural layout of message that allows users to indicate the purpose of sending a particular message (for example, to answer or to question, etc.) |
| **Behaviour**      | - Users send messages to ask questions explicitly |
| (past/ current)    | |
| **Location**       | - RIM shows the latest message at the bottom of Message Canvas, while the message can be allocated at any node in Tree Canvas  
| (current)          | - ConDock indicates when new messages arrived at a particular conversation |
| **Event**          | - RIM shows elapsed time since users joined or left a conversation  
| (past/ current)    | - RIM indicates if anyone is typing a message  
|                    | - RIM indicates if anyone joins or leaves a conversation  
|                    | - Clock-like cue is proposed to show how long a user has stopped typing while composing a message |
| **Reaction**       | - Expressed via emoticons  
| (past/ current/ future) | - Users send messages to ask questions explicitly |

The “Why” dimension

The “Where” dimension

The “When” dimension

The “How” dimension

5.8.2 Reflection on the concrete level

As shown in Section 4.4.2, the concrete level formalises two aspects of conversational awareness—presence awareness of conversants and awareness of turn-taking. This section considers how those temporal logic formulas are interpreted and applied to the design of the mechanisms introduced in this chapter.

Presence awareness of conversants

Presence awareness of conversants involves information about the presence of past, current and future conversants. Past conversants are people who were in a conversation and had already left; current conversants refer to people who are currently participating in a conversation; and future conversants are those who might join a conversation. As
presented at the concrete level (Section 4.4.2), the presence aspect of conversational awareness is formulated as follows:

\[ M(s, \mathcal{I}) \models \text{aware}(\text{user}, \text{presence-conversants}) \iff \]
\[ (M(s, \mathcal{I}) \models (\forall p; p \neq \text{user}) \bowtie \text{part_of}(p, \text{Conversation}) \land \text{know}(\text{user}, p)) \land \]
\[ (M(s, \mathcal{I}) \models (\forall p; p \neq \text{user}) \text{part_of}(p, \text{Conversation}) \land \text{know}(\text{user}, p)) \land \]
\[ (M(s, \mathcal{I}) \models (\forall p; p \neq \text{user}) \Diamond \text{part_of}(p, \text{Conversation}) \land \text{know}(\text{user}, p)), \]

where \( p_i, p_j, p_k, \text{user} \in \text{Conversation} \).

RIM enhances support for presence awareness of conversants by showing both past and current conversants. As seen in Figure 5.28, RIM provides a list of current conversants—Buz, Karl, Pete—and past conversants—Minh. In addition, RIM displays the durations of how long current conversants have been participating in a conversation, and how long past conversants have left the conversation.

![Buddy View](image)

**Figure 5.28:** RIM supports presence awareness of past and current conversants.

Although support for presence awareness of future conversants is not provided in the current design of RIM, it can be easily extended. RIM can use an invite-and-accept protocol to identify who are going to join a conversation (that is, future conversants) and possibly anticipate when they would join a conversation.

**Awareness of turn-taking**

In Section 4.4.2, the concrete level presents three conditions that need to hold when a sender sends a message to a receiver in a conversation.

\[ M(s, \mathcal{I}) \models \text{aware}(\text{sender, send}(\text{sender, receiver, message}) \iff \]
\[ M(s, \mathcal{I}) \models \text{send}(\text{sender, receiver, message}) \] (CA1)
In face-to-face conversations, people are naturally aware of to whom they are talking and whether the listeners can hear them through verbal and non-verbal cues. However, these three conditions often do not hold in computer mediated communication tools such as IM applications.

In general, IM meets (CA1) and (CA3) by providing visual cues such as “Who is typing” in the case of text-based conversation, and a coloured bar that raises in the case of audio conversation to inform a sender and a receiver that the sender is sending a message. But, that is often only supported for the case of one-to-one conversations. When a conversation involves a group of more than two people, these two conditions often do not hold. For example, when two people are typing at the same time, IM does not indicate who they are; or when two people are talking at the same time, there is no visual indicator showing who is talking.

RIM was designed to support CA1 and CA3 by showing people who are concurrently typing in a conversation. Figure 5.29 shows the scenario of three users—Pete, Jane and Kate—composing messages simultaneously.

Current IM applications fail to meet (CA2). That is, a sender does not know whether a receiver actually receives the message. Hence, in many cases, a sender needs to ask the
receiver explicitly for confirmation (that is, if the receiver actually receives the message). To address this issue, IM applications can distinguish the cases when a message is delivered to a receiver successfully and when a message does not reach an intended receiver. IM applications can provide awareness mechanisms such as a message pool that keeps all failed-to-deliver messages, and allows a sender to choose if the sender wants to re-send or simply ignore those messages.

The issue of providing support for (CA2) is even more problematic in the case of audio and video communication. IM users are provided with no awareness cues informing them if receivers attend to their broadcasted audio and video contents. IM applications can include awareness mechanisms such as Track View (Figure 5.24) that are used to inform a local user of who else is currently listening to the user’s auditory track and who else is currently viewing the user’s Webcam. The author is not suggesting that Track View is necessarily the best way to resolve the problems addressed in (CA2), but rather show an example of how F@ can be interpreted in the design of awareness mechanisms.

5.9 Summary
This chapter presents a case study of designing and building mechanisms to support conversational awareness in Instant Messaging (IM). It starts with an analysis of current support for conversational awareness in IM. The analysis shows six major aspects of conversational awareness that need to be supported—awareness of presence, awareness of turn-taking, awareness of conversational context, awareness of emotions, awareness of identity, and awareness of multiple concurrent conversations.

Following the analysis of these six awareness aspects, the chapter introduces the designs, implementations and evaluations of two innovative prototypes—Relaxed Instant Messenger (RIM) and Conversation Dock (ConDock). RIM combines a threaded metaphor and a sequential metaphor to enhance awareness of group conversation, while ConDock utilises a fisheye view to enhance awareness of multiple, concurrent, one-to-one conversations.

RIM is a client-server application that was written in Java. ConDock was implemented as a plug-in for MSN Messenger, using the transparent adaptation approach. The transparent adaptation approach utilises the MSN Messenger APIs (Application
Programming Interface) to support interaction between ConDock and MSN Messenger. The approach allows implementing ConDock without accessing the source code of the Messenger.

A laboratory-based user test and a field trial were conducted to evaluate RIM and ConDock, respectively. The evaluations confirm that RIM and ConDock meet their design objectives. That is, RIM enhances awareness of a group conversation, and ConDock enhances awareness of multiple concurrent conversations. The evaluations also yielded several implementation issues that need to be resolved in further development of RIM and ConDock. Additionally, this chapter also proposes several designs to support awareness of viewing- and listening-in-progress, and awareness of multiple identities.

Finally, this chapter considers the relation between awareness elements addressed in F@ and those supported by the design of RIM, ConDock and proposed mechanisms. Generally speaking, many awareness elements of F@ are supported by RIM, ConDock and proposed design. Those awareness elements were also found useful in supporting users’ conversational awareness according to the evaluations of RIM and ConDock. This mapping between F@ and actual resulting mechanisms confirms the usefulness of F@ in facilitating the mechanism design.

In the next chapter, another case study of designing awareness support is presented. In particular, the next chapter describes the design, implementation and evaluation of two new mechanisms that are used to enhance workspace awareness in synchronous collaborative authoring.
Chapter 6

Designing Awareness Support for Synchronous Collaborative Authoring

6.1 Introduction

The previous chapter has presented the first case study of applying F@ to the design of awareness support in Instant Messaging. The purpose of this chapter is to present another case study of designing and building awareness mechanisms that are used to support group awareness in synchronous collaborative authoring. The design of mechanisms presented in this chapter has evolved in part from the knowledge of group awareness presented in F@, and in part from the design of existing awareness mechanisms.

Collaborative authoring is defined as “a process where co-authors, with abilities and different responsibilities, interact during the intention and revision of a common document” [134]. Collaborative authoring involves several people who are co-located in the same place or distributed in different places, working together on shared documents synchronously and/or asynchronously. In synchronous mode, co-authors of a shared document interact with one another at the same time, and therefore the changes made by one user are reflected at the other users’ computers almost immediately. However, in an asynchronous mode, users work on a shared document at different times. Generally speaking, groupware systems that support synchronous collaborative authoring can be used for facilitating asynchronous collaborative authoring, but not vice-versa. Although collaborative authoring tasks are often carried out asynchronously, it occurs that people sometimes need to come together to finalise a shared document. Within the scope of this thesis, an investigation of group awareness support is carried out particularly on synchronous collaborative authoring (SCA) of a small group.

30 The work presented in this chapter has been published in [234, 237].
In the CSCW literature, many groupware systems supporting SCA have been developed, such as Quilt [135], GROVE [73], MACE [171], ShrEdit [150], SASSE [10], PREP [170], DOME [251], REDUCE [260] and CoWord [258]. The literature shows that supporting group awareness is one important element of designing usable SCA systems [10]. Group awareness facilitates fluid and efficient collaboration by providing knowledge of an updated status of the document, team members’ tasks and activities. Unfortunately, despite various mechanisms that have been developed to support group awareness in SCA, only few of this type of groupware system are used in the real-world. Current SCA systems have not been able to match the diversity and richness of group authoring interaction [1, 103, 104].

This chapter presents a case study of designing and building mechanisms to support workspace awareness in SCA. The chapter begins in the next section with an analysis of current support for group awareness. Section 6.3 traces the evolution of two awareness mechanisms—Extended Radar View (ERV) and Modification Director (MD)—by describing the origins and the user interfaces of these two mechanisms. Next, the implementation and evaluation of ERV and MD are presented in Section 6.4 and Section 6.5, respectively. Before closing the chapter, Section 6.6 presents a discussion of how F@ has been reflected in the designs of the two mechanisms.

6.2 Support for group awareness in collaborative writing

This section examines support for four awareness schools in SCA, including conversational awareness, workspace awareness, contextual awareness and self-awareness.

6.2.1 Support for conversational awareness

Communication involves an exchange of information between team members, and plays an essential role in facilitating fluid and productive collaboration in SCA. As discussed in Chapter 5, in order to conduct natural and effective communication, team members need to maintain their conversational awareness, which refers to people’s knowledge of a conversation (for example, who is talking to whom). The exchange of information can be supported by either direct communication or indirect communication [156], as discussed in Chapter 2 (Section 2.3.1.1).
• Direct communication implies that team members send direct messages to one another. In distributed collaboration like SCA, computer-mediated communication (CMC) tools, such as IM and chat can be used to support direct communication.

• Indirect communication refers to a way that team members communicate with one another by a means of interacting with and through shared artefacts in the workspace.

In SCA, team members use the text of a shared document and annotations to it as a means of indirect communication. Therefore, providing information about workspace awareness (discussed in Section 6.2.2) is important in supporting indirect communication.

In addition, people often use direct communication to refer to particular shared artefacts. Many conversations in a group focus on referencing artefacts, and discuss a common referential practice. This phenomenon is known as references to artefacts [44]. As reported in the empirical study of SCA (Section 3.3.2.1), people often have difficulty with deictic reference in communication.

Section 2.4.1 examines three means that are often used to support communication in groupware, including text-based chat tools, mediaspaces and collaborative virtual environments (CVEs). In SCA, the first two communication means are more applicable than CVEs due to the difficulty of interacting with artefacts of a shared document in CVEs. This section examines support for conversational awareness in text-based chat tools and mediaspaces.

**Text-based chat tools**

Many groupware systems, such as Calliope [158], CEPE [199], NetEdit [264] and NetMeeting [155] provide text-based chat tools that are often in the form of chat windows or scratchpads, allowing users to send text messages or free drawings between one another. The UIs of such tools are very similar to those of IM and IRC applications, in which text messages are shown in sequential order, as seen in Figure 6.1. Some popular visual cues, which are often shown in the interface of this type of communication tool, include user names, colours and timestamp of each message. For
instance, messages can either include users’ names or be colour-coded to support identification of who is the sender of the message. A timestamp can also be shown to indicate when a message is received.

In Chapter 5, an in-depth discussion of various issues in relation to support for conversational awareness in IM (for example, awareness of presence, awareness of turn-taking, etc.) is examined. The discussion is also applied to groupware systems for SCA. Therefore, the two prototypes—RIM and ConDock—that are described in Section 5.4 can be incorporated into collaborative writing systems to enhance conversational awareness.

However, current text-based chat tools are mainly useful in facilitating direct communication among team members. They provide limited support for indirect communication because the chat tools do not provide a coherent integration of an ongoing conversation with a shared document [43]. That is, windows containing a shared document are often separate from, and have no link to, chat windows. Consequently, it is difficult to refer to artefacts of a document in text-based conversations.

**Mediaspaces**

In addition to text-based chat tools, mediaspaces (discussed in Section 2.4.1.2) are used to support synchronous communication in a distributed group [26, 89]. Mediaspaces provide rich visual and auditory information about physical space at which team
members are located to provide team members with a sense of proximity and co-location, and thereby allow more natural and effective communication [88, 242].

However, in the context of SCA, studies such as [78] showed that visual information provided by video does not improve group performance in comparison to audio-mediated communication. Some other studies, such as the study conducted by Matarazoo and Sellen [148], reported that there was no significant difference in participants’ performance when either providing high quality video or poor quality video. Furthermore, mediaspaces provide limited visual information about artefacts in the shared document. Thus, it is difficult to utilise mediaspaces for supporting indirect communication and deictic reference in SCA.

6.2.2 Support for workspace awareness
Supporting workspace awareness involves providing information about the presence of users, shared artefacts and users’ actions in a shared workspace. Various mechanisms, including those of 2D and 3D-based visualisation, have been introduced to support these aspects of workspace awareness. Examples of 2D-based mechanisms include telepointers, multi-user scrollbars, the radar view, etc. Examples of 3D mechanisms include virtual reality presentations. In the context of SCA, 3D mechanisms are very limited in supporting awareness of shared artefacts and actions that are performed upon the artefacts. As a result, 2D mechanisms are appropriate, and in fact, have been adopted as primary support for group awareness in SCA systems.

The following text examines major existing 2D mechanisms that have been developed to support three aspects of workspace awareness:

(1) people’s presence,

(2) actions and

(3) shared artefacts.

Awareness of presence
In a workspace, a person’s body is often used as a primary source to realise the person’s presence. Bodily movement, such as movement of heads and arms, is a rich source to convey information about the presence and location of people in a shared workspace. The embodiment technique has been considered as an effective technique to support
presence awareness. Telepointers and viewports, which are discussed in Section 2.4.2.2, are major forms of embodiment that are commonly used by SCA systems.

**Telepointers**

As discussed in Section 2.4.2.2, telepointers show the positions and movement of remote users’ mouse cursors in a shared document. To support identification of owners of telepointers, telepointers can be assigned different colours, different shapes or even have the user’s name and/or image attached to it [16, 95] (Figure 2.6). Telepointers are able to convey awareness information about the presence of remote users and their locations in the shared document. Additionally, they inform users of other people’s activity in the document and often the kind of actions as well.

However, telepointers fail to convey presence information when they are located out of the local user’s view. Furthermore, telepointers cannot show the precise location in the document at which remote users are working. In the case of drawing, telepointers are able to show working locations of other users, because the position of a mouse cursor is the location at which a user is drawing. However, in the case of writing, the position of a mouse cursor is not a position at which a user is typing.

**Viewports**

Viewports into a shared document indicate the area that users can see and where they can interact with the document. Figure 6.2 shows a snapshot of Flexible JAMM (Java Applets Made Multiuser), a shared editing groupware that supports viewports [16]. In the figure, two viewports are presented as shared rectangles in the window at the right.

Presenting viewports is useful in showing information about the presence and location of users in the document. The existence of the viewport indicates the presence of a user (for example, the viewport’s owner). Users’ positions are reflected by the locations of viewports. Furthermore, additional information, such as identity information, can be added to the presentation of viewports to make them more expressive. For example, viewports can be presented in different colours, as seen in Figure 6.2, or show users’ images [105].
Unfortunately, viewports fail to distinguish users’ working and viewing areas in the document. Viewports can be interpreted as either working or viewing areas. As a result, if solely viewports are implemented in a shared editor, information about users’ working locations could be either insufficient or incorrect.

