ING-INF/05

# SUPPORTING SYNCHRONOUS COMMUNICATION IN DISTRIBUTED SOFTWARE TEAMS

by

Fabio Calefato

A Thesis
Submitted to the Department of Computer Science
in Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

University of Bari, Italy

Advisor:
Professor Filippo Lanubile

PhD Coordinator:
Professor Anna Maria Fanelli

January, 2007

*To Me, My Family,*
*and Everyone Else I Care About:*
*You Know Who You Are*

# ACKNOWLEDGEMENTS

First and foremost, I want to thank my advisor, Prof. Filippo Lanubile for accepting me as a PhD student and for all of his support throughout these three intense years. I hope I absorbed from him even just a smallest part of his skills. Thanks also to Dr. Daniela Damian for giving me the chance to conduct the experiment at UVic and for her valuable guidance during my stay in Canada.

A big 'thank you' goes to all the affiliates with the CDG/Collab: Daniela, for her patience and tolerance with me and my jokes; Domenico, for turning me in a TV-show junkie; Mario, for his work with the development of eConference. Thanks also to all the people who have shared lunch with us in the lab, bringing in a little more fun and distraction.

Many thanks to Rocco Oliveto, Fausto Fasano, and the other guys at the University of Salerno for sharing with me some relevant papers from 'inaccessible digital libraries.' Thanks also to Vincenzo Fiorentino and Roberto Kudlica at the CSI for sharing their printing facilities.

Thanks to the graduate and undergraduate students who have helped me with the development of my project.

Thanks to Google Scholar. I cannot imagine my 'research life' without it.

Thanks to my friends in my hometown, who never complained about my presence, but did complain about my complete absence lately.

The biggest thanks go to my family for keeping me away from any worry, giving me the chance to keep myself focused only on my work in the last 'heavy' months.

Last but not least, I can't do without thanking Pearl Jam: Thank you guys, you can hardly imagine all that you have done for me and what your works mean to me. I'll always cherish the memories of your concert in Italy (Pistoia, Sept. 20, 2006) and try to keep on rockin' in a free world, as much as I can!

January 2007,

Fabio

# TABLE OF CONTENTS

# ABSTRACT

The goal of this dissertation is presenting an investigation on how to help distributed software teams cope with the negative effects of distance and dynamism, using synchronous, text-based communication media.

Working across distances has become commonplace for teams today, mainly because of economic factors (e.g., round-the-clock development, market proximity, access to low-cost human resources). Nevertheless, multi-site work presents considerable loss of opportunities for rich interaction, and a very substantial reduction in frequency of communication. Following the trend to business globalization, also software development has increasingly become distributed, with little or no possibility for developers to meet. Distance has a high impact especially on development process which rely heavily on the interaction, as in the requirements definition phase where stakeholders from different organizations are brought together to share information and take decisions. Among the software development activities, requirements engineering is one of the most communication-intensive, and then its effectiveness is greatly constrained by the geographical distance between stakeholders. For this reason, the need to develop a tool infrastructure to support teams of geographically dispersed stakeholders plays a key role for coping with the lack of physical proximity when developing requirements.

The definition of requirements is a highly collaborative, interactive, and interdisciplinary process involving heterogeneous teams of stakeholders. These groups create temporary networks of independent companies, and collaborate as virtual organizations, using information technology to share skills and costs. Such teams are ad hoc in that they tend to be highly dynamic in creation, participation and release, other than being geographically dispersed and cross-organizational. Hence, ad hoc teams need easy-to-setup tools, with infrastructure and administration costs kept at minimum. To provide communication support to such distributed teams, we have developed eConference, a tool for text-based, structured communication. The inceptive idea behind eConference is to reduce the need for face-to-face meetings,

using a simple collaboration tool that minimizes infrastructure costs, potential technical problems, and the learning curve.

We have initially focused on studying text-based communication because multipoint audio-video communication poses significant practical barriers to deployment (e.g., expense, infrastructure, support), especially for short-term groups. Rich media theories on computer-mediated communication, namely Social Presence, Media Richness, and Common Ground, have overwhelmingly reported about the inadequateness of text-based communication, as compared to rich media, like face-to-face or video communication. Lean media, such as email and instant messaging, lack the ability of conveying nonverbal cues that contribute to the level of social presence (e.g., gaze, tone of voice, facial expressions), which in turns fosters individuals' motivation and mutual understanding. However, running counter to these predictions, Media Synchronicity, Time, Interaction, and Process, and Task/Technology Fit theories assert that the effectiveness of computer mediated communication depends also on contextual factors other than media richness, such as communication channel synchronicity, task typology and group temporal scope. Furthermore, Media Richness Paradox argued that the use of rich media high in social presence should be used to assure attention for small amounts of information, whereas the use of lean media low in social presence causes a decreased motivation, but increases the ability to process large amounts of information during longer periods of time. Drawing upon these theories, we argue that, by understanding the paradoxical effects of rich media high in social presence, we may be better able to select and use the most appropriate sets of media to accomplish our goals. Thus, starting with a critical review of the very many existing, and often conflicting, theories on CMC, combined with the *Task Circumplex* model for task classification, we have developed a comprehensive framework for predicting, evaluating, and comparing the goodness of task-technology fits.

Providing evidence of collaborative tools effectiveness is a challenging task. An empirical study has been conducted to investigate the support that synchronous, text-based communication provides to ad hoc groups of stakeholders involved in distributed requirements elicitation and negotiations, as compared to face-to-face interaction. The findings from the study have confirmed the results of previous

research, showing that, during the requirements meetings, the subjects perceived a higher level of comfort with face-to-face communication mode than with CMC, while keeping an equal level of motivation to participate. Nevertheless, the findings also show that: (1) compared to face-to-face requirements workshops, synchronous text-based workshops grant a higher opportunity to participate in a more structured, equal, and open discussion; (2) stakeholders are more satisfied with performance in synchronous, text-based elicitations than in synchronous text-based negotiations.

Overall, these results suggest to distributed teams of stakeholders that synchronous text-based elicitations represent a better task-technology fit than synchronous text-based negotiations, for reducing the negative effects of distance, as well as the need and the number of collocated requirements workshops.

# ABSTRACT

# (IN ITALIAN)

L'obiettivo di questa tesi è presentare uno studio condotto su come aiutare i team di sviluppo software distribuiti a contrastare gli effetti negativi dovuti alla distanza e al dinamismo, utilizzando strumenti di comunicazione sincrona e testuale.

Il lavoro distribuito è una forma di collaborazione remota tra team oggi ormai diffusissima, soprattutto a causa di benefici economici (e.g., prossimità al mercato, accesso a risorse umane a basso costo). Ciononostante, il lavoro distribuito su diversi siti presenta un considerevole decremento di opportunità di interazione 'ricca,' nonché una sostanziale riduzione nella frequenza della comunicazione stessa. Seguendo la crescente tendenza verso la globalizzazione del mercato in altri settori, anche lo sviluppo del software è divenuto una pratica sempre più 'distribuita,' con possibilità scarse o nulle per gli sviluppatori di potersi incontrare.

La distanza ha un profondo impatto in particolar modo sul processo di sviluppo del software poiché esso dipende fortemente dall'interazione tra gli individui, come durante la fase di definizione dei requisiti software, quando stakeholder appartenenti a organizzazioni diverse si incontrano per condividere informazioni e prendere decisioni. Tra le varie attività legate allo sviluppo di software, l'ingegneria dei requisiti di un software è una delle più intense dal punto di vista della comunicazione e, di conseguenza, è fortemente condizionata dalla distanza geografica esistente tra gli stakeholder coinvolti. Pertanto, quando si devono definire requisiti software, gli strumenti a supporto dei team di stakeholder geograficamente distribuiti giocano un ruolo chiave per far fronte alla lontananza fisica. La definizione di requisiti software è un'attività altamente collaborativa ed interattiva, nonché un processo che coinvolge team di stakeholder eterogenei. Questi team formano dei network temporanei di compagnie indipendenti che collaborano come una unica organizzazione virtuale, usando sistemi informatici per condividere competenze e costi. Abbiamo definito tali team *ad hoc* in quanto essi tendono ad essere fortemente dinamici in fase di creazione, partecipazione e scioglimento, oltre ad essere costituiti da membri

appartenenti a organizzazioni diverse. Pertanto, i team *ad hoc* necessitano di strumenti facili da configurare e utilizzare, che richiedano costi di infrastruttura e amministrazioni ridotti al minimo. Al fine di supportare l'interazione tra team *ad hoc*, è stato sviluppato un tool chiamato eConference, per la comunicazione testuale sincrona e strutturata. L'idea alla base di eConference è ridurre la necessità di condurre meeting 'faccia a faccia,' utilizzando uno strumento di collaborazione semplice che minimizzi i costi infrastrutturali, la possibilità che si verifichino problemi tecnici e la curva di apprendimento.

Si è deciso di focalizzare inizialmente lo studio sulla comunicazione testuale poiché la comunicazione audio-video 'molti a molti' pone ancora ostacoli pratici per l'adozione (e.g., alti costi infrastrutturali, supporto tecnico), specialmente per quanto concerne gruppi la cui vita è di durata limitata. Le teorie sulla computer-mediated communication, in particolare le teorie di Social Presence, Rich Media e Common Ground, hanno enfatizzato l'inadeguatezza della comunicazione testuale rispetto a mezzi più ricchi, come la comunicazione video e faccia a faccia. I media cosiddetti 'lean' (scarni, leggeri, quali per esempi l'email e l'instant messaging) difettano della capacità di trasportare informazioni non verbali che contribuiscono ad incrementare il livello di 'compresenza' (e.g., direzione dello sguardo, intonazione della voce, espressioni facciali) e, di conseguenza, anche la motivazione e la reciproca comprensione. Tuttavia, rinnegando le precedenti, alcune teorie più recenti quali Media Synchronicity, Time, Interaction, and Process, e Task/Technology Fit, hanno postulato che l'efficacia di una comunicazione mediata da computer dipende anche da altri fattori contestuali, come il livello di sincronia del canale, il tipo di attività da svolgere e la durata del gruppo, e non solo dalla ricchezza dei media. Inoltre, la teoria di Media Richness Paradox sostiene che l'uso di media ricchi con un grande livello di compresenza assicura un'alta attenzione degli ascoltatori partecipanti, ma solo per un tempo limitato e per una quantità ridotta di informazione da scambiare. Al contrario, i mezzi di comunicazione leggeri causano non solo una diminuzione nel livello di partecipazione, ma, al contempo, aumentano anche la capacità di processare una maggiore quantità di informazione. Basandosi su queste teorie e comprendendo meglio gli effetti talvolta paradossali derivanti dall'uso dei media più ricchi è

possibile imparare a selezionare i mezzi di comunicazione più appropriati per completare un data attività. Così, partendo da una revisione critica delle diverse e spesso contrastanti teorie di CMC esistenti e sfruttando il modello del *Task Circumplex* per la classificazione dei task, abbiamo sviluppato un framework complete per la predizione, la valutazione e il confronto tra combinazioni 'task-mezzo di comunicazione'.

Fornire evidenza dell'efficacia di uno strumento collaborativo è un compito arduo. E' stato condotto uno studio empirico per investigare il supporto fornito dalla comunicazione testuale sincrona a gruppi *ad hoc* di stakeholder coinvolti in elicitazioni e negoziazioni distribuite di requisiti software, rispetto all'interazione faccia a faccia. I risultati dello studio empirico confermano che gli stakeholder percepiscono un livello di comfort comunicativo più alto durante i workshop faccia a faccia, pur mantenendo un livello di motivazione simile ai workshop testuali. Tuttavia, i risultati dimostrano anche che (1) la comunicazione sincrona testuale garantisce durante le elicitazioni e le negoziazioni una comunicazione strutturata più aperta e non dominata rispetto alla discussione faccia a faccia; (2) gli stakeholder percepiscono un livello di soddisfazione della propria e altrui performance più alto durante le negoziazioni faccia a faccia, mentre non vi sono differenze statisticamente significative fra le elicitazioni testuali e quelle faccia a faccia, suggerendo così, che elicitazione e comunicazione testuale sincrona rappresentano una combinazione 'task-mezzo di comunicazione' migliore rispetto a negoziazione e comunicazione testuale al fine di ridurre il bisogno di workshop collocati.

# LIST OF TABLES

# LIST OF FIGURES

# Chapter 1:

# INTRODUCTION

---

*"To bring the pieces back together*
*rediscover communication"*
*Tool, "Schism," Lateralus (2001)*

## 1.1. Background

The goal of this dissertation is presenting an investigation on how to help distributed software teams cope with the negative effects of distance and dynamism, using synchronous text-based communication media.

Nowadays, no one works completely independently. Almost everyone is part of at least one group, typically several groups at any point in time. Figure 1 shows a cooperative work framework: Groups of two or more participants (P) communicate together, share information, generate ideas, organize ideas, build consensus, make decisions, and so on. Being engaged in some common work, participants interact with tools and products (i.e., artifacts of work, A). The main purpose of communication is to establish a common *understanding* of the work shared between participants. The development of the understanding happens both indirectly and directly. *Indirect communication* is denoted by the arrows that link participants to the artifacts. It happens through the manipulation of shared tools and work objects (e.g., a document, a piece of code). *Feedback* represents the information gained by the participant who directly controls a shared artifact. Further, the changes applied to an artifact convey information also to the other participants (*feed through*). *Direct communication* is denoted by the arrow between the participants and happens by speech or over communication media, such as telephone, fax, and email.

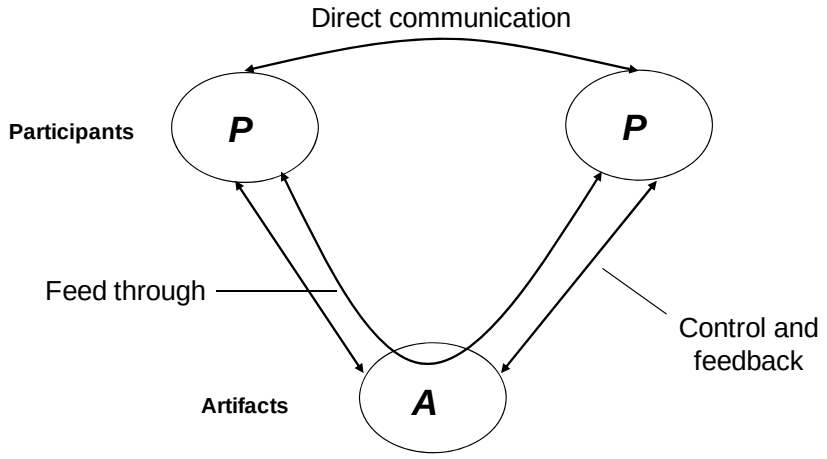


**Figure 1. Cooperative work framework: Communication as the basis of collaboration (adapted from [Dix03])**

Communicating face to face (F2F) by speech is easy for individuals. Communicating over media still remains challenging. Conducting a long-running, productive conversation through the digital medium is difficult, especially when more than a few people are involved. The difficulty of computer-mediated communication (CMC) and collaboration stands in stark contrast to our ability to easily communicate and collaborate with one another in the physical world. Yet, only a part of these shortcomings are inherent in the media used, whereas the remaining can be overcome by properly designing effective communication and collaboration tools.

Working across distances has become commonplace for teams today, mainly because of economic factors. The main benefits include the reduction of development cost (through the access to low-cost human resources) and time to market (through the round-the-clock, or follow-the-sun, development), and the increase of proximity to the different markets and customers. Nevertheless, multi-site work presents considerable drawbacks, such as the loss of opportunities for rich interaction, and a very substantial reduction in frequency of both formal and informal communication. Following the trend to business globalization, also software development has increasingly become distributed, with little or no possibility for developers to meet. Distance has a high impact especially on development processes, which rely heavily on interaction, as in the requirements definition phase, when stakeholders from different organizations are

brought together to share information and take decisions. Among the software development activities, requirements engineering is one of the most communication-intensive and then, its effectiveness is greatly constrained by the geographical distance between stakeholders. For this reason, the need to develop a tool infrastructure to support teams of geographically dispersed stakeholders plays a key role for coping with the lack of physical proximity when developing requirements.