**Awareness of actions**

Actions in a shared document are often performed upon shared document artefacts such as text, images and tables. Thus, in order to examine support for actions in a shared workspace, it is useful to analyse a manipulation where actions affect shared artefacts. When artefacts are manipulated, *feedthrough* information is generated [64]. This source of information provides direct feedback to the local user who manipulated the artefacts, and can also be used as awareness information to help remote users stay aware of what is happening in the document. Supporting feedthrough is a popular form of sharing information about the states of shared artefacts between users. Feedthrough information can be provided by a direct or indirect manipulation [107].

**Direct manipulation**

Direct manipulation of an artefact (for example, dragging a table from one page to another page in a shared document) is a popular and perceivable form of action in SCA. In order to provide awareness of direct manipulation, groupware systems need to display initial, intermediate and final states as an artefact is manipulated. This form of feedthrough is defined as *action feedthrough* [105]. For example, in the case of
dragging a table in a document, display of feedthrough information presented to remote users involves the initial position of the table, intermediate positions as the table is being moved, and the final position of the table.

Several mechanisms supporting action feedthrough have been developed. For example, strict-WYSIWIS (Section 2.4.2.4) that enforces all users to see the same view is implemented in early groupware systems to support feedthrough [210]. When all users see the same view of a document, actions occurring in the document are perceivable by both a local user and remote users. However, WYSIWIS is too restrictive as participants of a collaborating group often move back and forth between individual and shared activities [197].

To overcome the inflexibility of strict-WYSIWIS, relaxed-WYSIWIS mechanisms, such as the radar view [110], Workspace Teleportals [108], fisheye views [93], the gestalt view [10] and WYSIWID (What You See Is What I Do) [108], have been developed. These mechanisms allow users to change their views independently, while still maintaining knowledge of other users’ viewports. Because remote users’ viewports are visible to a local user, the local user is informed of action feedthrough.

**Indirect manipulation**

In document authoring, artefacts can be manipulated indirectly, such as via menu selection and shortcut commands. An example of indirect manipulation in document authoring includes formatting text (for example, italics, bold, or changing font size). These types of operations occur almost instantly, and provide almost no transitional information (that is, only the initial and final states are shown). Thus, providing feedthrough of indirect manipulation is less perceptible and less noticeable than the direct counterpart.

**Process feedthrough** [105] is a fundamental technique that is used to help remote users stay aware of indirect manipulation. Process feedthrough provides a certain amount of information about an indirect manipulation action to remote users. Representation of intermediate states can be supported by action animations, indicators, or sound [163]. Figure 6.3 shows an example of a symbolic indicator of formatting text (for example, italics). When text is formatted in italics, a symbolic indicator appears on remote users’ screens to indicate the italicised text.
Awareness of shared artefacts

As discussed above, feedthrough information about direct and indirect manipulation involves displaying states of shared artefacts as being manipulated. Here, additional information related to a lifecycle and the authorship of artefacts is considered.

Awareness of artefact lifecycle

Artefacts in a shared document can be manipulated many times. For the purpose of supporting workspace awareness, it is useful to keep track of an evolving lifecycle of an artefact. A common technique of showing a lifecycle of an artefact is recording and replaying changes that were made to the artefact in the past. Details of a change often include information about when the change occurred, who made the change and what was the nature of the change.
For example, the Flexible Diff collaborative editor [169] displays changes inline. As shown in Figure 6.4, Flexible Diff presents four columns of text to communicate the detail of a change. The original and changed versions of text are in the first and second columns, respectively. The third column shows the difference between these two versions, while the last column displays additional annotations attached to the change.

**Awareness of authorship**

Providing information about the creator of an artefact is useful. Information about authorship of artefacts helps users keep track of their work and thereby avoids confusion and minimises unexpected modifications made by other people. The authorship information in a shared document is often indicated by the use of different colours for different users. For example, Calliope [158] and REDUCE [260], as shown in Figure 6.5 and Figure 3.1, respectively, display text entered by local and remote users in different colours and also allow them to customise their colours.

6.2.3 **Support for contextual awareness**

As indicated in Section 4.3.3 and Section 4.4.3 in Chapter 4, within the scope of this thesis, contextual awareness refers to users’ knowledge of other team members’ goals, tasks and results. To support contextual awareness, SCA groupware systems provide mechanisms that can help users keep track of team members’ tasks and the progress of their work. For example,

- **GROVE (GRoup Outline Viewing Editor) [73]** is a synchronous collaborative editor that includes support for depicting group activity, specifying the roles of team members and changing roles (Figure 6.6a).

- **The Quilt editor [135]** defines and supports three social roles in collaborative writing—co-author, commenter and reader (Figure 6.6b).
• Col•laboració [39] is a Web-based collaborative writing application that supports contextual awareness by informing users of other team members’ activity and of their writing pace.

![Figure 6.6: Support for contextual awareness in collaborative authoring. Snapshots of (A) GROVE [73]; (B) Quilt [135].](image)

### 6.2.4 Support for self-awareness

Self-awareness involves users’ perception of their own actions, activity and progress in a group (Section 4.3.4). In other words, SCA groupware systems consider local users similar to remote users in the sense that awareness information about local users’ presence and activity is conveyed like that of remote users. For example, local users perceive both feedback of their own actions and feedthrough of remote users’ actions. When local users scroll up and down a shared document, they should see their viewports moving on the radar view.

Up to this point, this chapter has discussed a variety of mechanisms that existing SCA groupware systems provide to support the four awareness schools of conversational awareness, workspace awareness, contextual awareness and self-awareness. These four awareness schools cover various aspects of awareness and therefore require different support mechanisms. This chapter presents a case study of designing mechanisms to support group awareness in SCA. It is not realistic for this case study to develop a variety of mechanisms to address all four awareness schools. The next part of the chapter presents two mechanisms that were designed to enhance *workspace awareness* in SCA.
6.3 Design of CoWord+

CoWord [258] is a multi-user word processing system that was developed based on one of the most popular single-user word processing applications—Microsoft Word (referred to as Word for short henceforth). Two major advantages of CoWord are that CoWord has inherited a set of rich functionalities and user interfaces from Word, and also allows unconstrained simultaneous interactions of users in the same document. From the perspective of the users, retaining single-user Word functionalities to which users have been accustomed, and turning them into multi-user ones in a transparent manner, is important. CoWord, therefore, holds great potential in accommodating SCA tasks. Unfortunately, the current implementation of CoWord provides limited support for group awareness; merely telepointers and the radar view are presently provided by CoWord, as seen in Figure 6.7.

---

31 The term “transparent” is used here to indicate that users of CoWord will not notice any difference between executing single-user Word operations and the multi-user replicas.
To improve support for group awareness in CoWord, two new awareness mechanisms have been developed. CoWord+ is an extension of CoWord with the awareness mechanisms added. The two mechanisms include Extended Radar View (ERV) and Modification Director (MD). The following two sections discuss the origins of ERV and MD, and then describe their current user interface design.

6.3.1 Design of ERV

6.3.1.1 Design rationale
A shared document is a computational workspace that changes dynamically reflecting multiple inputs from users. Hence, it is important for users to be provided with easy access to information about the current state of the document. Furthermore, in order to help users maintain group awareness, users need to be presented with detailed visualisation of what other users are doing and viewing. Unfortunately, all too often, the view of a joint document is larger than the size of a screen. Consequently, users are only able to see a partial screenful of the document at a time. Two visibility problems associated with a document exist:

1. the inability to see the entire shared document (that is, the context problem), and
2. the inability to see detailed views of what other people are looking at in the document (that is, the focus problem).

To address these problems, two popular visualisation techniques—the radar view and fisheye views—have been constructed. Here, advantages and disadvantages of these two visualisation techniques are discussed, thereby showing how they inspire the design of ERV.

Radar view
To address the visibility problem in SCA, groupware systems often use the radar view as a solution to the context problem. The radar view is a miniature representation of a shared document that shows a high-level overview of the entire document as well as superimposed users’ locations of activity through viewports and telepointers (Section 2.4.2.2). The literature has shown that the radar view is useful in providing users with knowledge of a general structure of the document and users’ whereabouts in the document [105].
However, the radar view poses three usability problems [228]. First, the low-resolution representation of the radar view conceals details of other users’ actions. Thus, a viewport shown on a radar view of a large workspace contains too little detail to be useful (that is, the focus problem). Consequently, to determine another user’s working location exactly, a local user has to align viewports inconveniently by either dragging the local user’s viewport in the radar view or scrolling the user’s main view.

Second, as Greenberg et al. [110] pointed out, the radar view creates a contextual virtual gap between local details and the global context. Unnaturally, the radar view forces users to make an abrupt context shift between views of different scale when they interpret awareness information provided by the radar.

Third, the radar view fails to distinguish between users’ viewing areas and working areas in the document. Although viewing and working areas are usually the same, in certain cases they can be different. However, users’ viewports drawn in a radar view could be interpreted either as their viewing sections or as their working sections. As a result, if only the radar view is used for awareness, awareness information about other users’ working locations provided by the radar could be either insufficient or incorrect. In the literature, these two views have been considered as two separate aspects of conveying activity [19]. The empirical study of SCA, reported in Section 3.3, also shows that it is useful to separate these two views [227].

The above discussion shows that the radar view is useful in addressing the contextual visibility problem in SCA, but the radar encounters the focus problem. A fisheye view is a “focus+context” visualisation technique designed to accommodate the twofold need for showing a detailed area as well as a full context view [34, 35, 205].

**Fisheye views**

As discussed in Section 2.4.2.2, fisheye views present a single continuous surface that is able to display both local detail and the global context. Fisheye views also allow users to magnify different regions with various levels of magnification. This advantageous feature of fisheye views, therefore, addresses the contextual gap problem of the radar view by providing a seamless and smooth transition between detail and context views. In addition, multiple focal points of a workspace are supported by fisheye views. Previous evaluations of fisheye views show that fisheyes are found useful to some
extent in several task domains including graph layout, webpage navigation and menu selection [15].

However, previous studies also show that the distortion presentation of fisheyes causes usability problems in targeting, navigating and reducing users’ ability to remember locations as they shift between magnified and de-magnified areas [205]. In addition, fisheye views have not addressed the need for distinguishing users’ working and viewing areas in a shared document. Although the distortion visualisation of fisheye views addresses the detail-in-context issue, it does at the cost of an individual’s workspace estate: when multiple focal points are displayed in fisheye views, the size of a user’s viewport is proportionally reduced.

Given the advantages and limitations of the radar view and fisheyes view discussed above, this thesis has developed a variant of the radar view, thus so called “Extended Radar View”. ERV addresses the three usability issues of the conventional radar view mentioned earlier, particularly in the SCA task context:

(1) ERV provides an adjunct view that magnifies the detail shown on the radar view;

(2) The adjunct view also reduces the contextual gap by minimising gaze shifting between the radar view and the main view; and

(3) ERV provides two independent views that show where a user is working and viewing in a shared document respectively.

### 6.3.1.2 User interfaces of ERV

As described earlier, the radar view holds great potential in supporting group awareness for SCA, but it encounters three usability issues, including:

(1) limited support for viewing details shown on a low-resolution presentation,

(2) the contextual gap between the radar and the document, and

(3) no separation between users’ working and viewing areas in the document.

ERV was designed to address these three problems of the radar view. The user interfaces of ERV include three main views: the radar view, over-the-shoulder view, and telecarets-eye view.
Figure 6.8 illustrates the scenarios when two users—Jane and Tarzan—are working on a shared Word document. The radar view is shown in the middle, Jane and Tarzan’s over-the-shoulder views are shown at right, and their telecarets-eye views are shown at left.

**Figure 6.8: Extended Radar View (ERV).**

### Radar view

The radar view displays an overview of a Word document in a miniature form. The display is updated to reflect changes occurring in the document (for example, an image is deleted, dragged or added). The radar view provides **three major features.**

First, the radar view supports users’ embodiment by superimposing **viewports** and **telecarets** on the display. The embodiment is updated according to the movement that users make on the document. Viewports on the radar are represented as coloured transparent rectangles, and are used to convey awareness information about where users are currently looking in the document. Telecarets represent remote users’ text insertion cursors, and are shown as small, vertical, coloured bars. Telecarets are useful for...
showing exact locations of where remote users are working in the document. The size of telecarets is small, thus they are not distracting when moving in the radar view. It is often that a user’s telecaret is situated inside the user’s viewport. But in some cases when a user is working at one section in the document but looking at another section, then the telecaret is located outside the viewport. Figure 6.8 shows the radar view of a shared document, with two viewports and two telecarets. Tarzan’s telecaret is inside his viewport, but Jane’s telecaret is located outside her viewport.

Second, the radar view is an interactive display in the sense that it allows users to navigate the main document by dragging viewports in the radar: when a viewport is dragged, the main view of the document is scrolled to the corresponding location.

Third, the radar view supports a highlighting feature. When a user highlights a section of the shared document, that highlighted section is also marked in the display. This feature was implemented based on the findings from the empirical study of SCA, reported in Section 3.3.

**Over-the-shoulder view**

The over-the-shoulder view is a magnified version of a viewport shown in the radar view. The objects appearing in the over-the-shoulder view are smaller than their full size, but are much more readable than those in the radar view [10, 107]. As discussed earlier, a shortcoming of the radar view is the low-resolution display of contents shown in it. The design objective of the over-the-shoulder was, therefore, to address this focus problem by improving the visibility and the magnification of objects shown in the radar.

In ERV, the over-the-shoulder view is an adjacent view to the radar view, and magnifies details appearing in a viewport of the radar view in such a way so that they become readable: in most cases users can read the objects appeared in the over-the-shoulder view without having to refer back to the main document. Consequently, the over-the-shoulder view reduces gaze shifting between the radar view and the main view. Thus, it bridges the contextual gap between the radar view and the main document. In Figure 6.8, the over-the-shoulder views are shown on the right, and they link to two corresponding viewports in the radar view.
Furthermore, the over-the-shoulder view is updated whenever the main document is scrolled or whenever viewports in the radar view are dragged. And, the over-the-shoulder view also supports the display of a highlighted area like the radar view.

**Telecarets-eye view**

The radar view includes telecarets to show the locations of remote users’ insertion cursors. Telecarets convey awareness information about where the users are working, but at the cost of *document context*: telecarets in themselves provide no information about their surrounding contents. The telecarets-eye view was designed to address this context problem of a stand-alone telecaret.

The telecarets-eye view shows a limited area around a remote user’s telecaret. The objects shown in the display are smaller than those in the full-size viewport, but are larger than those in the radar view. This view allows clear viewing of the contents surrounding another user’s telecaret. In Figure 6.8, the telecarets-eye views are shown on the left.

The design of the telecarets-eye view is similar to WYSIWID [108] in the sense that the two mechanisms present contextual information about a surrounding area at which remote user’s actions occur. One the one hand, WYSIWID provides a selected area around a user’s telepointer. On the other hand, ERV provides a selected area around a user’s telecaret. However, in comparison to telepointers, telecarets present more accurate information about users' writing actions and contribution to a shared document. Furthermore, ERV includes the radar view to provide the global context of the entire shared document and uses the telecarets-eye view to provide the local context of remote users’ working locations.

Figure 6.9 compares the radar view, the over-the-shoulder view and the telecarets-eye view in terms of two parameters—view extent and view size. The view extent parameter refers to an area of the workspace in full size (that is, what a local user sees on his/her screen), whereas the view size parameter refers to a corresponding area but reduced in size. For example, in the first block shown in Figure 6.9, a viewport on the radar view shows the entire area, that is, the view extent is the same as a user’s actual viewport. However, the view size of a viewport on the radar is smaller than the user’s full-size viewport.
6.3.2 Design of MD

6.3.2.1 Design rationale

It often occurs in SCA that a user changes other people’s text [158]. This writing behaviour is part of cooperative work, and in some cases it is acceptable for those who are not concerned about whether their work is modified by others. However, in many other cases, they need to be informed of such modifications because those changes can affect the process and outcome of group collaboration. The term *modification awareness* is defined to refer to the capability of helping users stay aware of modifications that other users have made to their text.

This section describes the design background of MD, which supports modification awareness in a shared Word document. The design of MD has evolved in part from previous studies of document change tracking and annotations, and in part from the empirical user-based study of SCA (reported in Section 3.3) [227]. This section describes the design background of MD in terms of the properties of modification awareness, the initial sketch of MD, and Word’s current support for tracking changes.

**Properties of modification awareness**

Modification is an iterative and dynamic process and should be associated with a specific context. In addition, modification is part of a shared document, thus, it needs to be linked coherently to the document. In order to support modification awareness in SCA, the first step is to identify the properties of modification awareness.