The definition of requirements is a highly collaborative, interactive, and interdisciplinary process involving heterogeneous teams of stakeholders. These groups create temporary networks of independent companies, and collaborate as virtual organizations, using information technology to share skills and costs. Such teams are ad hoc in that they tend to be highly dynamic in creation, participation and release, other than being geographically dispersed and cross-organizational. Hence, ad hoc teams need easy-to-setup tools, with infrastructure and administration costs kept at minimum. One contribution of the research effort reported in this thesis is the proposal of a new definition of ad hoc group, which builds on the previous definitions given in the existing literature on group research and, at the same time, is compliant with the emerging scenario of short-term distributed collaborations. A second contribution is the design and implementation of two systems. The first is a plugin for Mozilla Thunderbird, aimed to blend synchronous and asynchronous text-based communication in the most used and successful collaborative tool ever made, i.e., the email client. The second system, called eConference, is intended to provide an environment for quickly and informal communication, as well as to support smooth, long-running, and structured conversations among members of ad hoc groups in particular. The inceptive idea behind eConference is to reduce the need for face-to-face meetings, using a simple collaboration tool that minimizes infrastructure costs, potential technical problems, and the learning curve.

Currently, the eConference tool only uses textual communication. We have initially focused on studying text-based communication because multipoint audio-video communication poses significant practical barriers to deployment (e.g., expense, infrastructure, support), especially for short-term groups. Rich media theories on CMC, namely *Social Presence*, *Media Richness*, and *Common Ground*, theories have

hypothesized group effectiveness to decrease when media other than F2F are used to accomplish equivocal tasks that require relational cues to be exchanged. They have reported about the inadequateness of text-based communication, as compared to rich media, like F2F and video. Lean media, such as email and instant messaging, lack the ability of conveying nonverbal cues that contributes to the level of social presence (e.g., gaze, tone of voice, facial expressions), which in turns fosters individuals' motivation and mutual understanding. However, these theories have also been criticized for considering the task to execute as an atomic activity. In addition, both Social Presence and Media Richness theories have generally been supported when tested on traditional media, such as F2F communication and telephone, whereas inconsistent empirical findings have resulted when tested on email and video.

These inconsistencies have encouraged a reconsideration of the descriptive and predictive general validity of these theories. Thus, other theories have asserted that the effectiveness of computer mediated communication depends also on factors other than media richness, such as the degree of synchronicity, task typology and group temporal scope. *Media Synchronicity* theory and *Media Richness Paradox* have started to investigate on media effects, looking at the underlying communication processes that happen in every group tasks. On the one hand, Media Richness Paradox represents a sort of 'Copernican revolution,' which capsizes the existing perspective of CMC theories, looking at communication as a cognitive process: Not only must the sender's comfort with the communication medium be taken into account, but also the motivation of receivers and, above all, their ability to process the message properly. Furthermore, Media Richness Paradox argued that the use of rich media high in social presence should be used to assure attention for small amounts of information, whereas the use of lean media low in social presence causes a decreased motivation, but increases the ability to process large amounts of information during longer periods of time. On the other hand, Media Synchronicity theory distinguishes between the interplay of two different communication processes (the conveyance of additional information, and the convergence to shared views) which vary with the degree of synchronicity of the medium. Furthermore, since a task is not actually atomic, but rather constituted of several sub-activities, Media Synchronicity theory suggests that the synchronicity level

of media should be aligned with the degree of conveyance or convergence of each sub-activity.

The concept of alignment between task and media characteristics is the very basis of the theories of *Time-Interaction-Performance* and *Task/Technology Fit*. The frameworks proposed by these theories evaluate the appropriateness of task-medium matches, considering tasks no more as somewhat atomic activities, like in Media Richness and Social Presence theories, but rather as complex sets of sub-activities and sub-processes, each having different characteristics. Likewise, also group and media characteristics have to be aligned for opportune collaborations to take place. The theories of Common Ground and *Channel Expansion* argue that groups without a history of previous collaborations, like ad hoc groups, do not share any experience and thus, have not established a level of common ground (i.e., shared understanding) sufficient for communicating effectively over lean media. Conversely, members of long-term groups are expected to communicate more effectively over impoverished media, using their shared experiences to compensate for the media leanness.

Drawing upon these theories, we argue that, by understanding the paradoxical effects of rich media high in social presence, we may be better able to select and use the most appropriate media to accomplish our goals. Another contribution of this dissertation is presenting a critical review of the very many existing, and often conflicting, theories on CMC, which have been combined in a comprehensive theoretical framework for predicting, evaluating, and comparing the goodness of task-technology fits. The proposed framework depends on McGrath's Task Circumplex, which is the most widely-used reference model in group research for task analysis, comparison, and categorization. The framework has been used as the experimental model for discussing the results from an empirical study that the researcher conducted with eConference in the field of distributed Requirements Engineering.

The last and most important contribution of this research effort is the design, performance, and analysis of an empirical study conducted to gain insights into task-technology fit for the execution of computer-supported requirements workshops, as compared to F2F interaction. The theoretical background briefly outlined in this chapter shows that providing evidence of communication tools effectiveness can be overly

challenging. The effects of collaborative systems are contingent on many of factors that differ from situation to situation, according to the context – i.e., *group composition*, *task typology*, and *communication medium*, of a group process. Thus, also the outcome (e.g., efficiency, effectiveness, product quality) depends upon the interaction between the group process and these varying contextual factors. Therefore, results from empirical study with collaborative tools must be qualified by the context – the group, task, medium – to which they apply (see Figure 2).

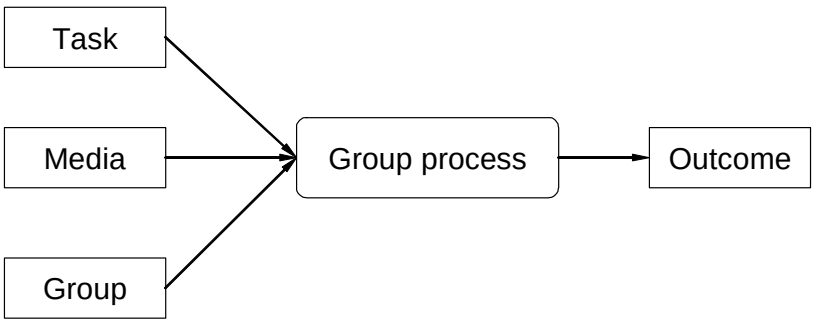


**Figure 2. Framework adopted to model the interaction of contextual factors with group process and their effect on the outcome (adapted from [Nun91])**

The context of the empirical study is provided by its overall goal: To evaluate the support of synchronous text-based communication (i.e., the communication medium) for conducting distributed requirements elicitation and negotiations (i.e., the task typology), involving ad hoc groups of stakeholders (i.e., the group composition). In particular, we investigated two research questions to understand ($RQ_1$) how text-based requirements workshops vary from F2F counterparts, and ($RQ_2$) whether both synchronous text-based elicitation and synchronous text-based negotiation represent an appropriate task-technology fit. The findings from the study have confirmed the results of previous research, showing that, during the requirements meetings, the subjects perceived a higher level of comfort with F2F communication mode than with CMC, while keeping an equal level of motivation to participate. Nevertheless, the findings also show that: (1) compared to F2F requirements workshops, synchronous text-based workshops grant a higher opportunity to participate in a more structured, equal, and

open discussion; (2) stakeholders are more satisfied with performance in synchronous, text-based elicitations than in synchronous text-based negotiations.

Overall, these results suggest to distributed teams of stakeholders that synchronous text-based elicitations represent a better task-technology fit than synchronous text-based negotiations, for reducing the negative effects of distance, as well as the need and the number of collocated requirements workshops.

## 1.2. Outline of the Research Contributions

The contributions of this research effort are summarized as follows:

1. New definition of ad hoc group, built on the previous definitions given in the existing literature on group research.
2. Design and development of the eConference tool for supporting both formal and informal synchronous text-based communication of ad hoc groups; design and implementation of a plugin for Mozilla Thunderbird to blend synchronous and asynchronous text-based communication in the email client.
3. Definition of a comprehensive theoretical framework, built upon the Task Circumplex model and the very many existing theories on CMC, and used to predict, evaluate, and compare the goodness of task-technology fits.
4. Design, performance, and analysis of an empirical study to gain insights into task-technology fit for supporting ad hoc group of stakeholders in the execution synchronous text-based requirements elicitation and negotiation tasks.

## 1.3. Outline of the Thesis Structure

The remainder of this thesis is structured as follows:

**Chapter 2** (**"The Role of Distributed Requirements Engineering in Distributed Software Development"**) discusses the challenges imposed by distance to Global Software Development, and to Requirements Engineering activities in particular.

**Chapter 3** (**"Group Research"**) provides a characterization of both issues encountered and techniques applied in group research, referring to ad hoc groups in particular.

**Chapter 4** ("**Computer-Mediated Communication**") frames the complex background of computer-mediated communication by reviewing the most prominent theories on media effect.

**Chapter 5** (**"Development of a Comprehensive Framework for Group, Task, and Media Factors"**) merges the contribution of Chapters 2 to 4, creating a general framework that consistently and comprehensively combines all the task-, group-, and media-related factors, relevant to the contextualization of this dissertation.

**Chapter 6** (**"Tool Support for Distributed Teams"**) reviews the research field of Computer-Supported Cooperative Work (CSCW) and the history of collaborative tools (groupware). The chapters also presents JabberPresence, a Mozilla Thunderbird plugin designed to bridge the gap between the asynchronicity of email, the most used and successful collaborative tool, and the synchronicity of IM and chat, the new means to foster and coordinate collaboration.

**Chapter 7** (**"The eConference Tool: History and Evolution"**) presents the history of eConference development: From the first generation, based on JXTA, through the second, based on XMMP and subjected to a pilot study from which feedback was gathered to implement the third and final version of the tool, built upon Eclipse RCP.

**Chapter 8** (**"Evaluating the Support of Synchronous Text-Based Media in Distributed Requirements Workshops"**) presents the empirical study, conducted at the University of Victoria, on the use of synchronous text-based communication in distributed requirements workshops.

**Chapter 9** (**"Conclusions"**) discusses the contributions of the dissertations, and presents the future work.

# Chapter 2:

# THE ROLE OF DISTRIBUTED REQUIREMENTS ENGINEERING IN DISTRIBUTED SOFTWARE DEVELOPMENT

---

## 2.1. The Benefits of Distributed Software Development

The last decades have witnessed a steady, irreversible trend towards business globalization, in particular, of software-intensive technology. Since the PC revolution in the 90s software development has become global, with a number of business factors also contributing to this trend. Indeed economic forces have relentlessly turned national markets into global markets, thus creating new forms of cooperation across national boundaries. This change has had, and is still having, a deep impact not only on marketing and distribution, but also on the way products, and software in particular, are designed, constructed, and delivered to customers [Her01a, Sen06].

In his book, Carmel listed the six main 'catalyst' factors, or potential benefits, which have driven to distributed (global) software development [Car99b]. The global demand

for software products and services that began in the 80s lead to a flood of (1) mergers and acquisitions, as IT firms strived to penetrate new markets and adjust or complement their products lines [Her00, Sen06]. Then, software firms began to aim at (2) positioning themselves as 'global organizations,' so as to signal the world they are 'global players,' selling products to global businesses and consumers. For instance, national policy of some countries, where the government may be a customer, requires suppliers to locate facilities in that country as a condition of sale or favorable tax treatment [Moc01, Her00]. In addition, it can make sense for market reasons to locate part of the corporation in (3) proximity to the market itself. The business advantages of proximity to the market include knowledge of customers and local conditions (e.g., product localization/customization, after-sale services), as well as the goodwill engendered by local investments [Ebe01, Her01a, Moc01]. Software companies that want to deploy the best software systems have to hire the most talented designers and developers in the world, regardless of their geographical location. 'Programming talent' (4) is not a pretentious term. There is in fact, strong evidence of huge differences in productivity between programmers. In [Hum97] Humphrey reported on a study conducted on 100 software engineers who were given identical specifications of 10 different programs. The results showed the fastest programmer to be 30 times faster than the slowest, while maintaining the same quality levels. Among the catalysts factors probably the most critical and strategic ones are (5) the reduction of development costs and (6) the reduction of time to market. Software companies in high-wage nations seek low-cost programmers in emerging countries (e.g., India, China, and several other nations in Central Europe and the Baltics), where software developers earn less than a half of what their counterparts do in the US. Already in 2001, 50 nations were participating in distributed software development, with India playing the role of leader, having more than 800 IT service firms that competed for work globally [Car01]. A study conducted in 2002 by Nasscom McKinsey reported that software development cost in India was four times less than in the US [Nas02]. For this reason, companies increasingly chose to focus on core competencies and outsource the other activities to specialized firms in those areas. Indeed, the study also estimated that the 10% of the workforce in the US IT companies was located in emerging markets and that the US's

savings from offshoring would grow from $6.7 billions to $20.0 billions by 2008. Offshoring (i.e., global outsourcing to contracting staff located offshore) brought in another benefit. Besides the savings that can be accrued through the access to large pool of cost-competitive skilled labor, it also offers the premise to further reduce costs by reducing the time to market [Sen06]. Global companies take an advantage from geographic dispersion: Since programmers are scattered across multiple sites, dispersion allows for round-the-clock, or follow-the-sun, development, which permits the reduction of development cycles by increasing the amount of time in a day that software is being developed [Her00].

Despite the premise of benefits, globalization has increased the challenges intrinsic to business due to distance. The remainder of this chapter is structured as follows. Section 2.2 discusses the negative effects generated by distance, focusing on communication issues. Section 2.3 introduces Requirements Engineering, as one of the most communication-intensive activities in software development, focusing on the elicitation and negotiation of requirements. Finally, Section 2.4 introduces Distributed Requirements Engineering and some of the tools developed to enable it.

## 2.2. The Negative Effects of Distance

Despite the premise of benefits described so far, distributed software development is fraught of substantial threats [Sen06]. Indeed, the success of a globally distributed project is not guaranteed by just opening a development center in another region of the world [Ebe01]. Developing software as a team is a challenging task, but developing as a distributed software team, that is, a team consisting of two or more sites separated by national boundaries, is even more challenging due to distance [Her01a].

Distance can be defined along three dimensions, namely geographical, temporal, and socio-cultural. *Geographical distance* and *temporal distance* are measures, respectively, of the spatial dispersion, occurring when team members are scattered across multiple sites, and temporal dispersion, occurring when members are in different time zones. Instead, *socio-cultural distance* is a measure of the differences in organizational and

national cultures among the sites (e.g., group's norms and practices, values, spoken languages).

When teams hand off process between sites, the lack of synchronization can be critical. An organizational unit cannot function properly without adequate communication, coordination, and control. Unfortunately, distance has a profound impact on all of them [Car99b]. *Communication* is the exchange between the members of information, whether formal or informal, occurring in planned or impromptu interaction. *Coordination* is that act of effectively orchestrating each task and organizational unit, so that they all contribute to the overall objective. *Control* is the process of adhering to goals, policies, standards or quality levels, set either formally (e.g., formal meetings, plans, explicit guidelines) or informally (e.g., team culture, peer pressure). Today, for knowledge workers, coordination and control have in many ways blended together, to the point where they are nearly inseparable [Car01]. The overhead of coordination and control associated with a new software project is astounding. Developers spend as much as 70% of their time working with others and as much as 40% of their time waiting for resources to be ready or available [Car99b]. The cost of controlling and coordinating activities increases when tasks are new or uncertain, and when units become independent as in the case of distributed software development. Distributed teams create in fact, further burdens on coordination and control mechanisms, primarily the informal ones. Because of distance, people cannot coordinate by just walking around and visiting the other team members. When control and coordination needs of global software teams rise, so does the load on all communication channels available. The absence of ongoing conversation can also result in coordination and control issues, like misalignment and rework. In sharp contrast to the popular image of software developers as relatively introverted and isolated, they in fact, spend a large proportion of their time communicating. For example, in an empirical study of time used by developers in a large software engineering organization, Perry et al. reported that developers spent an average of 75 minutes each day in interpersonal interaction [Per94]. Software development, particularly in the early stages, requires much communication. In fact, software projects have two complementary communication needs. First, the more formal, official communications is used for

crucial tasks like updating project status, escalating project issues, and determining who has responsibility for particular work products. Secondly, informal 'corridor talk' allows team members to keep a 'peripheral awareness' of what is going on around them, what other people are working on, what states the various parts of the project are in, and many other essential pieces of background information that enable developers to work together efficiently. In collocated settings, communication is taken for granted and then, its importance often goes unnoticed. When developers are not located together, they have much less opportunities of communication. There is very convincing evidence that the frequency of communication generally drops off sharply with the physical separation among coworkers' offices and that the sphere of frequent communication is surprisingly small. In the seminal study of engineering organizations, Tom Allen developed a profound relationship between distance and communication [All77]. He reported that, when engineers' offices were about 30 meters or more apart, the frequency of communication precipitously dropped to nearly the same low level as people with offices separated by many miles. Kraut et al. found similar results for scientists [Kra90b]. In addition, they found that the rate at which scientists collaborated spontaneously with one another was also a function of the distance between offices, and that this effect was more powerful than the effect of working in the same field. To fight the reduction in opportunities of communication, distributed teams use a variety of communication technology including audio, video, and text (e.g., video and telephone conferencing, email). Nevertheless, in addition to reducing opportunities, distance increases the costs of communicating as well. Technical issues, the difficulty to deploy complex systems, and the slowness in starting conversations make communicating awkward. These issues suggest that communication tools have to be readily available and easy to use.

In this dissertation, the researcher will specifically investigate on communication in a distributed software development setting. The motivation stems from the crucial role that communication plays in the effective orchestration of successful global software projects. As Figure 3 shows, distance exacerbates coordination and control problems directly or indirectly through its negative effects on communication. In other words,

communication disruption due to distance further increases and aggravates coordination and control breakdowns [Car01].

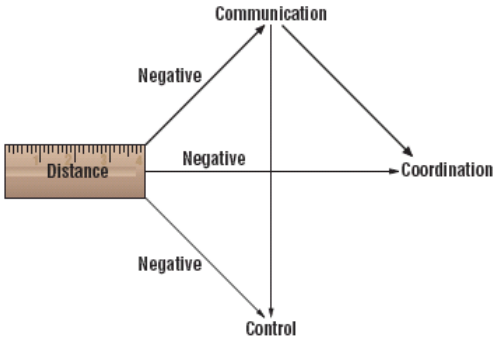


**Figure 3. Impacts of distance (from [Car01])**

Figure 4 summarizes the potential benefits and threats of distributed software development, discussed in this section. According to Carmel [Car99b], the threats are centrifugal, negative forces that pull distributed teams apart and inhibit the sense of 'teamness,' that is, the synergistic effect that makes a team a successful, cohesive unit (see Figure 5a). These centrifugal forces must be counterbalanced by centripetal, positive forces to make a distributed team productive and successful, thus realizing the potential benefits of distributed software development (see Figure 5b).

| | Temporal Distance | Geographical Distance | Sociocultural Distance |
|---|---|---|---|
| Communication | + Improved record of communications<br>- Reduced opportunities for synchronous communication | + Closer proximity to market<br>+ Access to remote skilled work forces<br>- Face-to-face meetings difficult | + Innovation and sharing best practice<br>- Cultural misunderstandings |
| Coordination | + Coordination needs can be minimized<br>- Typically increased coordination costs | + More flexible coordination planning<br>- Reduced informal contact can lead to lack of critical task awareness | + Greater learning and richer skill set<br>- Inconsistent work practices can impinge on effective coordination<br>- Reduced cooperation arising from misunderstanding |
| Control | + Time zone effectiveness can be utilized for gaining efficient 24x7 working<br>- Management of project artifacts may be subject to delays | + Communication channels can leave an audit trail<br>- Difficult to convey vision and strategy<br>- Perceived threat from training low-cost "rivals" | + Proactiveness inherent in certain cultures<br>- Different perceptions of authority can undermine morale<br>- Managers must adapt to local regulations |

**Figure 4. Potential benefits (+) and threats (-) of distributed software development (from [Age06])**

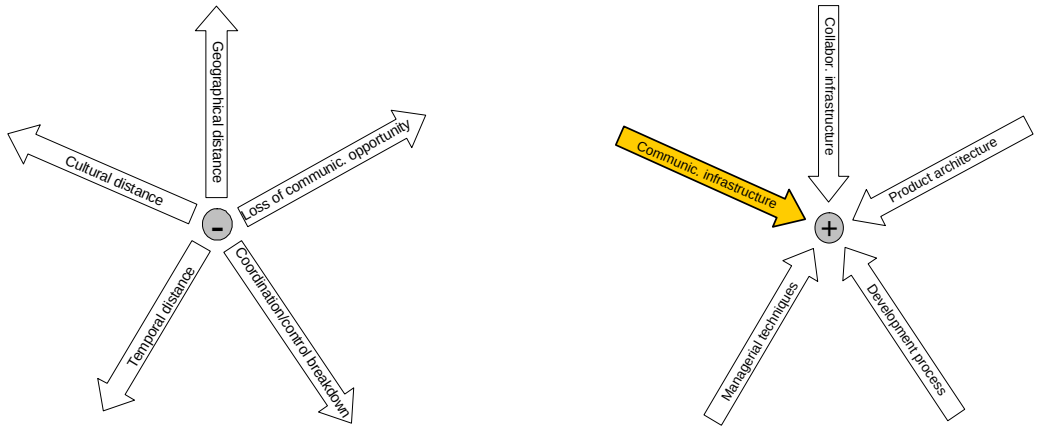


**Figure 5. (a) Centrifugal forces that threaten distributed software development and (b) centripetal forces to contrast them (adapted from [Car99b])**

Carmel & Agarwal discussed some tactics and arrangements to alleviate distance problems [Car01]. Socio-cultural distance can be reduced by applying several managerial techniques, namely the *offshore-onshore bridgehead* (i.e., having 75% of personnel work occur offshore, and the remaining 25% occur onshore), the *cultural liaison* (i.e., a manager or key executive who travels back and forth between the sites), teaching a language to be used as *lingua franca* between remote sites (typically English), and, finally, *internalizing* the offshore firms (i.e., acquisition, create a joint venture). Geographical distance can be alleviated by both reducing the need for intensive collaboration between organizational units through the right allocation of tasks and employing a collaborative infrastructure. As collaboration across sites increases, so do the cost and complexity of coordination and control. Hence, distributed software teams try to coordinate and control development through a number of mechanisms, including the use of particular types of architectures, design and development processes that minimize the need for cross-site interaction and dependency [Her99]. To do so, they need a 'collaborative development environment,' such as SourceForge[1] and GForge,[2] which provide a virtual space where all team members, especially when distributed by time or distance, may work together to carry out some

[1] http://sourceforge.net
[2] http://gforge.org

task, typically a software system and its supporting artifacts [Boo03]. Programming tools aside, these environments typically comprise configuration management systems, such as CVS and SVN, to manage and integrate the changes to source code in a controlled manner, issue trackers, for the tracking of bugs and defects, mailing lists, to communicate and archive decisions, and other tools to help managers keep project status under control by the opportune scheduling of tasks.

Temporal distance can be reduced by analyzing the advantages and disadvantages of synchronous and asynchronous communication tool, thus finding the right trade-off with respect to the time zones disparities. Among the centripetal forces, this dissertation focuses on studying synchronous communication in particular. While asynchronous communication can be helpful to reduce language and cultural problems, synchronous communication is preferable when rapid feedback is needed, even with a few overlapping business hours. The advantages of synchronous communication include resolving miscommunication, misunderstandings, and small problems in a timely manner, before they become bigger. Asynchronous interaction would introduce even more delay in the resolution of problems of distributed software development, which has already been proved to be substantially slower than collocated [Her01c].

## 2.3. Requirements Engineering

Requirements Engineering (RE) is one of the most communication-intensive practices in software engineering. Requirements engineering is accepted as one of the most crucial stages in software design and development, as it addresses the critical problem of designing a set of processes that operates on different levels. The development of a system requirements specification is widely recognized as the basis of system functionality. Software requirements are the critical determinants of software quality. It has been shown that errors in requirements are the most numerous, expensive and time-consuming in the whole software life cycle [Boe01c]. Correcting software defects in the maintenance phase can require up to two-hundred times the effort it would take if the correction was implemented during the requirements specification phase [Dav93].

Requirements are descriptions of how a software product should perform. The IEEE 610.12-1990[3] standard defines a requirement as a documented condition or capability, needed by a user to solve a problem or achieve an objective, and met by a system to satisfy a contract, a standard, or a specification. Therefore, requirements include needs arising from organizational, government, and industry standards, as well as from the end users and the various other stakeholders. A stakeholder is generally defined as a participant to the development process whose actions can influence or be influenced by the development and use of the system, whether directly or indirectly [Pou97]. Typical stakeholders are product managers, the various types of users and administrators from the client side, and the developers from the software development side.

The definition of software requirements is an interdisciplinary task that involves stakeholders with a different background and understanding. As software projects became increasingly complex, software developers face the challenge of identifying the goals of stakeholders who come from a diverse range of backgrounds. It may also be very difficult to represent the essential requirements in a way that is accessible to all stakeholders involved [Nus00]. The importance of stakeholder involvement in requirements engineering activities is widely accepted, given that an accurate identification of stakeholders' needs largely determines the quality of the software products [Aur05].

Requirements engineering refers to all software life-cycle activities related to requirements. Literature provides many process models to describe the requirements engineering process. Although these models encompass common requirements engineering activities, they differ in nature. They often depict the process as linear progression of activities [Kon98], but, sometimes, the process is represented by an iterative [Lou95] or even spiral model [Boe88]. The common limit of these models is that they represent normative models that tend to describe how the process should work, rather than how it does work in reality. Figure 6 shows the linear model of common requirements engineering activities proposed by Macaulay [Mac96]. Macaulay also points out that a validation process takes place at the end of each phase and describes the model as follows. The (1) product conceptualization triggers the requirements

[3] http://standards.ieee.org/reading/ieee/std_public/description/se/610.12-1990_desc.html

engineering process and the concept moves into a specific project. During the (2) requirements elicitation phase, the nature of the problem is analyzed and an understanding is developed. Another key aspect of requirements elicitation is generating an appropriate representation of the problem, which can help stakeholder to identify the set of alternative solutions. The (3) requirements negotiation phase is concerned with evaluating the feasibility, and the costs and benefits of the alternative solutions identified during the elicitation phase. In the fourth phase a (4) detailed analysis and modeling of requirements is provided. Finally, once this process is finished, the (5) requirements specification document can be completed.


**Figure 6. Macaulay's linear process model for RE activities (adapted from [Mac96])**

In the remainder of this dissertation we will implicitly refer to Macaulay's model to epitomize the activities in the requirements engineering process. However, the selection of the model is not of vital importance as the objective of this research is to study in particular the communication issues that affect stakeholders during the elicitation and negotiation activities.

### 2.3.1. Requirements Elicitation

Requirements elicitation is the process of seeking, uncovering, acquiring, and elaborating requirements for computer-based systems [Zow05]. Process of eliciting requirements is generally accepted as one of the critical activities in the requirements engineering process. Getting the right requirements is considered a vital but difficult part of software development projects [Jon96]. Requirements elicitation is concerned with learning and understanding the needs of users and customers with the ultimate aim of communicating these needs to the system developers. A substantial part of the elicitation is dedicated to uncovering, extracting, and surfacing the needs of the potential stakeholders.

As the field of requirements engineering began to develop, researchers and practitioners identified that the elicitation of requirements for software-based systems had some unique and complicated characteristics, and therefore, needed to be addressed as a new and separate topic from traditional knowledge acquisition [Dav94]. As a result, attention was directed to the development of specific tools and techniques to support this process in the hope of reducing its complexity and resolving some of the key challenges in its execution.

Despite the existing differences in the requirements engineering process models, requirements elicitation is generally accepted as the initial stage of the whole requirements engineering process. Typical activities of the requirements elicitation process involve understanding the application domain in which the system will reside, identify the existing sources of requirements and involve the relevant stakeholders, selecting the approach or technique to use, and, finally, elicit requirements from the stakeholders and other sources. Much of the research and practice conducted in the field of requirements engineering has been directed towards improving the elicitation process through the definition and application of various techniques. Many of these methods, such as interviews [Aga90], questionnaires [Fod94], and introspections [Gog93], have been borrowed or adapted from social sciences [Cou04], and only a few have been specifically developed for eliciting software requirements, such as Class Responsibility Cards (CRC) [Bec89] and prototyping [Som01]. Group work (i.e., collaborative meetings) is the most common and often default technique for requirements elicitation [Zow05]. Group meetings are particularly effective since they promote cooperation by directly involving and committing the stakeholders. Requirements workshop is the generic term used for indicating different types of group meetings (e.g., for both elicitation and negotiation), when the emphasis is on specifying software requirements of a system [Got02]. An example of less formal workshop is brainstorming. *Brainstorming* is a process where stakeholders from the different groups engage in informal discussion to rapidly generate as many ideas as possible [Osb79]. The main advantage in adopting brainstorms is the promotion of freethinking and expression, which foster the discovery of solutions to existing problems. *Joint Application Development (JAD)* meetings are an example of more formal requirements workshops

[Woo95]. The major difference between JAD and brainstorming is that the former is a well structured workshop, whose goals, steps, actions, and roles of participants (including a specialist facilitator) have been established in advance. These differences notwithstanding, the mostly used techniques for eliciting requirements are based on group communication. Indeed, requirements elicitation is a multifaceted activity that relies heavily on the communication skills of requirements engineers, and the commitment and cooperation of the system stakeholders. Problems of requirements elicitation can be grouped into three categories: (1) problems of scope, in which the requirements may address too little or too much information; (2) problems of understanding, within groups as well as between groups, such as users and developers; (3) problems of volatility due to the evolving nature of requirements. While volatility is an intrinsic problem of the definition of requirements, the first two stem from communication issues. One of the main problems in requirements elicitation is communicating and agreeing about the requirements. The main point is that concepts that are clearly defined to the problem owning party (i.e., the customers and the stakeholders) can be entirely opaque to the problem solving party (i.e., the system developers) and vice versa. When attempting to engage in meaningful dialogue, such a 'culture gap' between the parties makes articulating and understanding requirements particularly difficult [Nus00]. In some cases this maybe a result of the analysts and stakeholders not sharing a common understanding of concepts and terms, or the analyst is unfamiliar with the problem. More often stakeholders understand the problem domain very well, but are unfamiliar with the existing solutions and the way in which their needs could be met [Zow05].

### 2.3.2. Requirements Negotiation

Communicating and agreeing on software requirements requires a constant interplay among idea generation and conflict resolution [Mac96]. Agreeing on requirements by resolving conflicts or misunderstandings is a fundamental problem in requirements engineering, due the different and often conflicting goals and priorities that stakeholders have [Nus00]. Conflicts play a key role in software engineering and in requirements negotiation in particular [Cur98]. Given the highly-collaborative nature of software

development, conflicts inevitably arise as the project stakeholders pursue mismatching or even opposite goals [Boe00b]. Thus, shared or opposed views have to be identified and then, reconciled to develop mutual agreement, so as to ensure the project success [Kon96, Sut02]. Obviously, stakeholders will not always be in agreement in the end. However, identifying disagreements is also a result of the negotiation phase as they represent serious risks that have to be carefully addressed by project management [Boe00, Gru05b]. Conflicts have been regarded as destructive for a long time, and software engineering process in general either disregarded their presence or concentrated on their elimination and avoidance [Dam01]. Curtis et al. were among the first to suggest that the confrontation of divergent views can produce new perspectives and more comprehensive views, leading to superior decisions. Consequently conflict began to be regarded as beneficial to a 'more complete' system analysis and design [Cur98]. Negotiation has been defined as a collaborative approach to resolving conflict by exploration of the range of possibilities, in the attempt of finding a settlement which satisfies all parties as much as possible [Eas91]. In fact, requirements can be regarded as constraint on the solution space, which changes its shape as new requirements are formulated. Thus, requirements negotiation becomes a negotiation of such constraints on the solution space [Eas94].

The negotiation process is typically considered to be the interactions among stakeholders, which start when participants begin to communicate their goals and end (successfully) when all agree to a specified contract [Rob98]. According to Curtis et al., requirements negotiation is a process through which stakeholders make tradeoffs between requested system features, the capability of existing technology, the delivery schedule, and the cost [Cur98]. It has been argued that the negotiation process comprises also the execution of pre-negotiation and post-negotiation phases, respectively before and after the actual conduct of the negotiation [Rob98]. During the pre-negotiation phase, the activities to execute include the definition of the negotiation problem, the identification and solicitation of stakeholders, whose goals are collected and analyzed to find conflicts. Thus, during the negotiation phase, on the basis of the collected goal and identified conflicts, stakeholders seek mutually beneficial solutions. Finally, in the post-negotiation phase, the outcome of the negotiation is analyzed and

evaluated to ensure the quality of proposed solutions [Gru05b]. An alterative to this linear model of the negotiation process is provided by the WinWin spiral model, in which negotiation is used early on and then, repeated in later stages [Boe98].