Several studies, such as [32, 216, 253], have examined support for annotations and changes in documents. Although those studies mostly focus on asynchronous authoring,
they are highly useful for the study of modification awareness in SCA. This is because in both asynchronous and synchronous situations, annotations and modifications are dynamic activities in joint document authoring. Annotations and modifications need to be linked coherently to the shared document, and need to be attached to a certain context to be understood. Based on the previous studies of annotations, five properties of modification awareness are defined, which serve as a guideline for the design of MD.

- **Modification Context**: This property shows a meta-description of a modification to provide the context of a modification (for example, the anchor point at which a modification occurs, the surrounding text of the anchor point, comments on a modification, etc.). Without being situated within a specific context, a modification might likely be “orphaned” and therefore is difficult to understand in many cases [32].

- **Modification Body**: This property refers to the actual content of a modification. For example, if a modification is a delete operation, users should be able to view a partial and/or full content body of the deletion.

- **Modification Creator**: This property conveys information about a user who made a modification (for example, the user’s name).

- **Modification Time**: This property provides temporal information about when a modification was made. The value of this property needs to be updated according to a life cycle of a modification, as discussed in the status property (explained next).

- **Modification Status**: SCA is an iterative process, and so can be modifications. It is possible that users keep making changes to the same text. Thus, an argument here is that a modification has its own “life cycle” in which the nature or content of a modification is changed. For example, suppose a Word document contains text $T = \text{“hello world”}$. First, $T$ is formatted in bold, “**hello world**”, and later $T$ is italicised “*hello world*”. These two changes occur at different times, but to the same text. Therefore, the modification status should be updated to reflect this multiple-change modification. In addition, because the life cycle of a modification can involve many changes, the temporal property (that is,
Modification Time) needs to be renewed accordingly to convey the most recent change.

**Initial mock-up of MD**

As mentioned above, the current design of MD has also evolved in part from the empirical study of SCA. One finding emerging from the study is the design mock-up of MD [227] that was proposed by the participants (Section 3.3.2.2). In Figure 6.10, the mock-up illustrates the scenario when Tom modifies Kim’s text: a corresponding coloured icon flashes instantly on Kim’s screen to inform Kim of the change. Kim can click on the flashing icon to view the detail of the modification.

The mock-up is very much a sketch of the participants’ ideas and still contains several usability problems (for example, pop-up window and flashing icons). However, the mock-up is useful in the sense that it justifies the need for supporting two aspects of modification awareness, including notifying users of a modification, and showing the context of the modification.

**Word support for tracking changes**

In Word, MD is developed based on the change tracking feature. Here, a brief introduction of several properties, which are made available by the change tracking
feature, is presented. Figure 6.11 illustrates three main functions of tracking changes that are currently supported by Word, including deletion, insertion and formatting.

When change tracking mode is turned on, Word provides a reviewing pane allocated on the right-hand side of the main document. The reviewing pane displays information about these three types of modifications. For example, for a deletion, the content of a deletion is shown as a bubble on the reviewing pane, and Word provides a dotted line to link the deletion bubble with the deletion anchor in the document. Word provides an informative description of a modification, such as when a modification was made, who made it, where it occurred in a document, what is the content of a modification, and what is the type of a modification. Word also provides an annotation that allows users to justify the motive of a modification.

However, Word provides limited information about the status of a modification. For example, if a user updates an existing insertion by adding more content into it, Word does not provide any visual or textual cue to indicate that the insertion is altered. Table 6.1 summarises Word’s support for tracking changes, in relation to the five properties of modification awareness discussed earlier.
In the next section, the actual design of MD is described, and an analysis of the design addressing these five properties of modification awareness is presented.

6.3.2.2 User interfaces of MD

In a Word document, a modification of text includes:

1. two primitive operations of deleting existing text and of inserting new text (the replace operation is a composite operation of deletion and insertion), and
2. content-formatting operations, such as changing font sizes, colours, aligning text, and so on.

The current version of MD only tracks a deletion operation—MD notifies a local user whenever his/her text is deleted by other users. The implementation of MD can easily be extended to track other types of modification, such as insertion and formatting.

MD was designed to support the five properties of modification awareness, including:

- Modification Context,
- Modification Body,
- Modification Creator,
- Modification Time, and
- Modification Status.

As discussed in Section 6.3.2.1, Word currently supports these five properties to a certain extent, adopting situated placement to present modification awareness [107]. That is, information about modification awareness is displayed within the workspace (that is, a shared document). MD retains what has been supported by Word, and provides additional support for modification awareness using separate placement.
Separate placement refers to the presentation of information outside the workspace in a separate view [107], as discussed next.

Figure 6.12 shows the user interfaces of MD at the bottom and a shared Word document at the top (reduced in size). MD includes a list of modifications that are sorted in chronological order with the most recent modification at the bottom. Whenever a user’s text is deleted, partial content of the modified text is pushed to the bottom of the list, with a highlighted background colour. This background colour fades gradually after a period of time indicating the age of a modification [13].

**Figure 6.12: Modification Director (MD).**

First, MD supports the modification context by displaying information about where in a document the modification was made. In Figure 6.12, MD shows a page number at which a modification occurs in the document (for example, the first modification was made at “page 7”).
Second, MD supports the modification body by showing a *symbolic* representation [107] of a deletion. That is, only a subset of the content of a deletion is shown. As seen in Figure 6.12, MD shows the first $n$ characters$^{32}$ of a modification body (for example, “be aware of evolving status of group”). In addition, MD also links modification items to their corresponding anchor text in the main Word document. In Figure 6.12, when a user clicked on the modification item—“MIH is responsible for intercepting m...”—the item’s corresponding anchor text is highlighted in the reviewing pane of the Word document.

Third, MD supports the modification creator by showing the name of a user who made a modification (for example, Minh is a modification creator in Figure 6.12).

Fourth, MD supports the modification time by displaying elapsed time that measures for how long the modification has been made (for example, “0:6:11”, six minutes and eleven seconds, in Figure 6.12). This temporal property is updated if a multiple-deletion modification occurs, to ensure that the modification time always shows the most recent change.

Finally, MD supports the modification status by providing information about the life cycle of deletion (for example, single deletion or multiple deletions). MD distinguishes between a single-deletion operation and a multiple-deletion operation in a Word document.

- A single deletion operation refers to a deletion that occurred only *once* at a particular location in a document.

- A multiple-deletion operation refers to a deletion which is an accumulation of *many* deletions occurring at the same location in a document. For example, suppose a shared document initially contains User1’s text $T = “abcdef$”. If User2 deletes “bc”, then $T = “adef$”. This operation is considered a single-deletion operation. Next, User1 adds “12” to $T$, so $T = “a12def$”. User2 then deletes “12”, thus $T = “adef$”. So, an accumulated deletion is “bc12”, and this is considered a multiple-deletion operation.

---

$^{32}$This number of characters varies depending on the size of the MD window. At the moment, the maximum value of $n$ is set to 256, but this number can be easily changed.
The distinction between single deletion and multiple deletions is important, as it helps inform users that an existing deletion in MD has been further modified.

Table 6.2 shows how MD supports the five properties of modification awareness, which are discussed in Section 6.3.2.1.

<table>
<thead>
<tr>
<th>Properties of modification awareness</th>
<th>MD supports for modification awareness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modification Context</td>
<td>Apart from Word’s support for modification awareness, MD shows additional information about the context of a modification (for example, page number).</td>
</tr>
<tr>
<td>Modification Body</td>
<td>Supporting symbolic and literal display of modification body. If the content of a modification is long, a partial body of the modification is shown by MD (that is, symbolic presentation). MD also allows users to view the entire content of the modification as supported by Word.</td>
</tr>
<tr>
<td>Modification Creator</td>
<td>MD provides the name of a user who made a modification.</td>
</tr>
<tr>
<td>Modification Time</td>
<td>MD shows elapsed time since a modification was made. This information is updated when the modification is altered. MD also sorts modifications in temporal order, and uses the colour-fading technique to convey the age of a modification.</td>
</tr>
<tr>
<td>Modification Status</td>
<td>MD updating the status of a modification, such as a single deletion or multiple deletions.</td>
</tr>
</tbody>
</table>

Table 6.2: MD supports modification awareness.

6.4 Implementation of CoWord+

ERV and MD were developed in light of the transparent adaptation approach utilising the Word APIs [237]. Word provides a comprehensive set of APIs including objects, methods and events conforming to COM (Component Object Model) Automation. Adapting these APIs, ERV and MD are able to access and manipulate properties of document objects, such as text, images, viewport, etc. [153].

ERV and MD intercept users’ local operations and transform them into abstract forms which are then interpreted at remote sites via the APIs. Interactions amongst ERV, MD and Word are transparent via the Word APIs. Hence, there is no need to access the source code of Word. Since the approach to developing ERV is very similar to that of MD, details of the implementation presented here apply to both ERV and MD.

The system architecture of ERV and MD includes two main processes: API Adaptation and Message Interception, as illustrated in Figure 6.13. The functions of API Adaptation and Message Interception are introduced in Section 5.4.2.
**API Adaptation**: To support group awareness, ERV and MD are required to know both the evolving statuses of Word data objects and users’ interaction with Word. For this reason, the Word APIs are categorised into two groups: *Object Interfaces* (OIs) and *Event Interfaces* (EIs). OIs allow ERV and MD to access and manipulate *properties* of data objects of Word, whereas EIs allow ERV and MD to listen to *events* generated by users’ interaction with Word.

**Message Interception**: In addition to interacting with Word, ERV and MD also intercept messages (for example, data message, events, etc.) sent between the operating system and Word. This message interception is important because not all events are supported by the Word EIs.

![System architecture of ERV and MD](image)

**Figure 6.13**: System architecture of ERV and MD.

Both ERV and MD are responsible for four main tasks:

- First, they need to interact with Word through the Word APIs, and intercept event messages sent between the operating system and Word.

- Second, ERV and MD have to process data gathered from a local site after carrying out the first task.

- Third, ERV and MD need to communicate with another ERV and MD respectively at remote sites to send local data and receive remote data of users’ interactions.
The final task of ERV and MD is to display data coming from the local site and remote sites in relevant graphical representations.

To support these four tasks, four software layers are included in the design of ERV and MD. The four layers are Adaptation Layer, Data Layer, Communication Layer, and Presentation Layer (Figure 6.14).

**Figure 6.14: Four-layer architecture of ERV and MD [237].**

**Adaptation Layer (AL)**
AL is responsible for interacting with Word via the Word APIs, and for intercepting messages between Word and the operating system. AL includes two modules: Message Interception Handler (MIH) and API Consumer (APIC), as seen in Figure 6.14a. MIH is responsible for intercepting event messages sent between the operating system and Word. APIC is a software module that is responsible for gathering data and events from Word via its OIs and EIs (Figure 6.13). For these purposes, APIC consists of two sub-modules: OI Consumer (OIC) and EI Consumer (EIC).
Figure 6.15 shows a solution of how APIC adapts the Word APIs using the COM technology. The Word COM objects, such as Application, Document, Selection, etc. expose their interfaces that allow ERV and MD to access and manipulate the objects’ properties. APIC is an ActiveX client that accesses the Word COM objects using the IDispatch interface provided by the objects. APIC creates a list of references to object interfaces corresponding to a list of Word COM objects. For example, a WA_Application object in APIC maps onto an Application object in Word. The term “WA” in front of each object’s name stands for “Word Adapted”.

**Data Layer (DL)**

DL is responsible for receiving and processing awareness data of Word objects and users’ interaction. DL includes two modules: Local Interaction Handler (LIH) and Remote Interaction Handler (RIH) (Figure 6.14b). LIH and RIH are responsible for processing data from local and remote sites (for example, cursor movement, viewport changes, etc.), respectively.
Communication Layer (CL)
CL is responsible for the communication between distributed awareness mechanisms (for example, ERV and MD). To support awareness in group collaboration, an awareness mechanism must provide information about both the local user’s and remote users’ interaction. The Word APIs do not provide interfaces for transmitting data between awareness mechanisms, as Word is a single-user application and, moreover, it does not have knowledge of what ERV and MD want to exchange. Therefore, CL is required.

CL includes two modules: Connection Manager (CM) and Local/Remote Interaction Exchange (LRIE) (Figure 6.14c). CM is responsible for establishing and maintaining a connection channel between awareness mechanisms. LRIE is responsible for sending data of the local interaction and receiving data of the remote interaction.

Presentation Layer (PL)
PL is responsible for presenting data gathered by DL in a selective graphical form. PL uses different visualisation techniques (for example, zooming) to display awareness information collectively. A separation of data (that is, DL) and presentation (that is, PL) is necessary to provide selective and tailorable awareness support. For example, it may occur that DLs collect the same data for two users, but PLs at the sites can show different presentations depending on different needs of users.

PL includes two modules: Local Awareness Glyph (LAG) and Remote Awareness Glyph (RAG) (Figure 6.14d). LAG and RAG are responsible for displaying awareness information coming from local and remote users respectively (for example, LAG shows local feedback, RAG shows remote feedthrough, etc.).

6.5 Evaluation of CoWord+

6.5.1 Laboratory-based evaluation of CoWord+

6.5.1.1 Goals of the evaluation
A laboratory-based user test was conducted to evaluate CoWord+. The evaluation of CoWord+ focuses on four main objectives:

(1) First, the evaluation tests three design objectives of ERV, including addressing the focus problem of the conventional radar view, bridging the contextual gap
between the radar view and a main document, and showing both working and viewing locations.

(2) Second, the evaluation verifies the usefulness of MD in supporting participants’ awareness of modification in a shared document.

(3) Third, the evaluation explores the usefulness of the dual monitor setting in enhancing awareness support in the context of SCA.

(4) Finally, the evaluation uncovers the strengths and weaknesses of the design of ERV and MD. These results will be useful for further improvement of the two mechanisms.

6.5.1.2 Participants

Sixteen participants, including seven females and nine males, were involved in the evaluation of CoWord+. Participants were recruited from Swinburne University of Technology, and averaged 21 years in age. Participants were allocated into eight pairs. Each pair participated in one two-hour test session to perform three SCA tasks.

6.5.1.3 Tasks

SCA tasks of the test involved writing argument articles. Participants of each pair were asked to work together to produce three articles. An example of one task is:

“Some people believe that university students should be required to attend classes. Others believe that going to classes should be optional for students. Which point of view do you agree with? Within 30 minutes, write an article with specific reasons and examples to support your answer.”

The SCA tasks were designed to ensure that participants needed to collaborate with one another in order to complete the tasks. For example, participants had to discuss and agree whether they were for or against a supplied statement, and then worked out how they would support their argument.

6.5.1.4 Experimental procedure

The evaluation involved eight test sessions. In each session, a pair of participants worked on three SCA tasks under three different conditions, as shown in Table 6.3.
Table 6.3: Three test conditions of the CoWord+ evaluation.

<table>
<thead>
<tr>
<th>Test condition</th>
<th>Monitor setting</th>
<th>Awareness mechanisms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test condition 1</td>
<td>Single monitor</td>
<td>MD and Radar View</td>
</tr>
<tr>
<td>Test condition 2</td>
<td>Single monitor</td>
<td>MD and ERV</td>
</tr>
<tr>
<td>Test condition 3</td>
<td>Dual monitors</td>
<td>MD and ERV</td>
</tr>
</tbody>
</table>

In the first test condition, a pair performed their tasks in a single monitor setting, and the two awareness mechanisms of MD and the conventional radar view were used. In the second test condition, the pair also carried out their tasks in a single monitor setting, but MD and ERV awareness mechanisms were used. In the third test condition, they performed a task in a dual monitor setting, and MD and ERV were used (Figure 6.16). The purposes of these settings were to compare the conventional radar view with ERV, and to compare a single monitor setting with a dual monitor counterpart.

![Figure 6.16: The dual monitor setting. (A) primary monitor shows a shared Word document; (B) secondary monitor shows ERV and MD awareness mechanisms.](image)

After completing the three SCA tasks, participants filled in a 7-point Likert scale questionnaire (for example, “How useful is ERV in supporting your tasks?” with the scores ranging from 1 as not useful to 7 as useful), and took part in an informal interview. In the interview, participants were asked to comment on their experience with ERV, MD and the dual monitor setting.

---

33 The Likert scale questionnaire was developed based on Questionnaire for User Interface Satisfaction [42] and Purdue Usability Testing Questionnaire [137], and is included in Appendix B.3.
6.5.2 Results of CoWord+ evaluation

The results of the evaluation of CoWord+ are reported corresponding to the three test objectives mentioned earlier, including testing the usefulness of ERV, MD, and the dual monitor setting in supporting group awareness.