Negotiation is a conflict-laden process. Occurrence of conflicts is inevitable in any collaborative setting: The inherent differences between individuals' experiences, personalities, and commitments make the potential for conflict inherent to any group of people [Nus00, Dam03a]. Requirements negotiation is fundamentally a task involving decision-making and the resolution of technical as well as social and behavioral issues. Thus, researchers have adapted and applied to requirements engineering the existing approaches and techniques on effective decision-making and conflict resolution. Decision-making and conflict resolution are impaired by communication breakdowns: If the different interest groups involved do not communicate effectively with each other, each group will see the other as attempting to exert power and influence unreasonably [Mac96]. Thus, for successfully running these activities, group meetings are regarded as the most effective means, as they ensure the contemporary commitment of all the parties [Kon98, Dam01].

## 2.4. Distributed Requirements Engineering

The previous sections of this chapter have shown that distance has a high impact especially on development processes which rely heavily on the interaction. Hence, being one of the most communication-intensive activities in software development, the effectiveness of requirements engineering is greatly constrained as its intrinsic challenges are exacerbated by the geographical distance existing between stakeholders in global software development. Nevertheless, research has only recently begun to focus on requirements engineering during global software development [Dam03b]. Indeed, initially multi-site projects tended to perform requirements engineering activities during face-to-face requirements workshops. It was found that group meetings account for 65% of the communication channels in requirements engineering [AlR96]. However, allowing relevant stakeholders to travel and attend collocated workshops has become impractical. First, not all organizations can afford the costs of arranging face-to-face

workshops on an ongoing basis [Dam03a]. Secondly, case studies revealed that, when it comes to selecting participants for such workshops, managers tend to strike a balance between allowing developers to talk to 'the right people' and maintaining a smooth running of the rest of the business [AlR96, Dam01]. Thirdly, it has been shown that, when distributed, other software process, such as remote software inspection [Lan03b] and architecture evaluation [Bab06] can be improved with adequate tool support.

This sheer need for group communication during the requirements definition process stands in sharp contrast with the increasing impracticability of conducting collocated requirements workshops during global software development, thus leaving space and motivation to conduct research on requirements engineering in global settings.

### 2.4.1. Tool Support for Distributed Requirements Workshop

While computer-mediated communication can be synchronous as well as asynchronous, the importance of distributed requirements workshops, in which all relevant stakeholders are allowed to participate despite distance, translates into the need of designing effective support tools [Dam03b]. The need to develop an infrastructure to support communication in teams of geographically-dispersed stakeholders plays a key role for coping with the lack of physical proximity when developing requirements. In this section, we briefly review the most relevant research and commercial tools available for conducting effective distributed requirements workshops.

*EasyWinWin*[4] [Boe01a] is a requirements negotiation approach that combines the win-win spiral model with Group Support System (GSS) collaborative knowledge and automation techniques. GSS is not a single piece of software, but a collection of computer-based collaborative tools (e.g., brainstorming tool, voting tool) extensively used during the 90s by distributed teams to collaborate from a distance (see [Fje00] for an exhaustive compendium of GSS field research). EasyWinWin is a comprehensive platform because it supports all the activities of the RE process through both synchronous and asynchronous text-based interaction.

[4] http://sunset.usc.edu/research/WINWIN/EasyWinWin/

*TeamWave*[5] has been used by Herlea & Greenberg [Her01d] to develop a requirements engineering collaboration space for facilitating interaction of distributed stakeholders. TeamWave offers shared whiteboarding, post-it notes, bulletin boards, note organizers, brainstorming tools, voting tools, action-item organizers, and text-based chat.

*CRETA* (Cooperative Requirements Engineering Support Tool) is a system developed by Togneri et al., which supports and integrates the work of the knowledge managers, requirements engineers, domain specialists, users, project managers and sponsors in one platform [Tog02]. The goal of the system is to support the main activities of the Requirements Engineering process and to promote mechanisms for sharing information, and facilitating communication, coordination and cooperation among people, as well as awareness and knowledge management. The participants make use of the cooperative tools available – such as electronic agendas, electronic mail and appointments, discussion lists, synchronous (chat) and asynchronous (forum) virtual meetings.

Lanubile [Lan03a] has developed a toolset for distributed requirements elicitation in Groove,[6] a peer-to-peer platform intended for communication, content sharing, and collaboration. The inceptive idea behind this toolset is exploiting a decentralized architecture to support the key activities of global software development. It includes tools for the different tasks of the elicitation phase, including synchronous text-based workshops, questionnaires, and a voting system. Groove Networks has been acquired by Microsoft, which reused its underlying technology to inject distributed collaboration capabilities in the Office suite (also known as Groove Virtual Office).

All the tools and platforms described so far in this section come from academia and are specific for the requirements engineering field, although they rely on text-based communication channels to enable the running of distributed requirements workshops with many participants. Instead, to enable distributed requirements workshops with multipoint audio/video support, research has usually run experiments using general

[5] http://www.markroseman.com/teamwave/
[6] http://www.groove.net/

purpose, commercial conferencing systems, such as Microsoft NetMeeting[7] [Dam03a, Dam03b] and Centra Symposium[8] [Llo02].

## 2.5. Summary

In this chapter we have shown that, due to economic factors, the general tendency to the globalization of business is transforming software development in a distributed activity. We have also shown that, besides the economic gains, distributed software development has to face many difficulties and challenges caused by distance. In fact, distance (temporal, geographical and cultural) causes a substantial reduction of the opportunities to communicate, both formally and informally, with distant team members. Being one of the most communication-intensive activities of the software development lifecycle, Requirements Engineering is deeply affected by distance. In this chapter we focused on analyzing the elicitation and negotiation of software requirements, since these tasks are typically executed bringing the stakeholders together in collocated meetings, which are hard to be arranged in distributed settings.

[7] http://www.microsoft.com/windows/netmeeting/
[8] http://www.centra.com/products/symposium/info.asp

# Chapter 3:

# GROUP RESEARCH

## 3.1. Methods for Group Research

The study of groups has been of interests for social psychologists for a long time, since groups tend to exhibit performances that their individual members might never manifest, acting on their own [McG84]. There are numerous methods for studying group behavior. All empirical research methods suffer from the dilemma of having inherent shortcomings, as well as certain advantages. According to McGrath [McG93b]:

- All methods allow conclusions, yet all methods have limited scope.
- All methods have errors, yet all methods are useful.
- Errors can be corrected by successive applications of methods.
- Different methods should be combined, so that one method's shortcoming can be corrected by another method's advantage.

The study of group work encompasses three basic questions [Bor00]:

(A) Who are the subjects involved?

(B) What behavior is to be examined?

(C) What is the current and realistic context of the group work?

In particular, the first question (A) aims at generalizing the results on the type of the subjects (group members) involved in the study, whereas the second (B) aims at providing data on group activities to be studied, as accurately as possible. However,

measurements that increase the accuracy of data on a given behavior often tend to influence the situation under examination, thus making it "more artificial".

McGrath classified the research methods along two dimensions, namely *behavior universality*, that is the general validity of study results, and *study obtrusiveness*, that is the influence of the study itself on the results. Figure 7 shows existing research methods classified along the two dimensions.

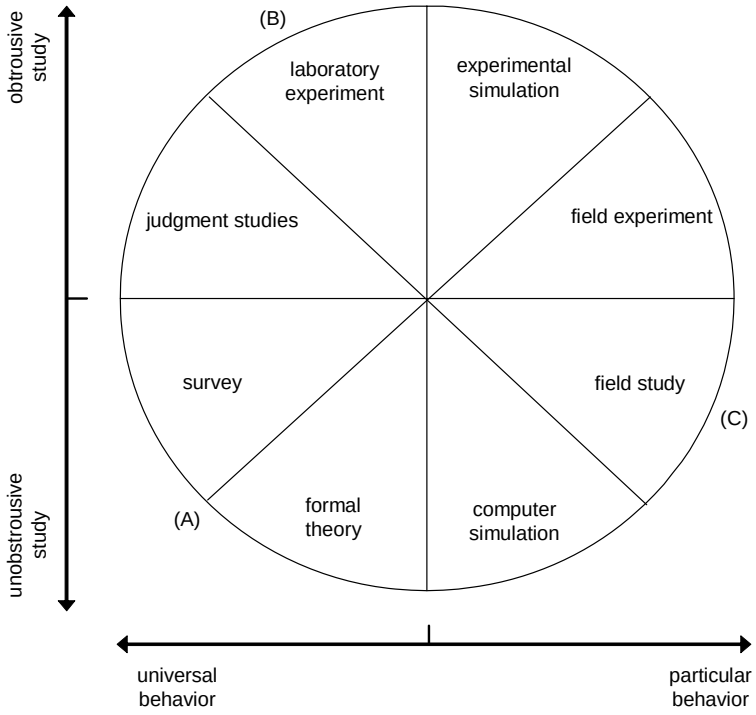


**Figure 7. McGrath's classification of research methods (adapted from [McG93b])**

The figure sopra shows that each research study that focuses more on one of the three basic questions for group work research is inevitably far from being optimal with respect to the two remaining aspects. Extreme solutions are those in correspondence of the studies marked with the letters (A), (B) or (C). In the following we review the methods relevant to group research, focusing on laboratory experiments in particular.

Field studies focus entirely on question (C) in that they are observations on natural work groups, which do not modify the real group context in any way. Ethnographic

methods, originated in social studies for studying ethnic groups, are the preferred approach for field studies because they provide results both on real work practices and on social structures at work (tool used, direct view of group members). Hence, field studies are highly realistic with respect to the authenticity of the situation (C), but conversely have less result generalizeability (A) and data accuracy (B) [Bor00].

Field experiments are similar to field studies, except that researchers actively influence the group process and alter certain conditions to obtain information on their effects. Laboratory experiments, instead, attempt to recreate a group environment where researches can control all external conditions in order to explore specific questions. Hence, laboratory experiments are suitable for exactly determining behavioral patterns, in that they maximize measures accuracy (B), but, at the same time, lose generalizeability (A) and realism (C). Experimental simulations are laboratory studies which attempt to mirror exactly a real-life situation or system. They are artificial in that the system or situation is created only for the purpose of the research and subjects are observed while performing contrived tasks [Bor00].

Surveys aim at collecting information from several subjects on a fixed set of issues. The choice of subjects to be surveyed is typically based on certain criteria relevant to the research. Surveys are highly generalizable (A), however, the accuracy of measurement behavior and real-life authenticity are low, since there is no guarantee that subjects answered truthfully [Bor00].

In Chapter 1 we have already highlighted how group research is affected by the contextual factors of media, task, and group, which act on the group interaction process, and, consequently on the outcome. This causal relationship is graphically depicted in Figure 8. Through the rest of the chapter we will update the graphical representation to include those characteristics of each of the three factors that are relevant to explicitly define the context of the overall research study discussed in this dissertation.

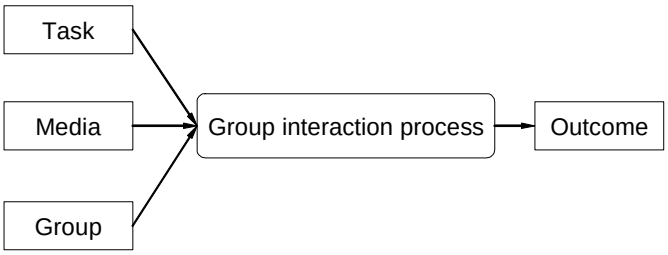


**Figure 8. The three contextual factors acting on group interaction process and outcome**

The remainder of the chapter is structured as follows. In Section 3.2 framework for tasks classification are discussed. Section 3.3 outlines the dynamics of group development and group members' interaction. Finally, the group history factor is presented, along with ad hoc groups, that is, groups with limited temporal scope. The theories on media effects are discussed in Chapter 4, whereas the interaction of media with both task- and group-related factors is presented in Chapter 5.

## 3.2. Task Classification Frameworks

When differences in group task performance are studied, differences in tasks must be taken into account with the due regard. A number of task classification schemes have been proposed in the literature, such as Wood's *Model of Task Complexity* [Woo86], and Mennecke's *Model of Task Processing in Groups* [Men93] (see [Zig98] for a comprehensive list). However, the most prominent theoretical framework formulated to provide a classification of group tasks is McGrath's Task Circumplex [McG84]. *Task Circumplex* classification scheme draws upon Hackman's *Task Framework* [Hac69], which defined three types of tasks, namely tasks of idea production, tasks of discussion for group consensus, and tasks of problem solving. In addition, McGrath's classification considers a task to be characterized by its own objective (i.e., what the group members are supposed to do to accomplish it), including not only *what* must be done, but also *how* to do it. Task Circumplex, shown in Figure 9, categorizes all group tasks as belonging to one of four basic task processes, each of which has in turn two subtypes: (I) Generate (ideas or plans); (II) Choose (correct or preferred answers); (III) Negotiate (conflicting viewpoints or conflicting interests); (IV) Execute (in competition against

other groups or in evaluation against standards of performance). The four process categories are related to one another and arranged in a circumplex along two dimensions, namely the degree to which processes involve cooperation (i.e., low task interdependence) versus conflict (i.e., high task interdependence), and the degree to which the processes involve conceptual versus behavioral activities. Furthermore, McGrath designed the four process categories to be mutually exclusive, collectively exhaustive, logically related, and useful for comparing similarities and differences of tasks used in group research.

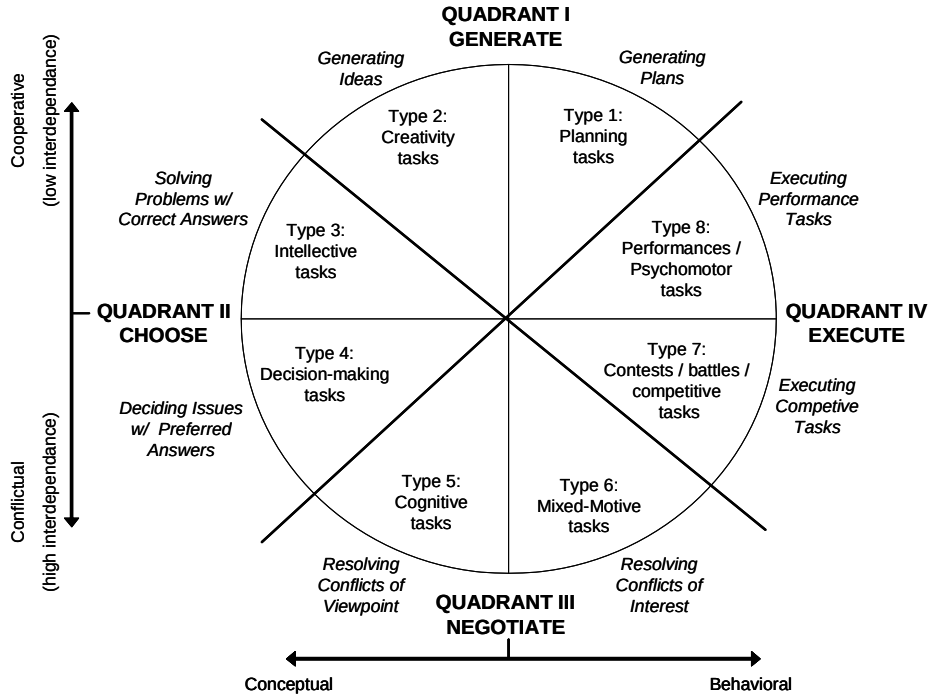


**Figure 9. The Task Circumplex (adapted from [McG84])**

As an example, we categorize the activities of requirements elicitation and negotiation described in Section 2.3.1 and Section 2.3.2, respectively. According to the framework sopra, eliciting requirements is almost exclusively a creativity task (Type 2), since it is about generating ideas, with very limited need for decision making and problem solving. Conversely, the negotiation of software requirements involves tasks of Type 3 to 7, namely creativity, intellective, decision-making, cognitive, mixed-motive,

and competitive tasks [Dam01]. Thus, comparing the two forms of requirements workshops, in Task Circumplex terminology, requirements negotiation is a more complex activity, in that it involves different tasks, both conceptual and behavioral, with medium-to high degree of member interdependence. In contrast, elicitation is a simpler activity in that it is only a conceptual task of creativity, with low behavioral issues involved and low degree of member interdependence.