6.5.2.1 Evaluation of ERV

As discussed in Section 6.3.1.2, the design goal of ERV is to address three usability problems of the conventional radar view. ERV was, therefore, evaluated with regard to the following three design objectives:

1. addressing the focus problem of the radar view by providing the over-the-shoulder view;
2. aligning the over-the-shoulder view and the viewport in the radar view to bridge the contextual gap occurring in the stand-alone radar view; and
3. showing both users’ working and viewing areas in a shared document by providing the over-the-shoulder view and the telecarets-eye view.

Addressing the focus problem

The majority of the participants (more than 85%) responded that the over-the-shoulder view helps resolve the poor visibility of the radar view. As shown in Table 6.4, overall the over-the-shoulder view was found useful in showing magnified content of a viewport in the radar view (mean = 6.06; std. dev. = 0.54; n = 16).

<table>
<thead>
<tr>
<th>Questions</th>
<th>Mean (n = 16)</th>
<th>Std. dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>The over-the-shoulder window is useful in showing a detailed view of what another user is viewing.</td>
<td>6.06</td>
<td>0.54</td>
</tr>
<tr>
<td>It is convenient and easy to map views shown in the radar and the over-the-shoulder window.</td>
<td>4.88</td>
<td>1.26</td>
</tr>
<tr>
<td>It is useful to separate other users’ working and viewing areas.</td>
<td>5.12</td>
<td>1.09</td>
</tr>
<tr>
<td>The over-the-shoulder view (at right) is useful in facilitating your tasks.</td>
<td>4.81</td>
<td>0.83</td>
</tr>
<tr>
<td>The telecarets-eye view (at left) is useful in facilitating your tasks.</td>
<td>4.06</td>
<td>1.06</td>
</tr>
</tbody>
</table>

Table 6.4: Participants’ responses to the usefulness of ERV.
Scores range from 1 = “strongly disagree” to 7 = “strongly agree”.
**Bridging the contextual gap**

Because the over-the-shoulder view is able to provide a detailed view of what is appearing on the radar view, it reduces gaze shifting between the radar view and the main document. The study shows that participants found it easy and convenient to map the content appearing in the radar view and the corresponding magnified display shown in the over-the-shoulder view \((mean = 4.88; std. dev. = 1.26; n = 16)\). Furthermore, the over-the-shoulder view is attached adjunct to a remote user’s viewport, thus users are no longer required to align the radar view and the main document. As commented by participants,

“I often looked at the right window [over-the-shoulder view] to see what other people are doing, especially when I noticed that the window was moved”.

“I had no difficulty in using the middle window [radar view] and the right window, it is pretty straightforward”, and

“It is good to bind the radar to the right window [over-the-shoulder view] because it is not so easy to scroll my radar to the exact location of another person’s radar”.

**Showing both working and viewing areas**

In order to justify the usefulness of ERV in providing two separate views that show where other users working and looking, two aspects are analysed:

1) **necessity** of separating the two views, and

2) **importance** of the telecarets-eye view (that is, working view) and the over-the-shoulder view (that is, a looking view).

The results show that participants agreed it is necessary to distinguish the two views \((mean = 5.12; std. dev. = 1.09; n = 16)\). However, the study showed that the over-the-shoulder view was used more often than the telecarets-eye view. One participant stated, “I did use the left window [telecarets-eye view] as much as the right window [over-the-shoulder view] because the left window is smaller, and they are often the same. But having both views is sometimes useful. For example, when I saw the two windows are located far from one another, I knew that the other person was browsing around.”
It is observed from the study that most of the time participants’ working and viewing areas were identical. In other words, the telecarets-eye view often shows partial content of the over-the-shoulder view. As a result, participants often used the over-the-shoulder view to track where other people were working and viewing. As seen in Table 6.4, participants rated the importance of the over-the-shoulder view and the telecarets-eye view 4.81 (std. dev. = 0.83) and 4.06 (std. dev. = 1.06), respectively. A t-test comparison shows that the over-the-shoulder view is rated more useful than the telecarets-eye view by participants: $t(15) = 4.39; p$-value < 0.05.

### 6.5.2.2 Evaluation of MD

MD was designed to help users stay aware of changes that other users make to their text. In particular, MD supports the five properties of modification awareness, including *Modification Context*, *Modification Body*, *Modification Creator*, *Modification Time*, and *Modification Status* (discussed in Section 6.3.2.2). MD is evaluated in two aspects: the overall usefulness of MD, and the usefulness of each of the five properties.

#### Overall usefulness of MD

As seen in Table 6.5, the overall participants’ rating shows that information displayed in MD is easy to understand ($mean = 5.56$, $std. dev. = 0.89; n = 16$), and was found useful to some extent in supporting modification awareness ($mean = 4.38$, $std. dev. = 0.89; n = 16$). An observation of participants’ use of MD shows that MD was often used nearly at the end of an experimental session when a task was almost completed. From the beginning to the middle of a session, participants tended to focus on performing their own tasks, and paid little attention to MD. Near the end of the session, when a document was rather complete, participants looked at MD to see what changes were made by others.

Generally speaking, MD was not used to notify “up-to-the-moment awareness” [107] of modifications, but rather as a tool for tracking modifications in a shared document. As commented by one participant, “I focused on doing my part so I did not really look at MD. I saw a red flag few times, but did not really care. I remember that I clicked on
MD few times just to see quickly what Tarzan\textsuperscript{34} had changed, but I did not take his changes seriously”.

In the study, one pair of participants adopted the writing and editing approach. That is, one person was responsible for writing the draft of a document (that is, the writer) while the other person edited the article (that is, the editor). Hence, there were a lot of modifications shown in MD, but the writer did not really pay much attention to those changes because he knew that the editor was supposed to edit his text. This indicates that the usefulness of MD is also dependent upon participants’ approach of performing SCA tasks.

Five properties of modification awareness

There were diverse responses from the participants regarding the usefulness of the five properties of modification awareness that are currently supported by MD, as seen in Table 6.5.

<table>
<thead>
<tr>
<th>Questions</th>
<th>Mean (n=16)</th>
<th>Std. dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Showing the page number at which a modification occurred is helpful (i.e., Modification Context)</td>
<td>3.81</td>
<td>0.91</td>
</tr>
<tr>
<td>Showing part of a modification helps me know what had been changed (i.e., Modification Body)</td>
<td>4.94</td>
<td>1.00</td>
</tr>
<tr>
<td>Showing the name of a person who made a modification is useful (i.e., Modification Creator)</td>
<td>3.25</td>
<td>1.00</td>
</tr>
<tr>
<td>Displaying elapsed time of a modification helps me know when the modification was made (i.e., Modification Time)</td>
<td>3.94</td>
<td>0.93</td>
</tr>
<tr>
<td>Showing the status of a modification, such as single deletion and multiple deletions, is useful to keep updated with the changes (i.e., Modification Status)</td>
<td>3.69</td>
<td>1.01</td>
</tr>
<tr>
<td>Fading colours to indicate the age of a modification is effective.</td>
<td>4.25</td>
<td>1.13</td>
</tr>
<tr>
<td>Information shown in MD is easy to understand.</td>
<td>5.56</td>
<td>0.89</td>
</tr>
<tr>
<td>Information shown in MD is useful to stay aware of the changes in a document.</td>
<td>4.38</td>
<td>0.89</td>
</tr>
</tbody>
</table>

Table 6.5: Participants’ responses to the usefulness of MD. Scores range from 1 = “strongly disagree” to 7 = “strongly agree”.

First, MD supports the modification context by retaining situated presentation of the anchor text which is shown within a Word document. In addition, MD adds a separate and symbolic representation of the modification location. That is, showing the page

\textsuperscript{34} Participants’ names are altered to protect their identities.
number at which a modification was made in a separate window outside the document, this awareness information was rated 3.81 (std. dev. = 0.91) by participants. A substantial proportion of the participants either were unable to decide if showing the page number is useful (nearly 40%) or believed that it is not useful (nearly 45%). Participants argued that it is often hard for them to recall what part of a document is on a particular page. Therefore, knowing the page number is not really useful. All too often, participants clicked on a modification item to allocate the anchor text of a modification within the document.

Second, nearly 70% of the participants found the technique of showing partial modification body useful (mean = 4.94, std. dev. = 1.00; n = 16), because it conveys brief information about what had been modified. If participants were keen to view the full details of a modification, they can click on the modification. One participant stated, “Tarzan often corrected my typos. And, MD helps me pick up those changes easily. However, in some cases, he removed the entire paragraphs of mine, thus I wanted to view in full what had been deleted”.

Third, most participants (more than 60%) believed that showing the name of a person who made a modification was not useful for their knowledge of a modification (mean = 3.25, std. dev. = 1.00; n = 16). However, this result may be influenced by the fact that participants were working in pairs, thus they knew immediately who made a modification.

Fourth, as reported earlier, MD was not used to maintain up-to-the-moment modification awareness, thus displaying elapsed time of a modification was not found useful (mean = 3.94, std. dev. = 0.93; n = 16). Participants were more interested in if there were any new modifications rather than when they were made. Furthermore, showing elapsed time of a modification of which participants were already aware is even unnecessary. As commented by one participant, “It makes no difference for me if it [a modification] was made 10 minutes or 30 minutes ago”. However, participants responded that it is helpful that MD sorts all modifications in temporal order, as committed by one participant, “As long as the software [MD] sorts the changes in order of their occurrences that is enough. It is not necessary to display the actual time”. In addition, the transition of a background colour used to show the age of a modification was found quite effective in distinguishing new modifications from existing ones.
“Highlighting a new [modification] item in red caught my attention. Often, I looked at MD whenever there is a red flag”, said one participant.

Finally, the study shows that it is necessary to provide information about the modification status. However, the current design of MD does not present the full history of a modification (that is, the full details of the life cycle of a modification). For example, if a modification consists of several iterative changes, users should be able to view information about those changes. One comment by a participant was “When clicking on a multiple deletion item in MD, I want to see the changes that were made earlier to the item”. Because of this shortcoming of MD, participants’ rating on MD’s support for the modification status was low (mean = 3.69, std. dev. = 1.01; n = 16).

### 6.5.2.3 Usefulness of a dual monitor setting

In the study, dual monitors were used to display two types of information. A primary monitor was used to show the main document, which is considered as primary information. A secondary monitor was used to show awareness mechanisms, including ERV and MD, which are considered as secondary information [107]. Participants’ responses to the dual monitor setting are shown in Table 6.6.

<table>
<thead>
<tr>
<th>Questions</th>
<th>Mean (n=16)</th>
<th>Std. dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keeping track of windows on two monitors is difficult.</td>
<td>3.38</td>
<td>1.15</td>
</tr>
<tr>
<td>It is easy to switch views between two monitors.</td>
<td>4.38</td>
<td>1.09</td>
</tr>
<tr>
<td>The dual monitor is useful in separating the window of a main document and the windows of ERV and MD.</td>
<td>4.94</td>
<td>0.93</td>
</tr>
<tr>
<td>Preferring the dual-monitor setting to the one-monitor setting</td>
<td>5.56</td>
<td>0.84</td>
</tr>
</tbody>
</table>

Table 6.6: Participants' responses to the dual monitor setting. Scores range from 1 = “strongly disagree” to 7 = “strongly agree”.

First, in general, participants did not have difficulty in tracking views between two monitors. The mean of participants’ responses to the statement “Keeping track of windows on two monitors is difficult” is 3.38 (std. dev. = 1.15; n = 16). This means that overall participants disagreed with the statement. In addition, participants found it relatively easy to switch views between two monitors (mean = 4.38, std. dev. = 1.09; n = 16). The display of ERV is very self-contained in the sense that both the over-the-shoulder view and the telecarets-eye view are connected to the radar view, therefore participants were not required to make an abrupt contextual shift between the main...
window and ERV. Regarding MD, after participants clicked on a modification item in MD, they needed to shift their attention to the main document to see the full content and the exact location of the modification. However, this attention shifting caused by MD was not found inconvenient by participants, as stated by one participant “When only one monitor was used, I had to change view from your window [MD] to the Word document anyway. So, two monitors make no different”.

Second, to verify the usefulness of dual monitors, participants were asked to choose their preferred setting (for example, “Do you prefer the one monitor setting or dual monitor setting?” with the scores ranging from 1 as “one monitor” to 7 as “dual monitors”). Consistent with the study of computer games reported earlier in Section 3.4, the dual monitor setting was preferred to the single monitor setting (mean = 5.56; std. dev. = 0.84; n = 16). Participants’ justification for their preference of dual monitors was mostly because the dual monitor setting offers more screen real estate. As stated by one participant, “I like two monitors more because they give me more space. Having all windows on one monitor is too crowded. I found less distracted when other windows [ERV and MD] are on the second monitor.”

It might be argued that a single monitor that is as large as dual monitors could be equally effective. The answer is “yes” provided that a single monitor has a wide-screen dimension like a dual-monitor screen, and a single monitor supports maximising a window to only half the size of the entire screen, which is similar to local maximisation currently supported by dual monitors. This feature is very important as it saves participants time resizing and aligning positions of different windows.

6.5.3 Discussion of CoWord+ evaluation

In addition to testing the usefulness of ERV, MD and dual monitors, the evaluation was used to identify strengths and weaknesses as well as possible improvement to some usability issues of the mechanisms.

6.5.3.1 Improving ERV

First, there is a visibility problem in the current design of ERV when two over-the-shoulder views or two telecarets-eye views overlap. Users can set different zoom values for their views of a shared document. As a result, viewports are not guaranteed to be aligned when overlapping.
Second, the radar view in ERV is implemented based on the Word APIs, which limits the minimum zoom value to 10% of a Word document. Thus, if a Word document comprises many pages, the radar view of ERV is not able to capture the entire view of the document on the screen. Consequently, users are required to scroll up and down to navigate the radar view.

6.5.3.2 Improving MD
First, as reported in Section 6.5.2.2, MD was often found to be used at the end of a collaborative session as a tool for reviewing changes in a document rather than keeping up-to-the-moment awareness of changes. Thus, it is useful to distinguish between modification items, which have already been viewed by users (for example, after they click on those items on MD), and those that have not been viewed by users. For example, colours can be used for this purpose, such as coding visited changes in one colour and unvisited changes in another [205].

Second, MD currently displays separate placement of awareness information. That is, awareness information is situated outside a shared Word document in a separate window. Adopting separate placement helps MD overcome the visibility problem of situated placement, as discussed in Section 6.3.2.2. However, the evaluation of MD shows that the current design of separate placement in MD is weak in supporting the modification context. Hence, participants often clicked on a modification item on MD to allocate the anchor text within the document (that is, inside the workspace). Thus, the challenge posed to the design of MD is how to design the separate-placement display that can convey the modification context effectively.

Third, the evaluation shows that MD provides limited information about the modification status, especially a history of changes that has been made to a modification. MD should be extended so that it shows both the current status of a modification and a history of changes occurring in the life cycle of a modification. One possible solution to this issue could be replacing the current sequential list in MD with a tree-based layout. Each modification item is presented as one “thread” under which contains all revisions made to the modification.
6.6 Reflection on the use of F@

The previous few sections have reported the design, implementation and evaluation of ERV and MD. These mechanisms were designed to support various aspects of workspace awareness (for example, awareness of working and viewing areas and awareness of modifications). This section presents a comparative analysis of workspace awareness, between the principles covered in F@ and the features supported by ERV and MD. The analysis reflects both the abstract level and the concrete level.

6.6.1 Reflection on the abstract level

Table 6.7 presents a comparative analysis between elements of workspace awareness addressed at the abstract level of F@ and those supported by ERV and MD. As emphasised in Section 4.3.2, the need of supporting the 5W1H dimensions varies from situation to situation, depending on the group context and the nature of cooperative tasks. Hence, it is not a goal of ERV and MD to support all elements of the 5W1H dimensions. The comparison shows that many awareness elements of the F@ framework are supported by ERV and MD. In addition, those awareness elements were found useful in enhancing workspace awareness as reported in the evaluation of ERV and MD. Some awareness aspects of F@ are not supported by ERV and MD. This phenomenon is reasonably understandable because while the framework covers a wide variety of awareness elements, ERV and MD are designed to support several aspects of workspace awareness in the context of SCA.

<table>
<thead>
<tr>
<th>Awareness elements</th>
<th>ERV and MD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presence (current)</td>
<td>ERV shows users viewports, over-the-shoulder view, and telecarets view. The existence of one of these three views indicates the presence of users.</td>
</tr>
<tr>
<td>Authorship (current)</td>
<td>MD displays the name of a creator who made a modification.</td>
</tr>
</tbody>
</table>

The “Who” dimension

<table>
<thead>
<tr>
<th>Awareness elements</th>
<th>ERV and MD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action (current)</td>
<td>ERV allows a local user to see a text selection that is highlighted by a remote user.</td>
</tr>
</tbody>
</table>

The “What” dimension


<table>
<thead>
<tr>
<th>Awareness elements</th>
<th>ERV and MD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location (current)</td>
<td>ERV provides the telecarets-eye view to show where local and remote users are working.</td>
</tr>
<tr>
<td>Gaze/View/Reach (current)</td>
<td>ERV provides the over-the-shoulder view to indicate where local and remote users are viewing in a shared document. This view also reflects the range within which users can reach and gaze.</td>
</tr>
</tbody>
</table>

The “Where” dimension

<table>
<thead>
<tr>
<th>Awareness elements</th>
<th>ERV and MD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event (past)</td>
<td>MD provides temporal properties of a modification to show when the modification was made.</td>
</tr>
<tr>
<td>Action (past)</td>
<td>MD includes information about the nature of a modification such as single deletion or multiple deletions.</td>
</tr>
<tr>
<td>Location (past)</td>
<td>MD provides information about a location in a shared document where a modification occurred.</td>
</tr>
</tbody>
</table>

The “When” dimension

<table>
<thead>
<tr>
<th>Awareness elements</th>
<th>ERV and MD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motivation (past/current)</td>
<td>MD can also include an annotation that remote users made when they modified a local user’s text. This can be used to justify users’ motivation.</td>
</tr>
</tbody>
</table>

The “Why” dimension

<table>
<thead>
<tr>
<th>Awareness elements</th>
<th>ERV and MD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action (past)</td>
<td>MD shows information about how an artefact was changed (for example, deleted).</td>
</tr>
</tbody>
</table>

The “How” dimension

Table 6.7: A comparative analysis between dimensions of workspace awareness and awareness support of ERV and MD.

6.6.2 Reflection on the concrete level

The concrete level presents the formalisation of fundamental awareness aspects, including workspace awareness. This section examines how temporal logic formulas of workspace awareness (for example, presence awareness of group members, awareness of artefacts and awareness of actions) are reflected in the designs of ERV and MD.

Presence awareness of group members

As analysed in Section 4.4.2, presence awareness of group members involves users’ perception of the presence of past, current and future group members. Past group members refer to those who were in a group and had already left. Current group members are those who are currently participating in a group. Future group members
Chapter 6 Designing Awareness Support for Collaborative Authoring

refer to people who are going to join a group. Presence awareness of these group members is formulated as:

\[
M(s, \mathcal{S}) \models \text{aware}(\text{user}, \text{presence\_of\_members}) \Leftarrow
\]

\[
(M(s, \mathcal{S}) \models (\forall p_i: p \neq \text{user}) \part\_\_(p_i, \text{Group}) \land \text{know}(\text{user}, p_i)) \land
\]

\[
(M(s, \mathcal{S}) \models (\forall p_j: p \neq \text{user}) \part\_\_(p_j, \text{Group}) \land \text{know}(\text{user}, p_j)) \land
\]

\[
(M(s, \mathcal{S}) \models (\forall p_k: p \neq \text{user}) \part\_\_(p_k, \text{Group}) \land \text{know}(\text{user}, p_k)),
\]

where \( p_i, p_j, p_k, \text{user} \in \text{Group}.\)

Section 5.8.2 analyses reflection of the design of RIM on supporting presence awareness of conversants. A similar interface can be designed to support presence awareness of group members, as seen in Figure 6.17.

![Figure 6.17: Presence awareness of past, current, future group members (mock up).](image)

**Awareness of artefacts**

In SCA, mechanisms, such as the radar view are able to show the current positions of shared artefacts in the workspace, but not their past positions. F@ emphasises a need for providing information about the past and current positions of artefacts:

\[
M(s_i, \mathcal{S}) \models \text{aware}(\text{user}, \text{position\_of\_artefact}) \Leftarrow
\]

\[
(M(s_i, \mathcal{S}) \models \text{position}(\text{artefact}, x_i) \land
\]

\[
M(s_i, \mathcal{S}) \models \bullet_{-j} \text{position}(\text{artefact}, x_j) \land
\]

\[
\text{know}(\text{user}, x_i) \land \text{know}(\text{user}, x_j),
\]

where \( 0 \leq j \leq i, \text{artefact} \in \text{Artefact}.\)

In order to support this form of workspace awareness, groupware systems can develop visualisation techniques, such as a local replay mechanism proposed in Multimedia.
Lecture Board [247] to view the current position of an artefact as well as being able to play back to a certain point in the past.

As discussed in Section 6.5.3.2, a tree-based layout can be incorporated into the design of MD to show information about the current and past changes that have been made to document artefacts such as text, diagrams, etc., as depicted in Figure 6.18.

![Figure 6.18: Support for awareness of artefacts (mock up).](image)

**Awareness of actions**

Popular awareness mechanisms like radar views and multi-user scrollbars often show the locations at which users are currently viewing, and fail to distinguish between users’ viewing and working areas. As mentioned in Section 4.4.2, groupware systems should separate different activities, and provide values of the corresponding locations at which activities occur.

\[
M(s, S) \models aware(user, action) \iff \\
M(s, S) \models (\forall p \in People : \alpha \in Artefact) A^{\alpha}_{action}(p, \alpha) = \Omega \land know(user, \Omega)
\]

ERV was designed to address the issue by providing simultaneous views of users’ working and viewing areas in a shared document, as seen in Figure 6.8.

In addition to the dual views provided by ERV, MD also provides details of users’ actions (that is, modification). For example, MD informs users of the location at which a modification occurs in a document (for example, page number), the type of a modification (for example, single deletion), and so on.
6.7 Summary

This chapter presents a case study of designing, implementing and evaluating mechanisms to support workspace awareness in synchronous collaborative authoring (SCA). To begin with, the chapter examines major solutions that have been used by current SCA systems to support conversational awareness, workspace awareness, contextual awareness and self-awareness.

Second, the chapter describes two awareness mechanisms—**Extended Radar View** (ERV) and **Modification Director** (MD)—that were designed to enhance support for workspace awareness in SCA. ERV extends the conventional radar view to enhance workspace awareness support. ERV includes two additional views—the telecarets-eye view and over-the-shoulder view—to the radar view to show both users’ working and viewing areas, respectively. MD improves support for workspace awareness by tracking modifications in a shared document, in particular modifications made by remote users. MD helps users interpret and understand modifications by providing information about the lifecycle of a modification (for example, time when a modification occurred, the nature of a modification, its location in a document, etc.). ERV and MD were developed using the **transparent adaptation** approach. This approach allows implementing the mechanisms without accessing the source code of MS Word. ERV and MD were evaluated in a laboratory-based experiment. The evaluation shows that these two mechanisms help users maintain workspace awareness and meet their design objectives. The evaluation also yielded some usability concerns that need to be addressed in further development of the two mechanisms.

Finally, this chapter presents a comparative analysis between the principles of workspace awareness that is covered in F@, and the design of ERV and MD. Overall, many aspects of workspace awareness addressed by F@ are implemented in ERV and MD, and they are found useful in maintaining workspace awareness as confirmed from the evaluation of ERV and MD.

The previous chapter and this chapter have reported two case studies of developing mechanisms to support group awareness in Instant Messaging and collaborative authoring. The next chapter examines possible applications of F@ to the design of awareness mechanisms in other synchronous distributed groupware systems, including
systems for collaborative drawing, electronic meetings, collaborative programming and multi-player computer games.
Chapter 7

Discussion

7.1 Introduction
The last two chapters have presented two case studies of developing awareness support for Instant Messaging and synchronous collaborative authoring. In particular, Chapter 5 examines the design, implementation and evaluation of two innovative mechanisms, Relaxed Instant Messenger (RIM) and Conversation Dock (ConDock), which are used to enhance conversational awareness in text-based communication. Chapter 6 presents the design, implementation and evaluation of Extended Radar View (ERV) and Modification Director (MD), which were developed to improve workspace awareness in collaborative authoring. Chapter 5 and Chapter 6 also present the reflections on the use of F@ in the design of these resulting mechanisms.

This chapter examines briefly the support for group awareness in some other synchronous groupware systems, including systems for collaborative drawing, electronic meetings, collaborative programming, and multi-player computer games. This discussion also involves some mechanisms proposed to support awareness in these groupware systems. These proposed mechanisms are derived from the interpretive analysis of the F@ framework and experience learnt from the development of mechanisms discussed in Chapter 5 and Chapter 6.

It is important to note that the interpretation of F@ presented in this chapter does not endeavour to cover all possible enhancements for awareness support, rather it examines fundamental aspects of group awareness in these specific domains. Although mechanisms proposed in this chapter have been neither implemented nor evaluated, more importantly they illustrate potential applications of F@.

This chapter is structured as follows. Section 7.2 briefly reviews awareness support in four above mentioned groupware systems, and considers the potential of applying F@
to deriving awareness support for these systems. Section 7.3 presents a general discussion on F@.

7.2 Discussion of awareness support in other synchronous groupware systems

7.2.1 Collaborative drawing

Synchronous collaborative drawing systems (SCDS), also referred to as “shared whiteboards”, allow a group of users to view and draw simultaneously on a shared surface over distance. There are two main categories of SCDS: video-based systems and computer-based systems. On the one hand, video-based SCDS allow users to draw objects onto physical boards. Images of the boards are captured by video cameras and synchronised at remote sites. Examples of video-based SCDS include ClearBoard [126] and VideoDraw [219]. On the other hand, users of computer-based SCDS are able to use the mouse and keyboard to draw objects on a shared virtual drawing. Representative examples of computer-based SCDS include Conversation Board [31], Digital Lecture Board [252], GroupDraw [96], GRACE [212] and GroupSketch [96].

SCDS support both informal collaboration, such as brainstorming and sketching an initial design, and formal collaboration, such as structured graphic design, publishing and engineering. Like other groupware systems, supporting group awareness has been considered as an important aspect of SCDS [126, 219]. Group awareness is able to facilitate the collaborative drawing process by supporting communication, enhancing coordination and aiding conventions. Several mechanisms have been developed to support group awareness in SCDS, as briefly reviewed below.

First, text-based chat tools (discussed in Chapter 5 and Section 6.2.1) can be used to support direct communication between users of computer-based SCDS. As an example of this, N-ABLE™ [138] integrates a chat tool and a shared whiteboard to support awareness.

Second, audio and video-mediated systems can be used to support communication in SCDS. For example, VideoDraw utilises audio links to support conversational awareness. ClearBoard uses high-fidelity video images to convey information about users’ gestures (for example, pointing to things on the whiteboard) and eye contact. sTeam [111] supports embedding video objects into the shared whiteboard.
Chapter 7 Discussion

Third, relaxed-WYSIWIS mechanisms, such as telepointers, viewports, the radar view and distortion-oriented view (discussed in Section 2.4.2.2), can be used to support awareness of other users’ presence and activity in the shared drawing. To name a few, GroupDraw [96] provides telepointers, GroupDesign [13] displays viewports, and Ideogramic UML [114] supports the radar view.

Finally, the record-and-replay technique is also developed to support awareness of users’ actions (for example, what did users draw in the past?). For example, Multimedia Lecture Board [247] provides the record-and-replay mechanism that allows users to record a history of local or shared actions and replay them locally.

The F@ framework can be applied to enhance the support for conversational awareness and workspace awareness in SCDS, as discussed below.

First, F@ emphasises a need for supporting conversational awareness in communication between members of a group. Conversational awareness helps people stay updated of what is going on in a conversation (for example, who is talking, what is the latest message, etc.). To support conversational awareness in SCDS, mechanisms, such as Track View (Section 5.7.1.1) and RIM (Section 5.4.1), can be used. Track View is able to convey awareness information about listening-in-progress in video-based drawing systems. RIM can be integrated into computer-based SCDS to support deictic reference and increase coherence between shared drawing objects and a conversation. As an example of this, RIM can be extended to support anchored text-messages that are embedded into a shared drawing to support in-context conversation.

Second, F@ addresses a need for supporting workspace awareness, which involves information about users’ presence, actions and shared artefacts. Variants of Buddy View (Section 5.8.2) can be integrated into SCDS to support past/current/future presence awareness. Current techniques, such as the record-and-play mechanism, allow users to view a sequence of actions occurring in the shared drawing (that is, an action-centric approach). Given the current design, it seems to be difficult for users to view a history of actions and modifications performed upon a specific artefact. In this action-centric approach users need to go through a list of actions to select relevant actions applied to a specific artefact. The record-and-replay mechanism can be tailored in such a way so that it is able to show only actions that were performed upon a particular artefact (that is, an
artefact-centric recording approach). Furthermore, users of SCDS should be able to customise their recording preferences between an action-centric mode and an artefact-centric mode.

7.2.2 Electronic meetings

Electronic meeting systems (EMS) allow a group to conduct meetings using networked computers and associated collaborative technologies, such as text chat, audio, video, presentation, document management, group calendar and bulletin boards [162, 172, 178, 186, 261]. Examples of EMS include GroupSystems [101], Lotus Notes [122], Microsoft NetMeeting [155], Liveboard [74], Logan [187] and so on. EMS are used to support both informal and formal meetings in a wide range of domains, including online education [262], military planning [118], accounting [5] and application development [51].

In general, an electronic meeting involves three phases, including pre-meeting, during-meeting (or in-meeting) and post-meeting [69]. In the pre-meeting phase, a group facilitator is responsible for scheduling the availability of group members, acquiring necessary resources for a meeting, and preparing and distributing the meeting agenda. During the meeting, it is often that the facilitator steers the meeting using the meeting agenda, providing group members access to collaborative tools (for example, brainstorming, voting, audio, video, etc.) and ensuring that group members are informed of the meeting process. In the post-meeting phase, the facilitator needs to compile minutes of the meeting, and ensures that group members act on tasks scheduled in the meeting. In the pre-meeting and post-meeting phases, most of the interaction between the facilitator and group members often happens asynchronously. Thus, this research is interested in examining awareness support for the during-meeting phase, where synchronous interaction occurs.

In the during-meeting phase, a group can be engaged in different cooperative activities, such as brainstorming, discussing, voting, and so on. Thus, mechanisms to provide group awareness in EMS should support specific collaborative technology, such as awareness mechanisms for brainstorming, awareness mechanisms for audio and video communication, awareness mechanisms for shared whiteboard and so on.
F@ presents four awareness schools, including conversational awareness, workspace awareness, contextual awareness and self-awareness. In distributed electronic meetings, in particular the during-meeting phase, conversational awareness and workspace awareness are the most relevant compared to the other two awareness schools.

Regarding support for conversational awareness, whether a communication technique used in an electronic meeting is text-, audio- or video-based, it is important that participants of the meeting need to be provided with information about the conversation, such as who are attending the meeting, who is talking, what they are talking about, and so on. Mechanisms for conversational awareness in text-based chat have been comprehensively discussed in Chapter 5. Mechanisms to support awareness in audio- and video-mediated communication (that is, mediaspaces) are examined in Section 2.4.1.2. Those mechanisms are highly applicable to enhancing conversational awareness in electronic meetings.

Workspace awareness plays an important role in facilitating electronic meetings. Members of a group need to know activity and progress of other members. In addition, they need to be informed about the state of shared artefacts in the meetings. For example, if a group performs a brainstorming task, users of EMS need to maintain knowledge of what problem the group tries to solve, what ideas other people have proposed, who contributes what, and so forth.

Additionally, it should be noted that a need for awareness support in EMS varies depending on the culture of a group, the context of the meeting and task activities. An influential study of EMS presented by Nunamaker et al. [172] showed that the processes and outcomes of electronic meetings depend on four characteristics of context, group, task and EMS. For example, in groups where power and status differences are high (for example, more than two management levels), anonymity appears to have positive effects on meeting outcomes. In such situations, support for presence awareness is less important and probably unnecessary, as opposed to other situations where conformance pressure is low (for example, equal power and status).

7.2.3 Collaborative programming

Recently, there has been a rising interest in synchronous collaborative programming which involves more than one software developer working on the same code at the same
time. A representative example of this is *pair programming* [14, 119]. Pair programming is probably the most extensively investigated practice of *eXtreme Programming* [48] that is designed to produce high quality code in a light-weight fashion. This section considers possible design of awareness support for collaborative programming, using pair programming as an example.