Task Circumplex is not exempt from limitations and criticisms. While it gives a way to compare tasks, it does not provide with an objective means to measure the degree to which tasks in each wedge differ from tasks in both the same category or in different categories [Men93]. Despite its widespread adoption, however, Task circumplex has been the dominant task-classification scheme in the last two decades. It has been used not only as a task taxonomy, but also as the foundations to develop theories on communication media selection (see Chapter 4), which encompass the intertwined relationships between tasks and technology, discussed in Section 5.2. Task Circumplex has been adopted by Group Support Systems (GSS) research (see [Fje00] for an exhaustive compendium on GSS-related research studies). GSS studies have largely dominated group studies for almost more than two decades, until the end of the '90s. Christenesen & Fjermstad performed a meta-analysis of 67 GSS studies, conducted until 1997 [Chr97]. They found that more than a half of GSS studies employed creative tasks and that more than one-quarter employed decision-making tasks. Furthermore, most of the laboratory studies reviewed used contrived tasks designed or manipulated for the research purpose. To improve the generalizability of results, Dennis et al. called for the use of tasks as complex as 'natural tasks,' requiring knowledge already within subjects knowledge domain [Den90]. However, since students were and are likely to continue as the most common source of experimental subjects, the usually contrived laboratory tasks were puzzles or games (e.g., lost at sea, the parking problem, the philanthropic foundation task [Men93]), which required limited or no specialized knowledge to be recalled [Mur00]. These tasks represent a poor surrogate for the complexity of 'wicked' natural tasks, and their employment potentially limited the external validity and generalizability of GSS laboratory experiments, and likely accounted for much of the contradictory findings between field and laboratory research

[Den91, Den93, Men93]. The multi-faceted properties and complexity of natural tasks can be achieved by using 'realistic tasks,' that is, natural tasks replicated in controlled laboratory environments. The flipside of using realistic tasks in place of contrived tasks is the likely higher difficulty in evaluating group interaction processes and task performance. Effectiveness does not have a consistently held definition or interpretation in the group research literature [Nun91]. Satisfaction with both the interaction process and the outcome is an important variable in group research, since it has been acknowledged to be indicative of both individual and group performance [Hol93, Ben93].

## 3.3. Group Dynamics

Group task is only one of the fundamental aspects in the theoretical foundation of group research. The other lies in the inner dynamics of groups. Group dynamics is the field of study within the social sciences that focuses on the nature of groups and the way groups and individuals act and react to changing circumstances.

The most frequently cited model of "stages in group development" is Tuckman's *forming-storming-norming-performing* (also referred to as *stages model*), presented in 1965 in the classic article "Developmental Sequences in Small Groups" [Tuc65]. While his research was based on therapy groups, Tuckman argued that groups in any setting are likely to go through four distinct stages as they come together and begin to function (see Figure 10).

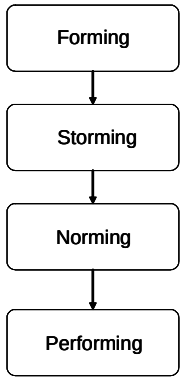


**Figure 10. Tuckman's stages model**

In the first stage, labeled as *forming*, groups initially concern themselves with orientation to identify the boundaries of both interpersonal behaviors (e.g., establishment of dependency relationships with leaders) and task behaviors. The second stage, labeled as *storming*, is characterized by conflicts on interpersonal and task-related issues. These behaviors serve as resistance to group influence and task requirements. Resistance is overcome in the third stage, labeled as *norming*, in which in-group feeling and cohesiveness develop, and task-related personal opinions are expressed. Finally, the group attains the fourth and final stage, labeled as *performing*, in which group energy is channeled into the task execution.

Tuckman's model has been subjected to several critiques, mostly coming from the strict linearity of the four consecutive stages in the model, whereas in fluid activities involving human beings it is hard to find such a clear-cut demarcation, as changes do not occur in a discontinuous, step-like sequence [Smi05]. Bales argued that group members tend to seek a balance between resolving conflicts, building interpersonal relationships, and accomplishing the task [Bal65]. The result is a continuous movement between storming, norming, and performing. Accordingly, Smith has proposed a representation of Tuckman's original, 'linear' model as a cyclical model with the same phases, but also allowing stages to recur at different points in a group's life (see Figure 11).

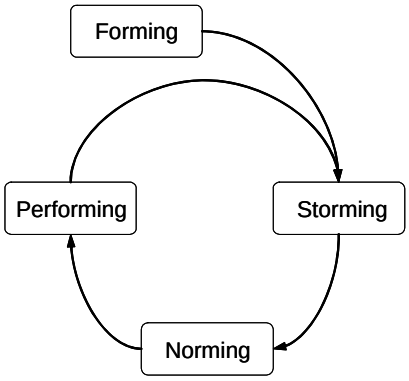


**Figure 11. Tuckman's stages revisited as a cyclical model (adapted from [Smi05])**

Building upon the Interaction Process Analysis [Bal51], Bales & Cohen developed SYMLOG, a methodology for observing, coding and analyzing group interactions in small, problem-solving groups [Bal79]. While Tuckman's model focuses on the dynamics of group development, Bales & Cohen's model, instead, focuses on the dynamics of group interaction. *SYMLOG* methodology is based on the assumption that there are three fundamental processes that underlie the interaction of group members: (1) task-oriented behaviors that deal with individual's concern of group task performance; (2) socio-emotional behaviors associated with interrelationships of members; (3) interpersonal behaviors associated with the influence/acquiescence of dominant/submissive members. This framework provides a basic classification scheme of group behaviors to draw upon when categorizing messages exchanged during group interactions (e.g. meeting).

## 3.4. Teams with No Past and Future, or Ad Hoc Groups

Besides task type, another contextual factor that influences group studies is temporal scope, that is, "the extent to which groups have pasts together, and expect to have a future" [McG91, p. 149].

Work groups are today increasingly nimble and subject to frequent changes [Hau05]. This underlying idea in *ad hoc* groups is that of a small entity highly dynamic in creation, participation, and release, formed to accomplish the goal at hand (e.g., solve a specific problem), and then, disband as soon as the collaboration is over. Hence, ad hoc teams are also called *goal-oriented* teams [Bor00]. These teams are sometimes associated with strike teams, which are small groups of people with a specialized purpose, such as responding to a critic situation, like a terrorist attack or a natural disaster, in a timely manner. In addition, ad hoc groups typically exhibit both loose affiliation and geographical dispersion (see Section 2.1), i.e., they are virtual teams, composed recruiting members from independent departments in different organizations [Hef04]. Virtual organizations of the future will be more and more comprised of flexible, ad hoc groups that individuals join when they can add value and disengage

when they are no longer needed [Kno95]. Today, a common scenario of ad hoc groups collaboration is provided by the partner consortium formed by representatives from different organizations in various sectors (e.g., academic institutions, industry), who have to co-author a funding proposal for applying to the Framework Programme of the European Commission. Also in the field of software development several processes, such as document inspections and reviews in general, can be carried out by ad hoc groups [Lan03b]. The scenario of distributed requirements described in Section 2.3 provides another example of a dynamic collaboration that can be accomplished by a virtual, ad hoc group, where some members (e.g. representatives from the customer organization) join the developer group, when they can add a value (e.g., to take part in the elicitation of the requirements, in a prototype demo session), and disengage at the end of the task.

The limited group size and temporal scope are the key characteristics of ad hoc groups. Ad hoc groups do not usually include more than 10 participants. However, every attempt to define the typical size is vain. Even research on small groups reports varying ranges, usually 3-5 participants for small-sized groups, and 6-12 for medium-sized groups [Dav95]. However, in absence of a widely accepted definition of group size, these ranges can be considered reasonable, bearing in mind the research already undertaken. The study of small- and medium-sized groups is important because it has been shown that larger groups do not necessarily produce a proportionally higher number of ideas and thus, there is likely to be an optimal group size, beyond which any further increase in membership does not equate with an increase in contributions [ibid.]. Temporal scope defines group history and future, that is, the shared experience that the group has developed in the past and the expectation of future collaboration, respectively. For ad hoc groups, temporal scope corresponds exactly with the time needed to carry out one collaboration. In other words, while traditional groups are conceived as *established*, i.e., long-term, standing teams that work together for a long time, across several independent projects, ad hoc groups are instead teams brought together for a short time to carry out only the collaborative effort in attendance. The meaning of “ad hoc groups” today differs greatly from the earlier definitions provided by researchers over the years. Ad hoc groups, also called single-task groups initially,

have been studied since the end of the'50s [Lor58, Hal66, Bor70] and over the last decades [Den90, Men95, Alg03, Bir05b]. According to the definition given by Mennecke et al. ad hoc groups are teams whose "members have no experience working together with other members and little or no expectation that they would work together in the future." In contrast, they defined established groups as "on-going groups, that is, groups where members have a significant history working together as a group and anticipate having a significant future together" [Men92, Men95]. Likewise, Dennis et al., defined ad hoc groups as single-task groups whose members have not worked together prior to the study and do not anticipate to continue working together after the study [Den90]. Although similar to the others, this definition is indicative of how past research considered ad hoc groups as single-task, "laboratory groups" of randomly assembled subjects to be studies merely as "experimental, microscopic models" of established groups, seen instead as natural groups [Lor58]. However, Bormann [Bor70], McGrath [McG84], and Mennecke et al. [Men92] pointed out the inadequacies associated with using single-tasks groups, in terms of the lower generalizability of results. Nevertheless, single-task groups have almost universally been used in laboratory experimentation, compared to field studies, where established groups are utilized instead.

Whilst previous research has almost exclusively treated ad hoc groups as a factor partially accounting for discrepancies between laboratory and field studies, current research cannot continue to neglect the relevance of studying ad hoc group *per se*. We cannot continue to refer to established groups as "natural groups," since nowadays ad hoc groups are functionally used as well, and no more employed only in laboratory studies. While established groups are still more traditional, they are to be considered *as natural as* ad hoc groups. We suggest to adopt the definitions given by McGrath et al. to distinguish *natural* groups, defined as "groups that exist independently of the researcher's activities," which are used in field experiments, from *concocted* groups, which are instead "brought together only for the purpose of laboratory experiments" [McG84, p. 41]. Thus, group research studies can employ natural as well as concocted ad hoc groups. In addition, compared to concocted established groups, laboratory studies on concocted ad hoc groups will suffer from minor problems of results

generalizability, since they represent a more adequate experimental model of their natural counterpart. We also suggest a new definition of ad hoc groups.

***Definition.*** *An ad hoc group is a small- to medium-sized team highly dynamic in creation, participation, and release, whose members have no past experience of working together and little or no expectation of collaborating again in the future, and temporal scope corresponds exactly to the time needed to carry out the collaboration in attendance.*

The definition above voluntarily omits the adjective 'distributed,' typically used to further characterize an ad hoc group, because while virtual ad hoc teams are more common and of our primary interest, there can be collocated ad hoc groups as well.

## 3.5. Challenges and Needs in Supporting Distributed Ad Hoc Groups

Our specific interest in supporting collaboration of ad hoc groups is two fold. We aim at understanding (1) the key challenges in ad hoc group communication processes and (2) the attributes of the technology to use in order to cope effectively with such challenges when ad hoc groups are distributed.

Very little is known today about the differences in group dynamics of ad hoc groups. In his research study, Tuckman only reported hypotheses on short-term groups development [Tuc01, p. 79]. He supposed that "duration of group life would be expected to influence the rate and amount of development." Nevertheless, short-term groups would also be expected to "essentially follow the same course as long-term groups […] with the requirements that the performing stage be reached quickly," to the detriment of the other phases that are not "as salient as task execution" in task-oriented groups.

The study of short-term groups has been somewhat neglected by group research, especially GSS, since it was only accounted as one of the factors that could explain variance of experimental results. Nevertheless, useful insights have been gained from a

review GSS research on the effects of group history and experience, in the comparison between established and ad hoc groups. Hall & Williams were among the first to report that conflicts and decision quality in decision-making tasks are moderated by group history [Hal66]. While decision quality resulted positively related to outcome quality in established groups (i.e., the more the conflicts, the higher the decision quality), the relationship resulted reversed for ad hoc groups (i.e., the more the conflicts, the lower the decision quality). This result was later confirmed by Dennis et al., who also found that established groups did not communicate more than ad hoc groups, which in turn showed a greater equality of members' participation (i.e., no domination as for established groups' communication), but also less openly critic messages (i.e., more inhibited communication) [Den90]. Mennecke et al. found partial evidence in support of the major quantity of information shared by ad hoc groups [Men95]. Benbasat & Lim performed a meta analysis of research on the effects of group history and found that decision quality is not significantly affected by group history, which instead was confirmed to negatively affect equality of participation (i.e., the more the past experiences share by a group, the less equal the members' participation) [Ben93]. In addition, with respect to traditional established groups, ad hoc groups typically exchange more task-focused, impersonal information, and exhibit less openness and trust [Chr97]. Finally, Alge et al. suggested the need to distinguish between past and future groups for investigating the effects of groups' experience and motivation [Alg03]. *Past* groups are teams nearing to completion of a collaboration, whereas *future* groups, instead, are newly formed teams just starting a collaboration. Past and future groups exhibit different level of motivation. Members of future groups are more likely to be motivated to engage in interactions than members of past teams who feel to be close to the end of the collaboration and thus, tend to exchange a lower amount of information. However, it is unclear how these results relate to ad hoc groups. Given our proposed definition, the characteristics of past and future teams blend in the temporary nature of ad hoc groups, in the sense that the limited temporal scope makes an ad hoc group a newly formed team, also close to the completion.

The technological challenges to be faced in supporting distributed, ad hoc groups stem from the limited temporal scope too. Given the rather occasional and temporary

nature of ad hoc groups' collaboration, the adoption and maintenance costs of complex collaborative platform (groupware, see Chapter 6) can hardly be justified and sustained. The adoption of such sophisticated collaborative platforms has proved to be problematic even for established groups, in both traditional [Orl92] and virtual organizations [Ols00]. Hence, we argue that ad hoc groups, to be effectively nimble, should be supported by collaborative tools that have a low learning curve, so that dynamic engaging of new members is facilitated, and whose infrastructure and administration costs are minimal, so that dynamic creation is facilitated. This need for supporting dynamism turns out to require the adoption of either commonly available tools, such as instant messaging, email, wikis, issue trackers, or systems that do not require administration and maintenance of any central resource by design [Lan03a, Cal04a, Cal04b]. In the latter case, P2P collaborative systems can support ad hoc groups in that they build overlay networks that sit on top of the Internet, and almost exclusively use resources (e.g., disk storage, bandwidth) already available on the same hosts running the peers (i.e., the edge of the Internet, see Section 7.2.1 for more on P2P). Thus, P2P systems do not charge users with any costs, other than those coming from the bare use of peers. However, P2P is not the only available solution (see Section 7.3.1).

To conclude this chapter, we show in Figure 12 the graphical representation of the causal model, updated so as to include all the variables identified here, which characterize the task, the media and the group contextual factors. Such variables will be of help in the study discussed in Chapter 8.

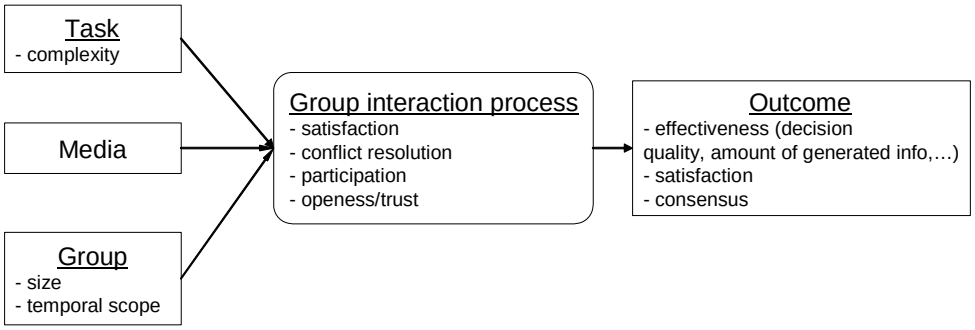


**Figure 12. Causal model updated with variables that define the contextual factors and their effects**

## 3.6. Summary

In this chapter we have briefly introduced the field of group research. We have presented several techniques to study groups, including laboratory experimentation, which has been used in the empirical study described in this dissertation (see Chapter 8). The chapter has also discussed McGrath's Task Circumplex framework, the model most widely-used in group research to categorize tasks, and objectively evaluate and compare their complexity. Finally, we have presented a particular kind of short-term, dynamic groups, namely ad hoc groups, for which we have reviewed the existing definitions given in the literature, provided a new one, and discussed the challenges and needs that technology has to face to effectively support them. Short-term collaborations represent an emerging scenario and, hence, it is extremely relevant to group research understanding how to support ad hoc teams of stakeholders, who are just a common example of such teams involved in distributed requirements engineering activities.

# Chapter 4:
# COMPUTER-MEDIATED COMMUNICATION

---

*"Txt is Gr8!"*

A. Dix, et al., *"Human-computer interaction,"* Prentice-Hall (2003)

## 4.1. Still Motivation for Research on Text-Based Communication?

As geographically dispersed individuals more and more communicate via computer, understanding the effectiveness of the very many available media has become vital. Media are usually classified in the time/space matrix (see Figure 13), according to both the spatial dimension (collocated/distributed, i.e., *where* interaction occurs) and the temporal dimension (synchronous/asynchronous, i.e., *when* the interaction occurs). For instance, F2F communication allows synchronous interaction and requires physical collocation of individuals. Instead, email allows asynchronous interaction and does not require collocation.