In current practice of pair programming, when two developers are situated at the same location, they work side-by-side on a single computer to produce software artefacts, such as code, algorithms, diagrams and test cases. Under such circumstances of close proximity, it is easy and natural for developers to acquire awareness information about what the other person is doing. When developers are geographically distributed, they use either shared applications, such as NetMeeting [155], or synchronous collaborative programming systems, such as COPPER (COllaborative Pair Programming EditoR) [166], to perform a joint programming activity. It is important for collaborative programming tools to provide sufficient awareness information about other developers and their activities [109, 166].

F@ addresses a need for supporting workspace awareness. In the context of pair programming, coupling between software developers is often tight. As a result, it is important for them to receive up-to-the-moment information about the other person’s interaction with shared programming artefacts. Mechanisms of workspace awareness support discussed in Chapter 6, such as telepointers, the radar view, multi-user scrollbars, action feedback and process feedthrough, can be used to convey awareness information about users’ presence and actions in the shared workspace. Furthermore, it could be the case that distributed developers attend to different source files. Thus, it would be useful to provide presence information about the files on which developers are working.

In addition, F@ indicates providing awareness information about changes made to shared artefacts. Monitoring changes in pair programming is critical because inappropriate and unexpected changes to shared software artefacts, even small changes (such as deleting the definition of functions or variables) could affect the group’s work severely. Hence, developing mechanisms to keep users informed of changes is essential. Section 6.3.2.1 describes five properties of modification awareness, including information about *content, body, creator, time* and *status* of modification. It would be
useful for collaborative programming systems to support these properties in order to maintain developers’ awareness of the state of shared software artefacts.

### 7.2.4 Multi-player computer games

Computer games have been for a long time considered as an entertaining tool and this recognition still applies today. Moreover, popular computer games now often involve interaction of multiple players simultaneously. People play in teams in these games. They are required to interact and coordinate in different ways in order to achieve game missions. For that reason, multi-player computer games are also considered as collaborative tools. Like other groupware systems, supporting group awareness holds the promise in improving coordination of players in games.

Section 3.4 reports an empirical study of multi-player computer games, involving Counter-Strike (a first-person-shooter game) and WarCraft III (a strategy game). This observational study yields insights into enhancements of awareness support in these games. Out of the four awareness schools of conversational awareness, workspace awareness, contextual awareness and self-awareness addressed by F@, this section discusses to what extent the support for conversational awareness, workspace awareness and self-awareness can be improved.

First, support for conversational awareness involves assisting deictic reference. The empirical study shows that players of both Counter-Strike and WarCraft III sometimes found it difficult to match game artefacts to those referred to in a conversation and artefacts in a game (that is, the issue of deictic reference). Simple mechanisms, such as telepointers and anchored text messages, can be added to the interface of these games to address this problem.

Second, F@ presents workspace awareness, particularly awareness of users’ presence. Presence awareness involves users’ past, current and (possible) future presence. In first-person-shooter games, such as Counter-Strike, only information about players’ current positions is provided. These games could be enhanced to convey some level of awareness of past presence of players. For example, when players are moving in a game, the radar view could provide a short trail showing the direction of players’ movement. This moving trail provides information about both players’ current positions and their previous positions. This information could be useful in coordinating a game.
Another alternative to supporting past and current presence of players is to implement immersive audio [28], which reflects realistic positions of players as they speak in game. For example, immersive voice allows determining whether other remote players are at the right or the left, and whether they move toward to or further away from the local player. Additionally, games should allow players to be able to specify and tailor their future positions by for example, marking locations to which they would move and other players are able to see these marked locations. Furthermore, F@ also emphasises a need for providing information about players’ actions. In the case of first-person-shooter games, information about players’ actions could be enhanced to accommodate tightly-coupled coordination. As observed from the empirical study, players found detailed views of other players’ actions more useful than low-fidelity displays shown in the radar view when tight coupling was required.

Third, F@ emphasises a need for supporting self-awareness. In the context of computer games, it is important that players are provided with feedback of their own presence and actions in a game. The empirical study of computer games shows that players sometimes found it difficult to relate their own positions in comparison to other players’ positions. Mechanisms such as a moving trail (discussed two paragraphs above) can be used to convey self-awareness information about players’ own movements in a game. Showing feedback of players’ own movements can be useful in helping players identify their whereabouts in a game (for example, their position and moving direction).

7.3 Discussion of F@

The previous section has presented an interpretive analysis of applying F@ to designing awareness support in groupware systems for collaborative drawing, electronic meetings, collaborative programming, and multi-player computer games. This section discusses F@. More specifically, first the following discussion focuses particularly on the questions of to what extent F@ provides an understanding of the concept of group awareness, and to what extent F@ can be applied to the design of supporting mechanisms. The discussion also examines briefly the applicability of F@ to other types of groupware. Second, this section examines possible limits of F@ and discusses potential enhancements to F@. The discussion is derived from lessons learnt from the development of awareness mechanisms reported in Chapter 5 and Chapter 6, and from the interpretive analysis presented in the previous section.
7.3.1 Analytical reflection

First, to extend groupware developers’ understanding of group awareness, F@ presents a comprehensive description of group awareness (that is, the abstract level). In particular, F@ addresses four sub-types of group awareness, which are referred to as awareness schools, including conversational awareness, workspace awareness, contextual awareness and self-awareness. F@ describes the concept and properties of these four awareness schools. The description shows a set of group components, such as people, artefacts, and actions, which are involved in forming these awareness schools. Furthermore, the description considers a set of fundamental awareness elements that contribute to maintaining these awareness schools. These elements are categorised into six dimensions of “Who”, “What”, “Where”, “When”, “Why” and “How”. Identifying these awareness elements is useful because it helps developers understand the concept of awareness by considering a concrete set of specific information.

It is important to note that a set of awareness elements identified in depicting awareness schools is generic. This indicates that awareness elements should be interpreted and understood based on a specific domain in which a particular form of cooperative activity is considered. For example, the Presence element (see Table 4.6) refers to information about “Who is in the workspace”. The scope of a shared workspace could vary across different groupware systems. In collaborative authoring, a shared workspace is an electronic document which users are editing simultaneously. However, in distributed electronic meetings, a shared workspace might include a number of electronic spaces such as a shared electronic document, a shared whiteboard and mediaspaces. Therefore, it is important to ground the elements of group awareness in a specific context in which the elements are considered.

In addition to identifying a specific set of descriptions of awareness schools, F@ presents formulation of requirements of supporting group awareness (that is, the concrete level). The purpose of presenting formal requirements is to provide developers with a precise and concise description of what information should be provided. In particular, the concrete level of F@ utilises temporal logic to describe fundamental aspects of group awareness. It has been emphasised at the beginning of Section 4.4 that the concrete level does not aim to formalise every awareness element that is identified.
in the abstract level. Rather the concrete level presents formalisation of fundamental
time-related awareness aspects.

To further this discussion of how F@ can be used to understand group awareness, the
author discusses briefly how F@ is used in the development of awareness mechanisms
for Instant Messaging (Chapter 5) and collaborative authoring (Chapter 6). F@ is useful
in providing the concept of conversational awareness discussed in Chapter 5. In
particular, F@ allows analytical comparison and reflection between the features
provided by mechanisms and principles covered by F@. The reflection is useful in
showing what has and what has not been supported by the mechanisms.

Second, to what extent does F@ support designing awareness mechanism? F@ supports
the design of mechanisms by showing a type of information that needs to be provided. It
is not an aim of F@ to specify exactly which visualisation technique should be used.
This is because a visualisation technique is dependent of the nature of groupware
systems and the form of cooperative tasks that they support. The same set of
information can be presented differently in different groupware systems.

As seen in Chapter 5 and Chapter 6, the exact graphical presentation of awareness
information is up to the groupware designers to choose for their situations. In the
development of awareness mechanisms for Instant Messaging and collaborative
authoring, the first task is to identify awareness information that the groupware systems
need to support. Determining requirements is highly important and should not be bound
to any specific visualisation technique. In other words, the first task of designing
awareness mechanisms is to identify what awareness information needs to be supported
rather than how to present awareness information. Once the requirements and needs are
identified, developers then consider which available visualisation technique is
appropriate in terms of whether the technique is able to represent the information, and
whether the technique is suitable for the user interfaces and interactive styles of
groupware systems under consideration.

Furthermore, mechanisms in Chapter 5 and Chapter 6 are implemented using different
approaches. For example, RIM was developed from scratch and awareness mechanisms
were clearly defined as one component of the RIM system architecture. On the other
hand, ConDock, ERV and MD were implemented as plug-ins for third-party groupware
systems, including MSN Messenger and Microsoft Word. As a result, there is a need for developing an effective and transparent approach that allows integrating awareness mechanisms with existing groupware systems in a light-weight manner. Though not being covered by the current version of F@, F@ could be enhanced to address different mechanical approaches that can be used to implement awareness mechanisms.

Up to this point, this section has examined the use of F@ in extending groupware developers’ understanding of group awareness and in helping the developers design awareness mechanisms. It is important to note that F@ is developed to examine the concept of group awareness in the context of distributed synchronous groupware systems. As a result, F@ might not be directly applicable to research on group awareness in asynchronous groupware systems, such as emails, blogs, version control systems and bulletin boards. Having said that, synchronous systems and asynchronous systems share many common awareness elements, for example, they both involve six dimension of “Who”, “What”, “Where”, “When”, “Why” and “How”. Support for group awareness in synchronous and asynchronous groupware systems could involve the deviating emphases on the importance of particular awareness elements. For example, in synchronous systems knowing who is currently participating in a shared workspace is important, but in asynchronous systems this awareness information could be less important because users often interact with the systems at different times.

7.3.2 Critique
This section briefly examines some limitations of F@ and discusses possible enhancements that can be added to F@.

First, the concrete level uses temporal logic to formulate fundamental time-related aspects of group awareness. The concrete level could be extended to address other aspects of group awareness that are not related to time, such as modelling context of actions, reasoning about users’ behaviour and even predicting users’ behaviour.

Second, the present F@ framework does not include specific visualisation techniques that can be used to display particular types of information. The exact graphical presentation of awareness information is left up to the groupware designers to choose based on the nature of their groupware systems. Further investigation of relations between representation of awareness information and the type of groupware systems.
can be conducted to enhance F@. Moreover, F@ could also be extended to address the relations between cooperative tasks and visualisation technique used to support task-related awareness information.

Third, F@ currently does not include guidelines and analytical descriptions of mechanical approaches that can be used to implement awareness mechanisms. F@ can be extended to provide more detailed descriptions of mechanical approaches, such as in-house, toolkit-based, and transparent adaptation methods [237], and suggesting an appropriate strategy of using these methods to develop awareness mechanisms.

Finally, F@ was developed to support group awareness in small groups. The empirical studies and case studies reported in this thesis involved small groups of participants. Thus, the results of the studies might not be consistent with those carried out with a large group of users. Therefore, it might require a further extension in order to apply F@ to other groupware systems for a large number of users (for example, peer-to-peer file sharing systems and Massively Multiplayer Online Role-Playing Game).

7.4 Summary

This chapter presents discussion of group awareness support in synchronous groupware systems for collaborative drawing, electronic meetings, collaborative programming and multi-player computer games. The discussion involves the interpretive analysis of the use of F@ in designing potential mechanisms to enhance group awareness in these systems.

This chapter also discusses F@ in terms of helping groupware developers understand the concept of group awareness and design corresponding awareness mechanisms. The discussion of F@ is based on the development of awareness mechanisms in Chapter 5 and Chapter 6, and on the interpretive analysis in Section 7.2. The discussion shows that F@ is useful in extending developers’ understanding of group awareness and in helping them design mechanisms. The discussion also points to possible enhancements to F@.

In particular, F@ can be extended to address visualisation techniques that are used to represent awareness information and mechanical approaches that are used to implement mechanisms. Additionally, the discussion considers the applicability of F@ to providing support for group awareness in other types of groupware.
The next chapter concludes this thesis by revisiting objectives and contributions of this research and presenting a view of future research and development.
Chapter 8

Conclusions and Future Work

8.1 Summary of the thesis
This research has investigated the issue of supporting group awareness in synchronous distributed groupware systems. The research is motivated by the importance of group awareness in supporting cooperative activity, and yet the support for group awareness is insufficient in current groupware systems. This thesis has reported the processes carried out in this research aiming to improve the support for group awareness. This thesis is organised into three parts:

- Part I—The Evolution of F@—describes the background of the research (Chapter 2) and reports empirical studies of distributed groupware systems (Chapter 3).

- Part II—The F@ Framework of Group Awareness—presents the F@ (read as “fat”) framework of group awareness (Chapter 4).

- Part III—Applying F@ to the Design of Awareness Mechanisms—presents case studies of applying the F@ framework to designing awareness mechanisms in Instant Messaging (Chapter 5), collaborative authoring (Chapter 6) and other groupware systems (Chapter 7).

Chapter 1 starts the thesis by introducing the topic of this research with the specific scope into which the research falls. Chapter 1 states the problem that motivates this research, and identifies the aim that is set out to address the research problem. This research aims to address two objectives:

(1) The first objective is to develop a novel framework of group awareness to gain a useful and relevant understanding of group awareness.

(2) The second objective involves designing innovative mechanisms to support group awareness.
Chapter 2 provides the foundation for the research.

- First, Chapter 2 examines briefly the notion of group awareness by considering various definitions of group awareness, and describing fundamental characteristics of group awareness.

- Second, Chapter 2 reviews the benefits and problems of providing group awareness. Three major benefits include support for communication, support for coordination and support for conventions. Two major problems involve privacy violation and disruption.

- Third, Chapter 2 examines existing mechanisms that have been developed to support group awareness. These mechanisms include mechanisms for communication, coordination, and conventions.

- Fourth, Chapter 2 analyses existing frameworks and models developed to conceptualise the concept of group awareness. Three major existing frameworks of group awareness have been examined, including spatial-based models, descriptive frameworks, and notification-based models.

- Finally, Chapter 2 presents the author’s matrix of group awareness. The matrix presents an overview of fundamental aspects of group awareness.

Chapter 3 reports three empirical studies of distributed groupware systems, including user studies of Instant Messaging, collaborative authoring and multi-player computer games. These studies were carried out to gain an understanding of how people maintain group awareness in distributed workspaces. The results drawn from these studies were used to construct the F@ framework.

Chapter 4 presents the F@ framework of group awareness, and compares F@ with other existing frameworks of group awareness. This research develops F@ to respond to the two research objectives of understanding group awareness and of designing mechanisms for group awareness. F@ includes two parts—the abstract level and the concrete level.
Chapter 8  Conclusions and Future Work

- The abstract level presents descriptions of four sub-types of group awareness (referred to as awareness schools), namely conversational awareness, workspace awareness, contextual awareness and self-awareness. The descriptions of these four awareness schools help extend groupware developers’ understanding of group awareness. More specifically, the abstract level identifies a specific set of information, named awareness elements, associated with each awareness school. These awareness elements are grouped into six dimensions of “Who”, “What”, “Where”, “When”, “Why” and “How” (5W1H).

- The concrete level formalises fundamental elements of group awareness, especially those related to time (for example, past, current and future events). The purpose of formulating awareness elements is to capture precise and concise requirements of group awareness support.

Chapter 5 presents a case study of applying F@ to the development of awareness mechanisms for Instant Messaging. In particular, Chapter 5 reports the design, implementation and evaluation of two awareness mechanisms, including Relaxed Instant Messenger (RIM) and Conversation Dock (ConDock). RIM and ConDock are designed to enhance conversational awareness in text-based communication. RIM addresses the issue of conversational awareness in group discussion, whilst ConDock addresses the issue of supporting awareness of multiple, concurrent conversations.

Chapter 6 presents a case study of applying F@ to the development of awareness mechanisms for synchronous collaborative authoring. Chapter 6 describes the design, implementation and evaluation of Extended Radar View (ERV) and Modification Director (MD). These two mechanisms are designed to improve workspace awareness in collaborative authoring. ERV is able to support workspace awareness and bridge the contextual gap caused by the conventional radar view. MD is designed to help people stay aware of changes in a shared document.

Chapter 7 presents an interpretive discussion of the application of F@ to designing awareness mechanisms for some other groupware systems, including systems for collaborative drawing, electronic meetings, collaborative programming and multi-player computer games. Although mechanisms discussed in this interpretive analysis have neither been implemented nor evaluated, more importantly, the analysis indicates the
potential of applying F@ to different groupware systems. In addition, Chapter 7 presents a discussion of F@ based on two case studies presented in Chapter 5 and Chapter 6 and the interpretive analysis. The discussion of F@ focuses on the two issues of:

1. to what extent F@ provides an understanding of the concept of group awareness, and
2. to what extent F@ can be applied to the design of supporting mechanisms.

8.2 Revisiting the research aim

As stated in Chapter 1, the aim of this research is to enhance the support for group awareness in synchronous distributed groupware systems. This aim involves two specific objectives:

- The first objective is to construct a framework of group awareness. The framework is used to help groupware developers gain a better understanding of group awareness.
- The second objective is to demonstrate that the framework is also useful in the development of specific mechanisms to support group awareness.

This research met these two objectives by constructing the F@ framework of group awareness and applying F@ to developing awareness mechanisms.

- First, the F@ framework is constructed to extend the current understanding of group awareness. F@ is developed based on theories of groups (including coordination theory and the model of virtual teams), previous research on group awareness and empirical studies of group collaboration. F@ is comprised of the abstract level and the concrete level. One the one hand, the abstract level presents detailed descriptions of four awareness schools. These descriptions are useful in gaining an understanding of the concept of group awareness. On the other hand, the concrete level presents formal requirements of fundamental awareness aspects. The formulas of awareness elements are useful in presenting precisely what needs to be provided to support group awareness.
Second, this research has devoted significant effort in applying F@ to developing awareness mechanisms. Chapter 5 presents the development of two innovative awareness mechanisms—RIM and ConDock—that are designed to enhance conversational awareness in text-based communication. Chapter 6 presents two awareness mechanisms—ERV and MD—that are designed to improve workspace awareness in collaborative authoring. In addition, Chapter 7 presents an interpretive analysis of the application of F@ to designing awareness mechanisms for other groupware systems.

8.3 Key contributions of the research

There are several original contributions that this research has made to research on groupware awareness. The major contributions include a novel framework of group awareness and four innovative awareness mechanisms. The minor contribution is devoted to the transparent adaptation approach that is used to implement awareness mechanisms for third-party groupware systems.

The first major contribution that this research has made to the literature is F@, an innovative framework of group awareness. The F@ framework includes the abstract level and the concrete level.

- The abstract level describes four awareness schools, including conversational awareness, workspace awareness, contextual awareness and self-awareness. The descriptions of these four awareness schools are useful in providing a useful, relevant understanding of group awareness. In addition, they help groupware developers design awareness mechanisms by examining specific sets of awareness information.

- The concrete level utilises temporal logic to present a formal description of fundamental aspects of group awareness. The formal description is useful in presenting precise and concise requirements of awareness support.

The second major contribution of this research is devoted to four innovative awareness mechanisms that are developed to enhance group awareness.

- The first awareness mechanism is RIM. To the best of the author’s knowledge, RIM is the first text-based communication tool that integrates the sequential
layout with threaded layout to improve awareness in a group conversation. In addition to its innovative interface design, RIM implements the utterance rule-based principle that is useful in creating a structured, coherent conversation.

- The second awareness mechanism is ConDock, which utilises fisheye views to support awareness of multiple, concurrent conversations. ConDock allows people stay aware of the presence of multiple conversations simultaneously and interact easily with each individual conversation.

- The third awareness mechanism is ERV, which integrates the radar view with the over-the-shoulder view and telecaret's-eye view to enhance awareness in collaborative authoring. By doing so, ERV contributes significant enhancements to the conventional radar view in bridging a contextual gap between the global view and local detail.

- The last awareness mechanism is MD, which supports tracking and notification of changes in a shared document. MD notifies users when other people modify their text, and also allows users to track where changes occur. MD presents changes occurring in the shared document in chronological order, and provides contextual information about the changes, such as information about the content of the changes, users who made the changes, locations in the document at which the changes occurred, and the time when the changes were made.

A minor contribution of this research is the transparent adaptation approach that can be used to implement awareness mechanisms for third-party groupware systems. This approach allows implementing awareness mechanisms for groupware systems without having to access the source code of the groupware systems. The transparent adaptation approach has been successfully applied to the implementation of ConDock (as a plug-in for Microsoft MSN Messenger), ERV and MD (as plug-ins for Microsoft Word).

8.4 Future work

This section presents an outline for future research to further evolve the F@ framework and address implementation issues of awareness mechanisms presented in Chapter 5 and Chapter 6.
There are a few aspects of F@ that could be extended so that it could be more comprehensive and more generalisable.

- First, as discussed in Section 7.3, F@ could be extended to address the mapping between cooperative tasks and visualisation techniques used to present awareness information.

- Second, an analysis of mechanical approaches used to implement mechanisms, such as the toolkit-based approach and transparent adaptation approach, can be added into F@.

- Third, the concrete level currently formalises fundamental elements of group awareness, particularly those related to temporal aspects. As future work, the concrete level could be enhanced to present a more comprehensive coverage of formalisation of other awareness elements.

- Finally, to make F@ more generalisable, it is necessary to apply F@ to designing awareness mechanisms for other groupware systems, such as systems for collaborative drawing, electronic meetings, collaborative programming and multi-player computer games. Designing and building mechanisms to support group awareness in other groupware systems will be one of the main tasks of future work.

Regarding specific awareness mechanisms, including RIM, ConDock, ERV and MD, they can be further improved to address current implementation issues. For example,

1. RIM can be enhanced to support multiple focal points of a conversation utilising the fisheye view technique;

2. ConDock can be extended to provide more graphical content;

3. ERV can be improved to overcome the overlapping issue (that is, when two over-the-shoulder views overlap); and

4. MD can be extended to support a history of modification.

Once proposed enhancements are made to these mechanisms, further usability tests would be carried out to evaluate the mechanisms and determine the usefulness of
enhanced features. The results of future evaluation can also be used to reflect on the application of F@ to mechanism design.
Bibliography


[115] Hayne, S., Pendergast, M., and Greenberg, S., "Implementing Gesturing With
Bibliography


[228] Tran, M. H., Raikundalia, G. K., and Yang, Y., "Methodologies and Mechanism Design in Group Awareness Support for Internet-Based Real-Time Distributed Collaboration", Proceedings of the Fifth Asia Pacific Web Conference


[237] Tran, M. H., Yang, Y., and Raikundalia, G. K., "The Transparent Adaptation


Glossary

**SWIH**: Who, What, Where, When, Why, How

**AIM**: American Online’s Instant Messenger

**ANOVA**: Analysis Of Variance

**API**: Application Programming Interface

**APIC**: Application Programming Interface Consumer

**AR**: Augmented Reality

**BLF**: Backward Looking Function

**CMC**: Computer-Mediated Communication

**COM**: Component Object Model

**ConDock**: Conversation Dock

**COPPER**: COllaborative Pair Programming EditoR

**CSCW**: Computer Supported Cooperative Work

**CVE**: Collaborative Virtual Environment

**EI**: Event Interface

**EMS**: Electronic Meeting System

**ENI**: Event and Notification Infrastructure

**ERV**: Extended Radar View

**FLF**: Forward Looking Function
**Glossary**

**HCI**: Human Computer Interaction

**HMD**: Head-Mounted Display

**IM**: Instant Messaging

**IRC**: Internet Relay Chat

**LAN**: Local Area Network

**MCG**: Multi-player Computer Game

**MD**: Modification Director

**MoMA**: Model of Modulated Awareness

**NESSIE**: awareNESS envIronmEnt

**OI**: Object Interface

**OS**: Operating System

**PC**: Personal Computer

**REDUCE**: REal-Time Distributed Unconstrained Collaborative Editing

**RIM**: Relaxed Instant Messenger

**SCA**: Synchronous Collaborative Authoring

**SCDS**: Synchronous Collaborative Drawing System

**TL**: Temporal Logic

**VoIP**: Voice Over Internet Protocol

**UI**: User Interface

**URP**: Utterance Rule-based Principle

**USB**: Universal Serial Bus
**WYSIWID:** What You See Is What I Do

**WYSIWIS:** What You See Is What I See

**XML:** eXtensible Markup Language
Appendix A

Materials of the Empirical Studies

A.1 Empirical study of Instant Messaging

A.1.1 Demographic questions

1. Gender
2. Age
3. Which Instant Messengers have you used and how long have you been using them? (If you started using an Instant Messenger, say 3 years ago and have used it for, say 6 months, your response to this question would be 6 months.)
   - Yahoo Messenger
   - ICQ
   - AOL’s Instant Messenger (AIM)
   - MSN/Windows Messenger
   - Other (specify name)
4. What is your most favourite Instant Messenger? And, why is it your most favourite messenger? (If your favourite Instant Messenger is NOT one of the following four Instant Messengers, please select the "Other" option and specify the name of your favourite Instant Messenger.)
   - Yahoo Messenger
   - ICQ
   - AOL’s Instant Messenger (AIM)
   - MSN/Windows Messenger
   - Other (specify name)
   Why is it your most favourite Instant Messenger?
5. How often do you log in Instant Messengers?
6. When do you log in Instant Messengers? (you can appear either offline or online)
7. Do you ever login Instant Messengers but you set your status as offline or invisible?
8. When do you often appear online (i.e., set your status available or visible)?
9. For how long per day are you actively using Instant Messengers to chat?
10. How often do you have *Webcam* conversations?
11. How often do you have *audio* conversations?
12. What is a *maximum* number of people you ever chat with at the same time? (Can be in a group chat or separate private chats). If a maximum number of people you ever chat with is **more than 1**, do you ever use different Instant Messengers at the same time to chat with different people? (No/Yes)
13. Do you ever find Instant Messengers limited to express your ideas?
14. Do you ever chat with people located at different countries?

### A.1.2 7-point Likert scale questionnaire

<table>
<thead>
<tr>
<th>Questions</th>
<th>Strongly Disagree</th>
<th>Moderately Disagree</th>
<th>Slightly Disagree</th>
<th>Undecided</th>
<th>Slightly Agree</th>
<th>Moderately Agree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>I like to have Webcam conversations.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>When my Webcam is on, I want to know who are currently viewing my Webcam.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>For each person in my buddy list, I want to know whether the person is able to have video conversations (i.e., whether the person has a Webcam installed).</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Having video on all the time is useful.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I like to have audio conversations (i.e., voice chat).</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>When I speak in an audio conversation, I want to know who can hear me.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>For each person in my buddy list, I want to know whether the person is able to have audio conversations (i.e., whether the person has microphone and speakers installed).</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Having audio on all the time is useful.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>In a group audio conversation, if a person is speaking, I want to know who that person is.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I feel more comfortable with text-based conversations than audio or Webcam conversations.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I want to set different statuses for different people in my buddy list.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I don’t want to control who can see me online.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I want to set different avatars for different people in my buddy list.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sometimes, I want to show other people, which earlier messages of the same conversation I want to refer to.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Displaying the time each message was sent is important.

After I send a message, I want to know if other people actually receive my message.

I want to link emoticons with correct messages sent earlier by other people.

I don't want to know geographical locations of other people.

I want to know other people's local time.

I want to know how long other people have been idle.

I want to know how long other people have been online.

Knowing if a person in my buddy list is busy chatting with someone else is necessary.

When I am in a group chat, I want to know a total number of messages sent by each person. (For example, how many messages were sent by person A, how many messages were sent by person B, and so on)

Playing sound when a friend comes online is not necessary.

When I join a group chat, I want to know who were there before and have already left.

When I join a group chat, I want to know who are going to join.

When a conversation is getting long, I want to know which part of the conversation other people are viewing.

In a group chat, I want to see pictures of other people next to their nicknames.

Playing sound when a friend goes offline is not necessary.

Playing sound when a new message arrives is annoying.

Being able to send offline message is important.

In a group chat, I want to send private voice messages.

I want to be able to edit previous messages of the same conversation.

I want to customise my emoticons.

### A.1.3 Open-ended questions

1. What are your uses of audio and video chats?
2. What are your uses of avatars, online statuses and sound alerts?
3. What other features do you want when using Instant Messengers?
Appendix A  Materials of the Empirical Studies

A.2  Empirical study of collaborative authoring

A.2.1  5-point Likert scale questionnaire

<table>
<thead>
<tr>
<th>Questions</th>
<th>Not at all important</th>
<th>Fairly unimportant</th>
<th>Neutral</th>
<th>Fairly important</th>
<th>Very important</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowing who is in the workspace</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Knowing tasks for which other users are responsible</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Knowing how much time has elapsed since other users have used REDUCE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Knowing where other users are geographically located</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Knowing how long other users have been in the workspace</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Being able to view the list of past actions carried out by a specific user</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Knowing parts of a document on which other users are currently working</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Knowing parts of a document at which other users are currently viewing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Knowing what actions other users are going to take in the future</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Knowing what actions other users are currently taking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seeing the position of other users’ cursors</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Knowing to what extent you have completed your work compared to the extent others have completed their work</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Knowing to what extent a portion of a document has been completed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Knowing if other users know what I have been doing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Being able to comment on what other users have done</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Knowing if other users are satisfied with what I have done</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Having voice communication</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Having video communication</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>In the case of nonverbal communication, having a communication tool that supports communication between users</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A.2.2  Open-ended questionnaire

1. How would you expect REDUCE to show you who is in the workspace?
2. How would you expect REDUCE to show you which tasks other users are responsible for?
3. How would you expect REDUCE to show you where other users are physically located?
4. How would you expect REDUCE to show you how long other users have been in the workspace?
5. How would you expect REDUCE to show you the list of past actions carried out by other user?
6. How would you expect REDUCE to show you which parts of a document other users are currently working on?
7. How would you expect REDUCE to show you which parts of a document other users are currently looking at?
8. How would you expect REDUCE to show you what actions other users are currently taking?
9. How would you expect REDUCE to show you what actions other users are going to take in the future?
10. How would you expect REDUCE to show you to what extent a portion of a document has been completed?
11. How would you expect REDUCE to show you an overall view of the document?
12. How would you expect REDUCE to show you to what extent you have completed your work compared to the extent other users have completed their work?
13. What communication tools do you think can be used?
Appendix A

Materials of the Empirical Studies

A.3 Empirical study of multi-player computer games

A.3.1 Demographic questions

1. Gender:
2. Age:
3. How long have you been playing this game?
   - □ <1 year
   - □ 1 – 2 years
   - □ 3- 5 years
   - □ > 5 years
4. How often do you play this game?
   - □ Many timers per day
   - □ Few times per day
   - □ Few times per week
   - □ Few times per month or less
5. Have you ever played computer games with multiple displays before?
   - □ Yes
   - □ No
6. Are you left-handed or right-handed?
   - □ Left-handed
   - □ Right-handed
7. Do you prefer the second monitor to be on the left or on the right of the primary monitor?
   - □ Left
   - □ Right
   - □ Undecided

A.3.2 Open-ended questions

8. Overall, what do you think of playing game with the presence of the second monitor?
9. Can you tell me about your experience with the dual monitor setting?
10. Do you feel it difficult to adapt to this game environment?
11. What you like about the presence of the second monitor?
12. What you don't like about the presence of the second monitor?
13. Do you think dual monitors improve or impair your performance?
14. To what extent does the second monitor improve (or impair) your performance?
15. During the game, when is the second monitor useful in supporting your performance?
16. During the game, when is not the second monitor useful in supporting your performance?
17. How do you feel when the second monitor is present?
18. Do you ever find the second monitor disruptive?
19. Do you think information presented in the second monitor helpful?
20. Do you enjoy playing the game more (or less) when the second monitor is present?
21. How do you compare the level of communication you have with other players in two different settings: when the second monitor is present and is not present?
22. Does the second monitor support the communication in your team? (if yes, to what extent)
23. Do you have any suggestions to improve the use of dual monitors?
## Appendix B