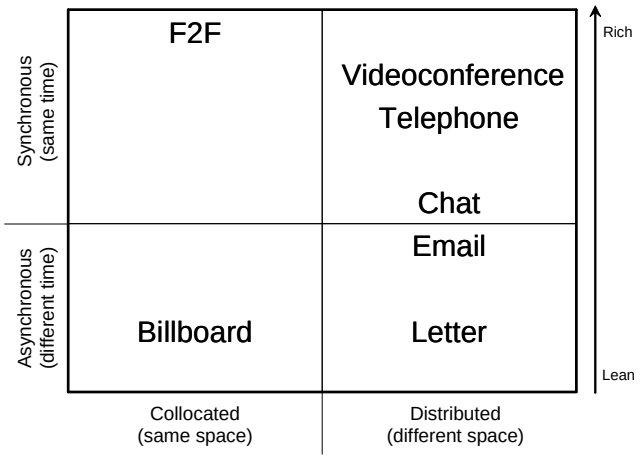


**Figure 13. Rich/lean media ranking in classic time/space matrix**

Media can also be classified according to another dimension: Richness. We can intuitively epitomize *richness* as the ability of media to convey a larger amount of information in different forms. The figure sopra shows the media along the media richness continuum. F2F is the richest form of communication, since it conveys information via audio and video channels, but also through cues like gesture and posture. Consequently, videoconference is richer than telephone, since the latter lacks video as information channel, whereas email is richer than letter, since electronic mail can also attach multimedia content. Many Computer-Mediated Communication (CMC) theories have provided different definitions of media richness, but, despite such differences, the resulting rank of media richness has never changed from the one presented above. Besides, where many CMC theories have agreed on the inadequateness of text-based communication for complex, collaborative tasks, suggesting that, as complexity increases, so should the level of richness of the media used.

Despite the negativity of the aforementioned technological and theoretical premises, the last decade has witnessed the success of many open-source projects which are coordinated through the almost-exclusive use of text-based technologies, such as web sites, email, and IM. These technologies, although not novel, have found their own way in supporting collaboration (see Chapter 6). Email is the most used collaborative tool to date, and a place where new collaborations emerge [Gey03]. IM, although initially

banned as an application intended only for teenagers, has found a number of uses in the workplace, including opportunistic interactions, and a 'signaling' function by which people negotiate their presence and availability [Han02, Her02]. Web sites and their natural evolution, the Wikis, foster collaboration throughout knowledge sharing [Cun01]. Open-source development provides just one of the scenarios where text-based communication is effectively used to perform complex collaborative tasks. Interaction of individuals is deeply influenced not only by media characteristics as well as by tasks requirements and group characteristics like history and experience (see Chapter 5 for more).

To further motivate the research on lean, synchronous communication presented in this dissertation, in the following we review the fundamental theories on CMC and media selection. Section 4.2 and Section 4.3 discuss the Social Presence Theory and the Media Richness Theory, respectively. The theory of Common Ground is introduced in Section 4.4. Section 4.5 presents the Media Synchronicity Theory. Finally, the Media Richness Paradox is discussed in Section 4.6.

## 4.2. Social Presence Theory

Social Presence refers to the degree to which one perceives the presence of participants in the communication. Social Presence theory argues that media differ in the ability to convey the psychological perception that other people are physically present, due to the different ability of media to transmit visual and verbal cues (e.g., physical distance, gaze, postures, facial expressions, voice intonation, and so on) [Sho76]. Some mediums (e.g., videoconferencing or telephone) have greater social presence than other mediums (e.g., email), and media higher in social presence are more efficient for relational communication (i.e. building and maintaining interpersonal relationships), as they involve social/personal issues and thoughts.

Social Presence presumes the outcome of an interaction to be determined by the capacity of the selected medium to support the type of communication required. More specifically, Short et al. argue that F2F interaction, thanks to the wider capacity of

conveying social presence, is more effective for relational communication than text-based media, such as emails, which do not transmit any cue and are then, more effective for task-focused communication.

Finally, Social Presence theory has also been found to be a strong indicator of satisfaction, that is, the higher the sense of social presence conveyed by a medium, the higher the satisfaction perceived by participants when communicating [Gun97].

## 4.3. Media Richness Theory

One of the most widely applied theories of media selection is Media Richness theory by Daft & Lengel [Daf84, Daf86]. Media Richness, which builds on the theory of Social Presence, argues that communication media differ in their ability to facilitate understanding. Daft & Lengel have defined information richness as the capacity of information "to change understanding within a time interval" [Dat79]. Thus, in Daft & Lengel's terms, what differentiates richer media from leaner media is the amount of information a medium could convey to change the receiver's understanding within a time interval. This capacity depends on several factors, such as the ability of the medium to transmit multiple cues, immediacy of feedback, and language variety. The perceived sense of social presence of a medium is proportional to the medium richness. As a result, rich media with a wide communication capacity also have a high level of social presence. F2F interaction is the richest media, due to its capability of expressing message context in natural language and conveying at the same time multiple cues via body language and tone of voice, and it is supposed to change understanding of participants in communication in a shorter time interval. The second richest medium is videoconferencing, because, although it still grants the use of natural language, and the access to some visual and verbal cues, it conveys a lower sense of social presence to conversation participants. Email, chat/IM, and letters are instead the leanest media because, when adopted, communication exchanged by participants is conveyed on a single channel, i.e., text, be it written or typed.

Like Social Presence theory, also Media Richness theory presumes that the outcome of an interaction is determined by the communication capacity of the selected medium.

While Social Presence theory relates performance primarily to the type of interaction required (relational vs. activity-focused), Media Richness Theory asserts, instead, that performance depends on the appropriateness of the match between media richness characteristics and information requirements of the task (clarification vs. additional information). Indeed, Media Richness theory postulates the existence of two complementary forces that act on participants when they process the information exchanged when communicating (see Figure 14). One force is *uncertainty*, which is defined as the "difference between the amount of information required to perform a task and the amount of information already possessed" [Daf86]. This definition builds on earlier research work about information theory (i.e., as information increases, uncertainty decreases [Sha49]). Uncertainty is reduced obtaining additional data and seeking answers to explicit questions. The other force is *equivocality*, which is the existence of multiple and conflicting interpretations about a situation [Daf86]. As uncertainty is more related to the amount of information available, equivocality is more related on the quality of information available: Equivocality means ambiguity and reflects confusion and lack of common understanding, whereas uncertainty means the absence of sufficient data necessary and reflects the inability to process information properly.

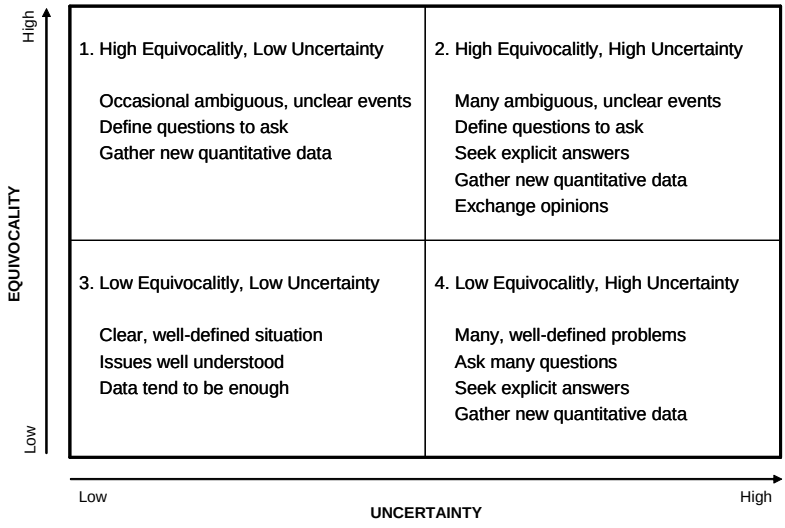


**Figure 14. The uncertainty and equivocality forces that act on individuals during communication**

Equivocality is reduced by seeking for clarification, reaching agreement, and deciding what questions to ask. The postulation of the existence of these two complementary forces has also implications on the selection of the most effective medium to use. Media Richness theory posits that rich media are better suited in equivocal communication situations (where there are multiple, even conflicting, interpretations for available information), whereas lean media are best suited in uncertain communication situations (where there is a lack of information). Equivocality is often symptomatic of disagreements and, thus, it can be reduced by providing sufficient clarifications. Rich media interaction (e.g., F2F), is preferred in situations of equivocality, as it allows for rapid feedback and multiple cues, thus facilitating the convergence to a shared interpretation. On the other hand, when messages are not equivocal, lean media are preferred. Thus, uncertainty can be reduced by obtaining sufficient additional information using media like email or written reports. Therefore, in short, Media Richness proposes that task performance will be improved when tasks needs are matched to the medium ability of conveying information.

Finally, we notice that Daft & Lengel have treated equivocality and uncertainty as independent constructs. However, it must be pointed out that a new amount of data can also generate ambiguity, and that equivocal scenarios may need more data to converge as well.

## 4.4. Common Ground Theory

The Common Ground theory by Clark & Brennan is a fundamental theory in the CMC field [Clar91]. It subsumes all the existing theories of communication in that it describes the basic process of grounding, a process orthogonal to all forms of communication, which encloses the essential goal of communicating: Reaching a common understanding. Indeed, *grounding* is the interactive process by which communicators exchange evidence in order to reach a mutual understanding, updating moment by moment their *common ground*, that is, the amount of shared information already owned.

Communicating is more than simply sending off messages. Speakers must assure themselves that the message has been correctly understood by receivers. Communication is a collective activity that requires coordinated action of all participants, and grounding is crucial for keeping track of the coordination. Individuals contribute to a conversation repeating two steps, namely *presentation*, that is, the speaker presents an utterance to the receiver(s), and *acceptance*, that is, the receiver(s) accepts(accept) the utterance, giving evidence of correctly understanding what the speaker meant. It takes both phases for a contribution to be complete: Grounding and the communication itself are impaired if the speaker does not get any evidence of acceptance. Evidence can be either positive (the message has been understood, the speaker can go on) or negative (the message is misunderstood and the speaker must repair before proceeding). Such evidence can be provided by different grounding techniques which change with medium. Grounding techniques include, to name but a few, *acknowledgements* (e.g., nodding, saying 'yes,' or typing 'ok'), *spelling* (e.g. spelling one's family name), and *verbatim displays* (e.g., repeating word by word a telephone number). But also speakers can explicitly seek for evidence asking questions (e.g., saying 'right?' at the end of an utterance). Questions asked from receivers are usually a form of negative feedback as they represent a request for clarification. However, the positivity or negativity of acknowledgements is not context-free.

Grounding techniques are deeply affected by media characteristics. Since text-based communication does not convey neither visual nor verbal cues (e.g., nodding, face expression, gaze direction are unavailable), it constraints the possible form of evidence that people can seek to acknowledgments (one would never use verbatim displays or spelling in text-based chat). Clark & Brennan go beyond the level of media richness and social presence and present eight properties that act as constraints on the grounding process (see Table 1).

**Table 1. Media constraints affecting grounding (adapted from [Ols00])**

| Medium | Copresence | Visibility | Audibility | Synchronicity | Simultaneity | Sequentiality | Reviewability | Revisability |
|---|---|---|---|---|---|---|---|---|
| F2F | ● | ● | ● | ● | ● | ● | | |
| Videoconference | | ● | ● | ● | ● | ● | | |
| Telephone | | | ● | ● | ● | ● | | |
| Chat/IM | | | | ● | ● | ● | ● | ● |
| Email | | | | | | | ● | ● |
| Letter | | | | | | | ● | ● |

Participants in a F2F conversation usually establish common ground on the fly, as they have access to cues like facial expression, gestures and voice intonation. Instead, when participants communicate over media, the fewer cues they have, the harder to construct it. As a consequence, according to Clark & Brennan's theory, "people who have little common ground benefit significantly from having a video channel" and, conversely, "only people who have previously established a lot of common ground can communicate well over impoverished media" (e.g., email or IM) [Ols00]. From the previous figure we notice that text-based communication lacks key attributes like *copresence* (owned only by F2F communication), *visibility*, and *audibility* that Common Ground theory claims to be necessary for communicators unknown to each other for developing mutual understanding. *Simultaneity* refers to the ability of the medium to allow for full-duplex communication, that is, individuals can send and receive at once and simultaneously. Simultaneity is strongly related to *synchronicity*, which distinguishes between same time and different time media. However, no medium has all the attributes at the same time. Text-based communication offers two characteristics that even F2F and audio/video communication lack, namely reviewability and revisability. *Reviewability*, also called reprocessability, is the extent to which a message can be reexamined or processed again within the context of the communication event. Text-based media enable the receiver to repeatedly process the message to ensure accurate understanding. *Revisability*, also called rehearseability or editability, is the extent to which media enables the sender to rehearse or fine tune the

message before sending. Text-based media enable the sender to carefully edit a message while it is being sent to ensure that the intended meaning is expressed exactly. Erickson and Kellogg [Eri00] have drawn attention to these two powerful characteristics of text-based communication, which make it persistent, traceable, thus enabling the use of search and visualization technologies.

When a medium lacks one of these characteristics, it forces people to use alternative grounding techniques. This happens because the costs (i.e., the effort for the speaker, the receiver or both) of using the different techniques of grounding change. Clark & Brennan count 11 different types of costs. For instance, *delay costs*, that is, the cost of waiting for messages to be completed, are paid by both speakers and receivers. Such costs have to be low in synchronous media, as long pauses would disrupt communication. *Production costs* of messages are paid only by speakers and are much lower in media carrying voice than in those text-based. In contrast, *reception costs* are only paid by receivers. Listening is generally easier than reading. However, reading may be less costly when messages content is particularly complex, to the point that they must be reviewed several times to allow for correct deliberation. Thus, grounding process is also affected by the purpose of communication (i.e., the task). This aspect, however, has not been examined in deep by Common Ground theory. When individuals communicate, they try to reach understanding minimizing the effort for themselves and the others, paying as few of these costs as possible. This rule is known as the *least collaborative effort* principle.

## 4.5. Media Synchronicity Theory

Both Social Presence and Media Richness theories presume that the outcome of an interaction is determined by the communication capacity of the selected medium. Media Richness Theory relates performance primarily to type of information required by tasks (clarification vs. additional information), whereas Social Presence theory relates it primarily to the type of interaction (relational vs. activity-focused). A number of empirical studies of media use have provided evidence that runs counter to the

predictions [Den99, Car99a], thus pushing researchers to theorize that media selection is also affected by factors beyond richness.

Social Presence and Media Richness theories have been refined by Media Synchronicity theory [Den98a, Den98b, Den99]. Social Presence and Media Richness theories are task-centric: A task is the key element to medium selection, but it is considered as a high level construct – i.e., relational or activity-focused, equivocal or uncertain. As suggested by McGrath [McG91], tasks are composed of many sub-elements, processes and activities which may need different media. For example, in Daft & Lengel's terms, resolving a task of equivocality would mean developing a shared framework for analyzing the situation, populating the framework with information of a shared meaning, and assessing the results to arrive at a shared conclusion for action. However, each of these steps may have different media needs, such that even tasks of uncertainty may include steps that require rich media [McG93a]. Media Synchronicity theory posits that group communication, regardless of the task (whether equivocal or uncertain, relational or activity-focused), is composed of two fundamental communication processes, conveyance and convergence. *Conveyance* is the exchange of information, followed by deliberation on its meaning. It can be divergent, in that not all participants need to focus on the same information at the same time, nor must they agree on its meaning. *Convergence* is the development of shared meaning for information, in that participants must understand each other's views and agree. The constructs of conveyance and convergence are not different from the concepts of uncertainty and equivocality developed by Media Richness theory. However, Daft & Lengel have treated equivocality and uncertainty as independent constructs. Therefore, for resolving equivocality Media Richness theory emphasizes the need to converge, whereas conveyance is left to tasks of uncertainty. Instead, Media Synchronicity theory argues that conveying information and converging on a shared meaning are equally critical for tasks of equivocality and uncertainty: New amounts of data can also generate ambiguity, and equivocal scenarios may need more data to converge as well. Thus, without adequate conveyance of information, individuals will reach incorrect conclusions, and without adequate convergence, the group cannot move forward.