### Materials of the Evaluations

#### B.1 Evaluation of RIM

**B.1.1 7 Likert scale questionnaire**

<table>
<thead>
<tr>
<th>Overall reaction to RIM</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td>RIM is terrible</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>wonderful</td>
</tr>
<tr>
<td>RIM is difficult</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>easy</td>
</tr>
<tr>
<td>RIM is frustrating</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>satisfying</td>
</tr>
<tr>
<td>RIM is rigid</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>flexible</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Learnability</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning to use RIM</td>
<td>difficult</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>easy</td>
</tr>
<tr>
<td>Remembering interaction styles of RIM</td>
<td>difficult</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>easy</td>
</tr>
<tr>
<td>Remembering the meaning of different icons</td>
<td>difficult</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>easy</td>
</tr>
<tr>
<td>Associating colours with topics of a conversation</td>
<td>difficult</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>easy</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chat Area</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Composing new messages in Chat Area</td>
<td>difficult</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>easy</td>
</tr>
<tr>
<td>A coloured icon shows to which topic the current composed message belongs</td>
<td>not useful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>useful</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tree Canvas</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td>The message tree helps you structure your discussion</td>
<td>bad</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>good</td>
</tr>
<tr>
<td>Navigating messages in the tree</td>
<td>difficult</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>easy</td>
</tr>
<tr>
<td>Keeping track of new messages in the tree</td>
<td>difficult</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>easy</td>
</tr>
<tr>
<td>Do colours help you recognise messages of different topics in the tree?</td>
<td>difficult</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>easy</td>
</tr>
<tr>
<td>Creating new messages by clicking on the tree</td>
<td>inconvenient</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>convenient</td>
</tr>
<tr>
<td>The tree helps you post a direct message</td>
<td>bad</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>good</td>
</tr>
</tbody>
</table>
### Appendix B Materials of the Evaluations

<table>
<thead>
<tr>
<th>Feature</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Message Canvas</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keeping track of new messages on Message Canvas</td>
<td>difficult</td>
<td>easy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Using coloured icons to associate messages on Message Canvas with topics</td>
<td>not useful</td>
<td>useful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mapping messages between Message Canvas and Tree Canvas</td>
<td>difficult</td>
<td>easy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Topic List</strong></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>NA</td>
</tr>
<tr>
<td>Showing the list of main topics</td>
<td>not useful</td>
<td>useful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Showing the number of replies under each topic</td>
<td>not useful</td>
<td>useful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Highlighting your current active topic</td>
<td>not useful</td>
<td>useful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Buddy List</strong></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>NA</td>
</tr>
<tr>
<td>Coloured keyboard icons show multiple people typing</td>
<td>not useful</td>
<td>useful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Using a colour to indicate the topic to which a new message goes</td>
<td>not useful</td>
<td>useful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Showing how long you and other people have been logged in</td>
<td>not useful</td>
<td>useful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Usability of RIM</strong></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>NA</td>
</tr>
<tr>
<td>Is the display layout consistent?</td>
<td>bad</td>
<td>good</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Is the display orientation consistent? (panning vs. scrolling)</td>
<td>bad</td>
<td>good</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Are users' actions required consistent?</td>
<td>bad</td>
<td>good</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Are icons used consistently?</td>
<td>bad</td>
<td>good</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does RIM provide visually distinguished colours?</td>
<td>bad</td>
<td>good</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Quality of discussion using RIM</strong></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>NA</td>
</tr>
<tr>
<td>RIM supports your discussion</td>
<td>not useful</td>
<td>useful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RIM helps structure the group discussion</td>
<td>ineffective</td>
<td>effective</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RIM helps you follow the discussion</td>
<td>difficult</td>
<td>easy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix B

Materials of the Evaluations

| How is the quality of your discussion using RIM compared to that of Gaim? | RIM is worse |  |  |  | RIM is better |
|---|---|---|---|---|

B.1.2 Open-ended questions

1. Please list the most negative aspect(s) of RIM.
2. Please list the most positive aspect(s) of RIM.
B.2 Evaluation of ConDock

B.2.1 7 Likert scale questionnaire

<table>
<thead>
<tr>
<th>Questions</th>
<th>Strongly disagree</th>
<th>Moderately disagree</th>
<th>Slightly disagree</th>
<th>Neutral</th>
<th>Slightly agree</th>
<th>Moderately agree</th>
<th>Strongly agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConDock helps me manage multiple conversation windows</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>It is easy to know if there is a new message</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>It is easy to distinguish conversation windows shown in ConDock</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dragging windows out of ConDock is difficult to use</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I like a zooming effect implemented in ConDock</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A zooming effect is difficult to use</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>It is difficult to map a MSN conversation window with a cloned window shown in ConDock</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

B.2.2 Open-ended questions

1. Please list the most negative aspect(s) of ConDock.
2. Please list the most positive aspect(s) of ConDock.
### B.3 Evaluation of CoWord+

#### B.3.1 7 Likert scale questionnaire

<table>
<thead>
<tr>
<th>Overall reaction to CoWord</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoWord is terrible</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>wonderful</td>
</tr>
<tr>
<td>CoWord is difficult</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>easy</td>
</tr>
<tr>
<td>CoWord is frustrating</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>satisfying</td>
</tr>
<tr>
<td>CoWord is rigid</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>flexible</td>
</tr>
<tr>
<td>It is useful to separate other users’ working and viewing areas.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>not useful usable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Extended Radar View (ERV)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning to use ERV.</td>
<td>difficult</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>easy</td>
</tr>
<tr>
<td>The telecarets-eye view (at left) is useful in facilitating your tasks.</td>
<td>not useful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>useful</td>
</tr>
<tr>
<td>The over-the-shoulder view (at right) is useful in facilitating your tasks.</td>
<td>not useful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>useful</td>
</tr>
<tr>
<td>Knowing the other people's working area in a shared document.</td>
<td>not useful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>useful</td>
</tr>
<tr>
<td>The over-the-shoulder window is useful in showing a detailed view of what another user is viewing.</td>
<td>not useful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>useful</td>
</tr>
<tr>
<td>It is convenient and easy to map views shown in the radar and the over-the-shoulder window.</td>
<td>not useful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>useful</td>
</tr>
<tr>
<td>How useful is ERV in supporting your tasks?</td>
<td>not useful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>useful</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Modification Director (MD)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning to use MD</td>
<td>difficult</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>easy</td>
</tr>
<tr>
<td>Showing the page number at which a modification occurred is helpful.</td>
<td>not useful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>useful</td>
</tr>
<tr>
<td>Showing part of a modification helps me know what had been changed.</td>
<td>not useful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>useful</td>
</tr>
<tr>
<td>Showing the name of a person who made a modification is useful.</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
</tr>
<tr>
<td>Displaying elapsed time of a modification helps me know when the modification was made.</td>
<td>not useful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>useful</td>
</tr>
<tr>
<td>Showing the status of a modification, such as single deletion and multiple deletions, is useful to keep updated with the changes.</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
</tr>
<tr>
<td>Fading colours to indicate the age of a modification is effective.</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
</tr>
</tbody>
</table>

267
Appendix B

Materials of the Evaluations

## Appendix B

### Materials of the Evaluations

<table>
<thead>
<tr>
<th>Information shown in MD is easy to understand.</th>
<th>strongly disagree</th>
<th>strongly agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information shown in MD is useful to stay aware of the changes in a document.</td>
<td>strongly disagree</td>
<td>strongly agree</td>
</tr>
<tr>
<td>How useful is MD in supporting your tasks?</td>
<td>not useful</td>
<td>useful</td>
</tr>
</tbody>
</table>

### Dual monitors

<table>
<thead>
<tr>
<th>Dual monitors</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning to use dual monitors.</td>
<td>difficult</td>
<td>easy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keeping track of windows on two monitors is difficult.</td>
<td>difficult</td>
<td>easy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The dual monitor is useful in separating the window of a main document and the windows of ERV and MD.</td>
<td>not useful</td>
<td>useful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Preferring the dual-monitor setting to the one-monitor setting.</td>
<td>one monitor</td>
<td>dual monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>It is easy to switch views between two monitors?</td>
<td>difficult</td>
<td>easy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How useful is the dual monitor setting in supporting your collaboration?</td>
<td>not useful</td>
<td>useful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### B.3.2 Open-ended questions

1. Please list negative aspect(s) that you found in ERV
2. Please list positive aspect(s) that you found in ERV
3. Please list negative aspect(s) that you found in MD
4. Please list positive aspect(s) that you found in MD
5. Please list negative aspect(s) that you found in using dual monitors
6. Please list positive aspect(s) that you found in using dual monitors

268
Appendix C

Usefulness of Dual Monitors

This appendix reports findings of the empirical study of multi-player computer games (Section 3.4) in relation to the usefulness of dual monitors. In particular, the effects that dual monitors made to participants’ game performance and their enjoyment, participants’ behaviours in and adaptation to the dual monitor setting are presented.

C.1 Dual monitors affect game performance

Participants had diverse opinions about the usefulness of dual monitors in supporting their game performance. In the study, participants played two games of different genres in which their tasks were quite dissimilar. Because of the different nature of two games, the reported effectiveness of dual monitors varied between the two games, as discussed below.

C.1.1 Dual monitors are useful in Counter-Strike

Most participants involved in the Counter-Strike experiments acknowledged that dual monitors were useful in supporting their game performance. Participants found themselves more confident when dual monitors were present because they were well informed of what their team player was doing in the game. During a game, participants mainly used the secondary monitors to quickly identify the location of the other player and to keep themselves updated on the status of the other player, such as how healthy they were, which weapons they currently possessed, and so on.

However, not all Counter-Strike participants found dual monitors useful. A few participants reported that the secondary monitor was in fact sometimes distracting them from their game. The secondary monitor drew participants’ attention away from what happening on their primary screen. A common negative occurrence was that when participants spent too much time looking at the secondary monitor then they were eventually killed. Also, at some point in the game they were confused between the two screens, especially if they and their team member were in the same location and held the
same type of weapon. This caused confusion as to which screen was actually the primary screen that they could control.

Another side effect of dual monitors was that the presence of the second monitor had changed the nature of the Counter-Strike game in a negative way. As reported by two participants, in a first-person-shooter game like Counter-Strike, they often made their decisions based on their sense of the game. However, by having dual monitors, they became too dependent of the secondary monitor and lost their sense of a game. That was considered as negative.

**C.1.2 Dual monitors are less useful in WarCraft III**

Dual monitors were not reported as being useful in WarCraft III to the extent that they were in Counter-Strike. At the beginning of the WarCraft III experiments, before the game started participants were really excited to know that dual monitors showed their team player’s screen and they believed that dual monitors would definitely be useful. However, when participants actually played a game with dual monitors, they tended to ignore the secondary monitors. In the interviews, participants responded that they did not use the second monitor much because the nature of WarCraft III required a lot of attention on the primary screen. Players were required to manage many units of different kinds, conduct many activities to gain resources and to control units.

**C.2 Participants enjoyed dual monitor gaming**

Even though 75% of participants had never experienced dual monitors prior to the experiments, participants’ comments reflected the overwhelming interest in playing games with the support of dual monitors in future. Some participants commented that dual monitors created a new level of atmospheric immersion in the games. Though participants felt more confident of playing the game with a real-time updated view of what happening to another team player, they also believed that dual monitors had made the game more challenging and interesting to some extent. Several participants would even consider adding a second monitor to their PCs at home.

**C.3 Participants’ use of dual monitors**

**C.3.1 Many quick glances; few quick glances followed by a long glance**

Two patterns of how participants used the second monitor emerged from the analysis of the head-and-shoulders video.
Firstly, most participants often glanced quickly at the secondary monitors. Here, a *quick glance* (QG) is defined as a look that lasts shorter than one second. On average, the Counter-Strike participants used around three QGs per minute, whereas the majority of the WarCraft III participants used less than one QG per minute. The difference in a number of QGs participants required could be caused by the difference in nature of the two games. Counter-Strike is a first-person-shooter game in which, as the name suggests, a player controls a [virtual] character with a weapon from a first-person perspective. It means, very specifically, that a player controls only one character. On the other hand, WarCraft III is a strategy game where a player needs to control a substantial number of different units (the number can be 100 units in many cases). As a result, WarCraft III players are often required to pay great attention to the primary monitor compared to Counter-Strike players. That explains why the WarCraft III participants used dual monitors less than the Counter-Strike participants. In the interviews, participants responded that QGs were very convenient for them to quickly keep track of their team player’s whereabouts (that is, the positions of their team mates in the game).

Secondly, participants often used one or two QGs followed by a *long glance* (LG)—a longer-than-1-second look. This could be explained as follows. After one QG or two, participants were quickly aware of the positions and what happening with their team players. If the situation required participants to pay more attention to what currently happening with their team players, a LG was used.

### C.3.2 Physical layout of dual monitors

The study was interested in knowing how participants arranged the physical layout of dual monitors.

**(a) How the second monitor was physically positioned relative to the primary monitor**

In the default setting, the primary and secondary monitors were horizontally, and the secondary monitor was on the right hand side of the primary monitor (Figure C.1a). But in many cases, participants actually adjusted their secondary monitor to create an angle between the two monitors as shown in Figure C.1b. This study has not identified what the most favourite angles are. This finding, however, raises interesting insights which should be taken into account when designing applications or hardware for dual monitors.
Appendix C

Usefulness of Dual Monitors

(a) Default layout

(b) Preferred layout

Key:
P M: primary monitor
S M: secondary monitor

Figure C.1: The layout of primary and secondary monitors.

(b) How the second monitor was positioned relative to participants’ primary hand

As shown in Table C.1, most participants wanted the second monitor to be on the side of their primary hand. More than 70% of the right-handed participants (that is, 25 out of 35 people) preferred the second monitor to be on the right hand side of the primary monitor; and 60% of the left-handed participants (that is, 3 out of 5 people) preferred the second monitor to be on the left hand side of the primary monitor.

In the interviews, participants were asked about a preferred side the second monitor, and their responses were as follows: participants were required to use both hands to play the game—the primary hand to control the mouse and the other hand to operate the keyboard. They preferred the second monitor to be on the side of their primary hand (that is, on the side of the mouse) because they did not want to look across their non-primary hand (that is, the hand was used to function the keyboard) when playing a game.

<table>
<thead>
<tr>
<th>Preferred side</th>
<th>Left</th>
<th>Right</th>
<th>Undecided</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary hand</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Left</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Right</td>
<td>5</td>
<td>25</td>
<td>5</td>
<td>35</td>
</tr>
<tr>
<td>Total</td>
<td>8</td>
<td>25</td>
<td>7</td>
<td>40</td>
</tr>
</tbody>
</table>

Table C.1: Relation between a participant’s primary hand and their preferred side for the second monitor.

C.3.3 How participants adapted to dual monitors

As well as identifying the basic potential of dual monitors, this study provides insight into another noteworthy issue, namely how participants adapted themselves and their game strategies to the dual-monitor environment. Although this is not the main focus of
this study, this section discusses it briefly. It was instructive to listen to participants’ accounts of how they learnt to adapt themselves to the dual-monitor setting. The majority of participants had not previously played games with dual monitors. Thus it was unsurprisingly that they adopted a range of different approaches in making use of them. Some participants forced themselves to use secondary monitors. That led to the fact that they spent too much time viewing the secondary monitor on which appeared their team player’s screen. As a result, they lost their focus on the game and missed out events happening on the primary display. Then, participants tried to avoid watching the secondary display so that they could pay more attention to the primary monitor. In those cases, dual monitors were found disruptive as it shifted participants’ focus away from their own role, and consequently their performance was impaired.

(a) Highly experienced players adapted to the dual monitor setting quicker than less experienced players

The study found that participants’ learning curve depended on their previous game experience: the more skilful a participant was, the faster they became accustomed to the dual monitor setting. For skilful players, their good understanding of the game could allow them to flexibly take their eyes off the primary display for a quick glance at the second monitor.

(b) Adaptation to dual monitors was game dependent

As mentioned before WarCraft III and Counter-Strike are two very different games; they represented two different tasks in the study. The difference between the games led to the difference in participants’ adaptation of dual monitors. It appeared that the Counter-Strike participants adapted to the dual monitor setting faster than the WarCraft III participants. That would partly explain the previous finding that in the Counter-Strike game dual monitors were generally more useful in providing shared awareness than in WarCraft III.

The map in the WarCraft III game changes every time a new round starts and players do not know the map until it is fully explored. On the other hand, the map in the Counter-Strike game is unchanged, thus the Counter-Strike players know the map really well. Consequently, in the Counter-Strike game just from a quick glance at the second monitor, participants knew immediately where the other player was located. That is one
reason making the Counter-Strike participants adapt to dual monitors more quickly than the WarCraft III participants.

Another reason for game dependence may have been because the Counter-Strike participants played more rounds of game than the WarCraft III participants in one experiment: sixty-four rounds of Counter-Strike and four rounds of WarCraft III, respectively (see section 3.4.1). Hence, the Counter-Strike participants had more chances to adjust their approaches of using dual monitors. As a result, after the same two-hour period, the Counter-Strike participants generally made the better use of dual monitors than the WarCraft III participants.