Social Presence and Media Richness theories assume the existence of the richest medium in absolute, which is F2F communication. According to Dennis & Valacich, ranking media in absolute terms is not practical, though. They argue that media should not be ranked in order of their richness without consideration of context, and that attempting to recommend a single medium based on a high level task is doomed to failure. Media possess many capabilities, each of which may be more or less important in a given situation. Media Synchronicity theory postulates that media have a set of capabilities, and that performance will be enhanced when such capabilities are aligned with the processes of conveyance and convergence. Thus, in Dennis & Valacich's terms, "the 'richest' medium is that which best provides the set of capabilities needed by the situation," that is, the individuals, the task, and the social context. Table 2 examines the capabilities of several media.

**Table 2. Capabilities of media (adapted from [Den99]). Media are listed as having a range of capabilities because they are configurable (e.g., email may or may not enable the use of tables or graphics)**

| Medium | Symbol Variety | Parallelism | Immediacy of Feedback | Rehearseability | Reprocessability |
|---|---|---|---|---|---|
| F2F | low-high | low | high | low | low |
| Videoconference | low-high | low | medium-high | low | low |
| Telephone | low | low | medium | low | low |
| Letter | low-medium | high | low | high | high |
| Email | low-high | medium | low-medium | high | high |
| Chat | low-high | high | low-medium | medium-high | high |

*Symbol variety* is the number of ways in which information can be communicated – the 'height' of the medium – and subsumes Daft & Lengel's multiplicity of cues and language variety. The importance of symbol variety depends upon the piece of information that needs to be communicated. In general, conveyance should require a greater symbol variety depending upon the task. In contrast, convergence requires

understanding others' interpretations, which can usually be communicated using a simpler symbol set. *Parallelism* refers to the number of simultaneous conversations that can exist effectively – the 'width' of the medium. In traditional media such as the telephone, only one conversation can effectively use the medium at one time. In contrast, many electronic media can be structured to enable many simultaneous conversations to occur. The importance of parallelism depends upon the number of participants. It is unimportant for small groups. For large groups, however, parallelism is very important to conveyance in enabling all members to participate. Usually, the greater the parallelism, the easier it is to generate divergent information (i.e., conveyance). Conversely, convergence will generally benefit from low parallelism because the focus of the process is on understanding others' viewpoint. As the number of conversations increases, it becomes increasingly difficult for the group to focus on one topic or issue, which may in some circumstances impede the development of mutual understanding (i.e., convergence). *Immediacy of feedback* is the extent to which a medium enables users to give rapid feedback on the communications they receive (i.e., the ability of a medium to support rapid bidirectional communication). It is important in improving understanding because it enables mid-course corrections in message transmission, so that any misleading elements in the message as sent can be quickly corrected. More immediate feedback can have significant benefits in improving the speed and accuracy of communication. Immediacy of feedback and parallelism dimensions define 'the level of synchronicity' of media. *Rehearseability* and *reprocessability* match respectively with the attributes of revisability and reviewability defined by Clark & Brennan for the Common Ground Theory. Rehearseability is probably unimportant for simple messages, but becomes more important as the complexity or equivocality of the message increases because increased rehearseability will lead to improved understanding. However, media with high rehearseability tend to have lower feedback. *Reprocessability* enables the receiver to repeatedly process the message to ensure accurate understanding, thus fostering conveyance. Reprocessability becomes more important as the volume, complexity, or equivocality of the message increases. Increased reprocessability will lead to improved understanding, regardless of the information or communication process (conveyance or convergence), although it is

often more important to conveyance. Conveyance often produces information requiring deliberation, for which reprocessability is important.

In media selection one must take into account that most tasks require individuals to both convey information and converge on shared meanings, and media that excel at information conveyance are often not those that excel at convergence. Thus, choosing one single medium for any task may prove less effective than choosing a medium, or set of media, which the group uses at different times in performing the task, depending on the current communication process (convey or converge).

According to Media Synchronicity theory, although the selection of the most appropriate medium (or set of media) depends upon all these five dimensions, the key to effective media usage is matching the synchronicity level to the level of conveyance and convergence required to perform a task. Indeed, Dennis & Valacich posit that media that support high immediacy of feedback and low parallelism encourage the high synchronicity, which is the key to the convergence process. Conversely, media that support low immediacy of feedback and high parallelism provide the low synchronicity, which is the key to the conveyance process. Although the formulation and the constructs names change, the task-media matching suggested by Media Synchronicity theory is the same one suggested by Media Richness theory. Indeed, high-synchronicity media, with immediate feedback and low parallelism, are exactly F2F, and audio/video conference, that is, the richest media high in social presence which best fit equivocal tasks. High parallelism, instead, is not feasible when audio and video channels are available. Thus, low-synchronicity media with high parallelism are exactly email, chat, and IM, that is, the lean media low in social presence which best fit uncertain tasks.

Beside synchronicity, there are other factors that influence the effectiveness of media in supporting different groups, even those performing similar tasks. Group history – i.e., the extent to which groups have worked together in the past – is a situational factor that can influence effectiveness because it can alter the perception of media richness of time. Established groups are more likely to have established norms (e.g., roles within the group), and well established processing norms for the task performing. The group will be more likely to move directly to execution with less storming and norming. During performing, group members are able to work separately on their assigned tasks. Thus,

performing requires more conveyance than convergence, although some convergence is clearly required. The need for media synchronicity is therefore lower during performing than during forming, storming, and norming. As a group matures they "are likely to become able to carry out all their functions, at least for routine projects, with much less rich information exchanges" [Den99]. This means that (1) the communication requirements of groups will likely differ over time, depending upon shared experiences; (2) the perceptions about medium usefulness for a task and the group's ability to perform a task in a given medium change over time. As group members come to know each other better over time, they share common experiences that may be evoked by very simple messages that refer to those shared experiences. Therefore, over time established groups will require less convergence communication processes, or, equivalently, less use of high synchronicity (high feedback, low parallelism) communication environment. Conversely, newly formed groups (e.g., ad hoc groups) will have fewer well established norms and will likely spend more time in forming, storming and norming, before moving to performing. This will result in more complex processes requiring more conveyance, and, especially, convergence. Before group members can effectively work together they often need to have a better understanding of each other, and socially related communication activities that are best developed through media with social presence. Thus, newly formed groups, groups with new members, and groups without accepted norms will require more use of media with high synchronicity (high feedback and low parallelism), and symbols sets with greater social presence.

## 4.6. Media Richness Paradox

Researchers have long studied the effects of social presence and media richness on media choice, and the effects of media use. However, it is not always the sense of presence that is vital to communication, but also having sufficient information in the appropriate format and the ability to properly process it [She92]. Furthermore, the original premise of Daft & Lengel's Media Richness theory was to understand how media effect a change in receivers' understanding. Nevertheless, the influence of media choices on the cognitive processes that underlie communication has been overlooked.

Robert & Dennis described a cognitive-based view of media choice and media use, based on dual process theories of cognition, which argue that in order for individuals to systematically process messages, they must be motivated to process the message and have the ability to process it [Rob05]. Communication is not only an exchange of information, but also an exchange of attention. Different media have different usage costs to the receivers. Running counter to past research (i.e., the more complex the task, the richer the media to be used), they argued that the use of rich media high in social presence induces increased motivation, but decreases the ability to process information, whereas the use of lean media low in social presence induces decreased motivation but increases the ability to process information (see Figure 15). Robert & Dennis called the inverse relationship between motivation and attention with the ability to process "media richness paradox."

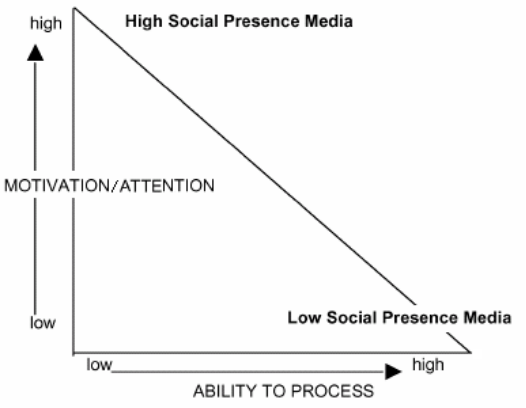


**Figure 15. Media Richness Paradox is the inverse relationship between motivation and attention with the ability to process (adapted from [Rob05])**

This paradox has profound implications on CMC research, since both Social Presence and Media Richness theories posits that face-to-face communication, as typical examples of rich/high-social-presence media, is better suited for highly equivocal tasks. One of the criticisms often moved against these two theories is that they consider the 'perceived' effectiveness of media from a sender's perspective. The cognitive-based model of Robert & Dennis reverses the perspective, analyzing from a receiver's point of view how media affect the change in understanding. In general the

greater the social presence of a medium, the greater the receiver's motivation has to be to participate in the communication process, but also the greater the sender's the ability to monitor attention. Thus, senders will require the use of rich media to ensure that receivers have high levels of attention and are motivated to process the message.

However, the level of social presence provided by media has an inverse relationship with the receiver's ability to process the message. One important media attribute is reviewability (or reprocessability), that is, the ability to allow the receiver to reprocess the information. In general, media with low social presence provide a higher level of reprocessability that allows the receiver to stop and think over important or difficult points. Also, the receiver can repeatedly access extra sources of information, and review the message until it is fully comprehended. In contrast, by social convention, media high in social presence do not allow individuals to elaborate at will, as they are supposed to respond quickly to avoid disrupting the conversation. Rich media high in social presence allow the receiver little ability to access multiple sources of information or reprocess the information. This is a major drawback because individuals have a natural constraint on the amount of information they can accept, process, and recall. Thus, when complex messages are sent over media high in social presence, reducing the amount of time one has to process ends up increasing the information load: A receiver can quickly become overwhelmed with information in a state, commonly referred to as information overload, "in which the amount of information that merits attention exceeds an individual's ability to process it" [Sch98].

Also the number of receivers may impact the relationship between attention, motivation, and ability to process. In large groups or audiences, some receivers may not actively engage in processing the messages and will assume others will do it for them. This is referred to as 'free riding.' Free riding can go unnoticed because the sender is less able to monitor the behavior. While free riding can occur in either high or low social presence media, it is likely to be worse in low social presence media because monitoring the behavior of the receivers is more difficult than monitoring that of the senders. Past research has shown that members of electronic groups are more likely to ignore information [Phi89].

As a conclusion, the use of rich media high in social presence should be used to assure attention for small amounts of information, whereas the use of lean media low in social presence causes a decreased motivation, but increases the ability to process large amounts of information during longer periods of time. Robert & Dennis argue that different media are needed for complex tasks where information overload may be generated. In such cases, the use of mixed media, or media switching, is motivated by the need to balance attention and motivation required by senders with the ability to process information of receivers. Depending on the task at hand, when senders want to get the attention of the receiver and motivate them for an immediate response, they should use a medium high in social presence. In contrast, when deep thought and deliberation are needed to process the information, the sender should use a medium low in social presence to give the receiver time to objectively elaborate on messages.

However, information overload is not the only risk when groups communicate F2F. The pressure on group members to conform to the view of the group majority has been acknowledged as the most severe dysfunctional aspect in F2F decision-making [Hil78]. The studies on group dynamics (discussed in Section 3.3) show that in group interactions there is a continuous interplay of task-oriented and relational process, as group members act certain roles while developing and maintaining some personal relationships. Thus, previous research on socio-psychological effects in CMC postulated that the reduction of socio-emotional exchange contributes to increase group efficiency in the sense that less-rich communication media allow groups to pay less attention to interpersonal aspects of the interaction, and focus more on task. Thus, groups interacting using lean media may benefit from using ‘less social’ channels because the restriction imposed on the interpersonal information exchange allows for more-equal participation and greater attention paid to the messages, not to the individuals (i.e., less influenced by high-status member and less susceptible to the pressure of social consensus) [Kie84]. For instance, the effectiveness in generative situations, like requirements elicitation, is less affected by ‘social noise’ in communication. Instead, in problem-solving situations, like requirements negotiations, where social, emotional, and relationship concerns take time and effort away from task

resolution, the use of 'depersonalized' media may enhance group efficiency by leaving a greater portion of group-work time to task-oriented interaction [Wal96].

## 4.7. Summary

In this chapter we have reviewed the most prominent, and often conflicting, theories on computer-mediated communication (CMC). The aim of this chapter was to show that there is still motivation to research on text-based communication. We have focused on studying text-based communication because multipoint audio-video communication poses significant practical barriers to deployment (e.g., expense, infrastructure, support), especially for short-term groups. Rich media theories on computer-mediated communication, namely Social Presence, Media Richness, and Common Ground, have overwhelmingly reported about the inadequateness of text-based communication, as compared to rich media, like face-to-face or video communication. Lean media, such as email and instant messaging, lack the ability of conveying nonverbal cues that contributes to the level of social presence (e.g., gaze, tone of voice, facial expressions), which in turns fosters individuals' motivation and mutual understanding. However, running counter to these predictions, Media Synchronicity theory asserts that the effectiveness of computer mediated communication depends also on contextual factors other than media richness, such as communication channel synchronicity, task typology and group temporal scope. Furthermore, Media Richness Paradox argued that the use of rich media high in social presence should be used to assure attention for small amounts of information, whereas the use of lean media low in social presence causes a decreased motivation, but increases the ability to process large amounts of information during longer periods of time. Drawing upon these theories, we argue that, by understanding the paradoxical effects of rich media high in social presence, groups may be better able to select and use the most appropriate sets of media to accomplish their goals.

# Chapter 5:

# DEVELOPMENT OF A COMPREHENSIVE FRAMEWORK FOR GROUP, TASK, AND MEDIA FACTORS

## 5.1. Managing the Context: The Intertwined Effects of Task, Media, and Group Factors

The theories discussed in the previous chapters have framed a complex theoretical background for the selection of communication media. Messages communicated to a group on channels that are inappropriate to the context may be misinterpreted by recipients or may be otherwise ineffective with regard to their intended purpose [Tre87, Tre90]. In group research, context is defined by the group, task, and media factors. In Chapters 2 to 4 we have analyzed the effects of each of these situational factors on group process and outcome. In addition, the effects of these factors depend also on their mutual interaction. Figure 16 shows the causal model updated to graphically represent the effects of these interactions. Given a specific group, its interaction process and outcome are heavily affected by the interaction occurring between task and media factors (A). For instance, task-medium mismatches may require communication participants to engage in compensating activities to clarify message content, leading to

possible communication inefficiencies [McG93a]. Likewise, given a specific task, group interaction process and outcome are heavily affected by the interaction occurring between group and media factors (B). For instance, group-medium mismatches may cause members of group unknown to each other to misinterpret message content due to the lack of shared experience, leading to possible performance inefficiencies [McG93b].

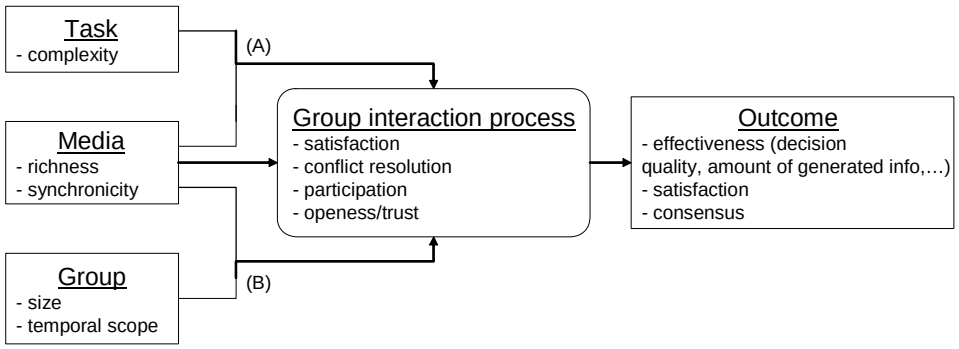


**Figure 16. The intertwined effects of media with task (A) and group (B) factors**

The remainder of this chapter is structured as follows. In Section 5.2 and Section 5.3 we respectively discuss the theories for appropriately matching media characteristics with the task and the group. Finally, in Section 5.4 we develop a comprehensive framework for the selection of communication media appropriate for the context, which consistently encompasses all the theories discussed so far.

## 5.2. Matching Task and Media Characteristics

Although often conflicting, the CMC theories reviewed agree with the need to consider task characteristics for selecting the most appropriate media. One of the most acknowledged limitation of McGrath's Task Circumplex is just its limited usefulness for determining technological support for executing groups task when group need to communicate over a medium (see Section 3.2). Thus, several frameworks have been developed to determine the best-fitting task-technology matches. In this section we review two of these frameworks, both building upon the Task Circumplex.

### 5.2.1. Time-Interaction-Performance Theory

The Time-Interaction-Performance (TIP) theory, developed by McGrath & Hollingshead, has been among the first conceptual frameworks proposed to take into account the interaction of task and technology characteristics, in the evaluation of electronically-mediated group interaction [McG94]. Time-Interaction-Performance theory builds upon Task Circumplex and Media Richness theories, and hypothesizes that communication that occurs in the four tasks categories of the circumplex can be ordered by complexity and the amount of information required. In other words, the four task categories of the Task Circumplex, ordered by complexity, can be arranged in the same order along the media richness continuum hypothesized by Media Richness Theory (i.e., showing again that the more complex the tasks, the richer the information exchange required). Figure 17 illustrates the task-media fit attempted by the theory, with respect to the communication media.

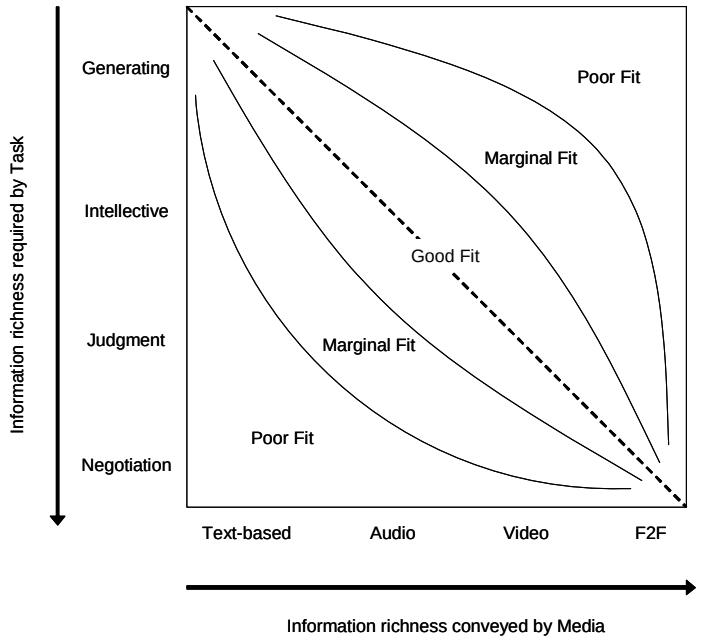


**Figure 17. The task-media fit suggested by the TIP theory (adapted from [McG94])**

The best-fitting combinations of information required by tasks and information conveyed by media lie near the main diagonal. Instead, the outer edges that are

progressively distant from the diagonal represent less well-fitting to poor-fitting matches. For instance, generating tasks (e.g., brainstorming) may require only the transmission of ideas or plans, hence "less rich" information. In contrast, tasks requiring groups to negotiate and resolve conflicts may require the transmission not only of facts, but also of affective messages or interpersonal communication, which are best conveyed by rich media. The previous figure shows that there are two types of poor-fit combinations: (1) when tasks require more information richness than selected media can deliver, groups are expected to suffer from problems of effectiveness and quality, forcing individual to exchange further compensative information; (2) when media provide more information richness than tasks require, groups are expected to suffer from problems of efficiency because media conveys not only facts, but also non-essential communication (e.g., interpersonal and affective messages), which brings distraction. In other words, the theory posits task-media fits are appropriate only when the level of information richness of a medium is adequate to the complexity of the task. Thus, although Time-Interaction-Performance theory seems to only add to Media Richness theory an objective measure of task complexity, it actually argues that rich media do not always provide the best-fitting combination regardless of the task type.

### 5.2.2. Task/Technology Fit Theory

Consistently with what hypothesized by Time, Interaction, and Performance theory, the theory of Task/Technology Fit (TTF), by Goodhue & Thompson and Zigurs & Buckland, establishes a correspondence between task requirements and technology [Goo95, Zig98]. Task/Technology Fit theory posits that, in a scenario of collaboration, the selection of an appropriate technology that provides features and support 'fitting' the task requirements, determines an increase of performance and, to some extent, of technology utilization itself (see Figure 18).

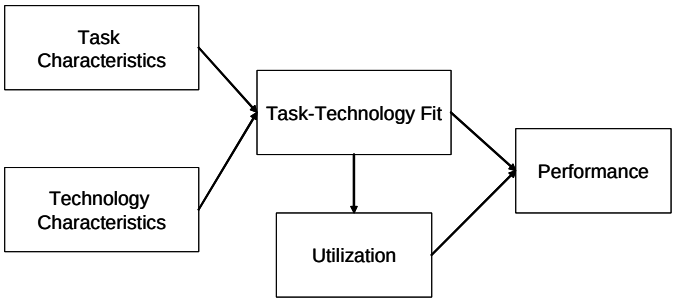


**Figure 18. Matching task and technology characteristics impacts performance and utilization (adapted from [Goo95])**

Hence, Task/Technology Fit theory states that effectiveness of computer mediated communication varies on the type of task. For instance, tasks of idea generation that involve divergent thinking and limited member interdependence (e.g., in Task Circumplex, Type 1: planning, and Type 2: brainstorming), do not require information-rich media. On the other hand, more intellective tasks (e.g., Type 3/4: problem solving, and Type 5/6: conflict resolution) involve a two-stage process: First, divergent thinking to identify all possible solutions, and secondly, convergent thinking to identify best suited solutions among those identified in the first step. Thus, convergent thinking involves a higher degree of member interdependence and requires information-rich media.

As a result, in the scenario of distributed requirements engineering, Task/Technology Fit theory predicts to be appropriate to the following task-technology fits: elicitation-lean media, negotiations-rich media. It is also interesting to note that these predictions are consistent with the predictions of the theory of Time-Interaction-Performance, if applied to the same context of distributed requirements engineering.

## 5.3. Matching Group and Media Characteristics

This section discusses the effects of temporal scope in matching group and media characteristics. The theory of Channel Expansion by Carlson & Zmud [Car94, Car95, Car99a] posits that gaining experience with channel use and communication co-

participants[9] increases the perceived richness of that channel and the ability of individuals to communicate more effectively over it. As communication participants acquire these experiences, they enhance their ability to encode/decode "richer messages," for instance, referring to shared experiences or using shared jargon [Car99a]. However, Carlson & Zmud also found that, over time, the influence of these experiences tends to diminish and eventually stabilizes.

Channel Expansion theory does not address the channel selection process. Instead, it is concerned only with the increasing "perceived" richness of a given channel and the ability to communicate more effectively over it with time. Nevertheless, the theory can be used as predictive of the effects of temporal scope in matching group and media characteristics. Channel experience is gained through use and thus, related to the length of time a channel has been utilized. Likewise, experience with group members is developed through interaction and thus, related to the group history, or the extent to which a group has worked together in the past. Hence, established groups with a shared history of previous collaboration, are expected to be able to communicate effectively also over impoverished media, like email. Conversely, ad hoc groups are newly formed and thus, do not have any shared experience that can help compensate for the leanness of the medium in use. Consequently, ad hoc groups are expected to benefit from the use of rich medium more than established groups. These results are consistent with the theory of Common Ground (see Section 4.4). Group with shared experiences have already established a certain amount of common ground and thus, can communicate well even over impoverished media.

## 5.4. Development of a Comprehensive Theoretical Framework

The theoretical frameworks reviewed on media effects, tasks, and group processes have depicted a complex research area. The complexity is reflected by the equivocality of the existing body of knowledge from previous studies conducted to evaluate the

[9] Actually, the theory identifies two other forms of relevant experience, namely experience with the messaging topic and the organizational context, for which Carlson & Zmud only found partial support. Besides, these forms of experience are not of interest in this dissertation.

(in)effectiveness of computer-mediated group interaction as compared to F2F. The consistent combination of all these group-, task-, and media-related theories resulted in a fully comprehensive framework, which encompasses all the forces, generated from situational factors, which act on the selection process of the most appropriate media for the context. Figure 19 illustrates a graphical representation of the framework, which will serve as a reference in the discussion of the findings from the experiment described in Chapter 8.

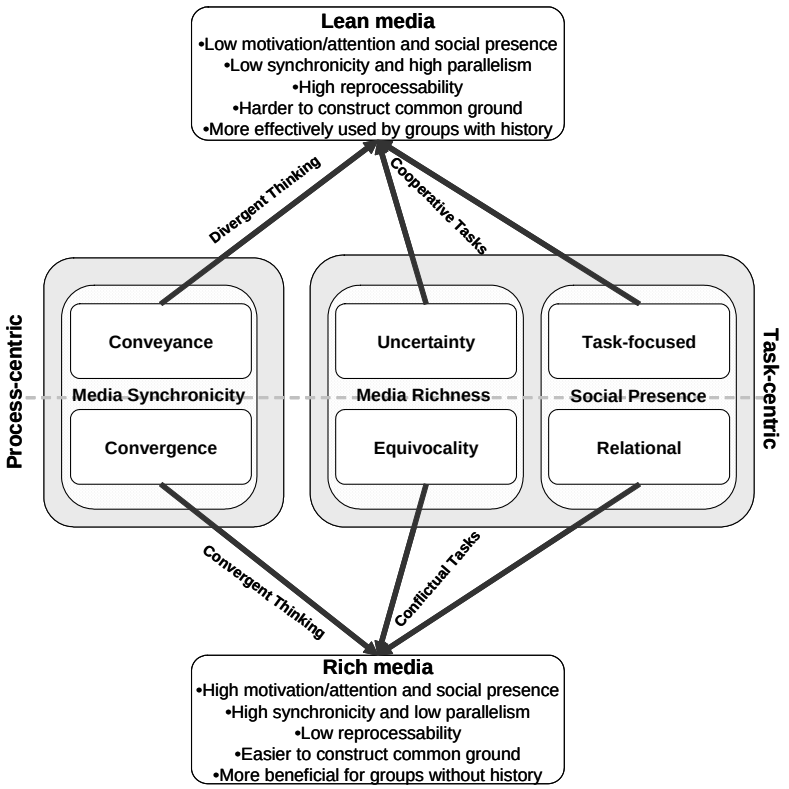


**Figure 19. The comprehensive framework for task/technology fit resulting from the consistent combination of group-, task-, and media-related theories**

The figure sopra shows the inversely-proportional, main characteristics of rich and lean media. Rich media (e.g., audio and video channels, F2F) are highly synchronous and low parallel, convey a high sense of social co-presence of individuals, ensure a higher level of attention and motivation, facilitate mutual understanding and thus, are more beneficial, especially for groups with no history, whose members are unknown to

each other. One risk with rich media is the information overload, due to the multiple channels available at one and the low reprocessability of the information conveyed over them. Conversely, lean media (e.g., email, text chat, IM) are lowly synchronous but highly parallel, convey a low sense of social copresence, motivation and attention. Lean media are more effectively used by groups of individuals who share a history of previous collaborations. One advantage of lean media over rich media is the possibility to reprocess the information exchanged, which is volatile.

The CMC theories reviewed have been divided into task-centric and process-centric theories groups. Task-centric theories (i.e., Social Presence and Media Richness) consider communication as a task to be accomplished by individuals, whereas process-centric theories (i.e., Media Synchronicity) regard it as a process to be performed by individuals. All these theories, however, define communication through task or process dichotomies. The arrows represent the 'driving forces' that act on the selection process, pushing for the selection of appropriate fits between tasks and synchronous media properties. These forces are not only useful for predicting and evaluating the goodness of task/technology fits (i.e., poor, marginal, and good fits). In fact, here we also use the framework to ultimately compare the fits between synchronous text-based communication and distributed requirements workshops.

According to Task Circumplex classification, negotiating software requirements is a complex, intellective task that involves different sub-activities, both conceptual and behavioral, where conflicts have to be resolved to converge readily to one solution among the many identified, thus reaching consensus in a timely manner and enhancing the decision-making process quality. From the point of view of the task-centric theories, a requirements negotiation is a conflictual task characterized by high equivocality and member interdependence, which requires not only task-focused messages, but also social information to be exchanged. From the perspective of communication as a process, resolving ambiguities means that opposing individual views must converge into a single shared view. All these forces consistently drive to the selection of rich media for conducting effective requirements negotiation workshops and, consequently, also show that synchronous text-based communication and requirements negotiation represent a poor task/technology fit.

According to Task Circumplex classification, elicitation is a creativity task, where new ideas or different solutions to a given problem have to be generated. Idea generation requires a low degree of member interdependence because it involves only divergent thinking. Thus, from the perspective of task-centric theories, elicitation is a cooperative, task-focused activity with limited degree member interdependence and consequently, a little need of communicating social information, which may make participant more susceptible to pressure of social consensus and domination, and take time away from task-oriented interaction. The uncertainty existing in a generative task can only be reduced by conveying additional information. Hence, from the perspective of process-centric theories, the conveyance of information is better supported by lean media, high in parallelism (or low in synchronicity), which foster idea generation by allowing multiple individuals to contribute information at the same time. Thus, all these forces consistently drive to the selection of lean media for conducting effective requirements elicitation workshops. Nevertheless, in the evaluation of this task-technology fit, we must also take into account the existing counter forces, since the use of lean mean has a detrimental effect on the level of satisfaction and motivation/attention perceived by participants. In addition, compared to established groups, members of ad hoc groups are expected to communicate less effectively over impoverished media, since they cannot use any shared experiences to compensate for the media leanness. Thus, overall, synchronous text-based communication requirements elicitations represent a marginal task/technology fit.

As a conclusion, the framework has predicted synchronous text-based requirements elicitation to be a marginal task-technology fit and, consequently, more appropriate than synchronous text-based requirements negotiations, which has been predicted to be a poor fit.

## 5.5. Summary

In this chapter we have merged the contributions from Chapter 2 (in particular, the challenges of distributed requirements elicitation and negotiation), Chapter 3 (in particular, the Task Circumplex and ad hoc group theories), and Chapter 4 (in

particular, the critical review of CMC theories). We have thus built two models, which will be used in the description and discussion of the empirical study reported in this thesis (see Chapter 8). The first model, adapted from Nunamaker et al. [Nun91], will serve as a reference model to define the context of the empirical study. The second model, instead, is an original contribution of this research effort. It consistently combines the most prominent theories on CMC and the Task Circumplex to graphically represent a theoretical framework for predicting and comparing the goodness of task-technology fits. Hence, this model will be used for the discussion of the findings from the empirical study.

# Chapter 6:

# TOOL SUPPORT FOR DISTRIBUTED TEAMS

---

*"What I got ain't what I need*
*And that's my inspiration."*
*Manmade God, "Bad Creation," Manmade God (2003)*

## 6.1. Computer-Supported Cooperative Work and Groupware

All distributed teams face substantially more and challenging difficulties as compared to collocated counterparts. Geographical distribution affects collaboration between team members in terms of reduced quality and quantity of communication and coordination. The research in the GSD has looked at and applied the same development processes adopted by open-source projects to partially overcome these problems. However, while these issues can be somewhat reduced faced adopting interaction processes that minimize interdependencies, a more direct solution is facilitating collaborative work by developing software systems, or *groupware*, that support direct communication and coordination between distributed team members. This research field is known as *Computer-Supported Cooperative Work* (CSCW).

Although the field of collaborative work still lacks a standardized terminology, the terms CSCW and groupware are prevalent. Both terms emerged in the '80s. CSCW was coined in 1984 by Greif & Cashman as the slogan for a workshop with various researchers from diverse fields, but all with an interest in studying group activities

[Gre88]. Groupware instead was first used by Johnson-Lenz [Joh82] and then adopted by the CSCW community [Gru94b, Bor00].

Although frequently used as synonyms, CSCW and groupware have a clear and distinct definition. The term CSCW is used to refer to the theoretical foundations and methodologies for teamwork and its computer support. Wilson [Wil91] defined CSCW as follows:

> *"CSCW is a generic term which combines the understanding of the way people work in groups with the enabling technologies of computer networking, and associated hardware, software, services, and techniques."*

In contrast, groupware refers to software systems supporting cooperative[10] work and integrating theoretical foundations achieved by CSCW research. Johansen [Joh88] defined groupware as follows:

> *"Groupware is a generic term for specialized computer aids [e.g., software, hardware, services, and/or group process support] that are designed for the use of collaborative work groups."*

Groupware can be viewed as the class of applications for small and organization-wide groups arising from merging computer-based and communication technology. The term groupware is nowadays almost disused in favor of preferred wordings such as 'collaborative software' and 'social software,' which also include systems used outside the workplace (e.g., blogs, wikis, instant messaging).

The remainder of this chapter is structured as follows. Section 6.2 illustrates history and focus on both CSCW and groupware technologies. In Section 6.3 email is presented

[10] A lot of confusion in the field of CSCW raised from the different interpretations of the terms collaboration and cooperation. While many authors simply considered both terms as synonyms, others drew a distinction between them. According to Dillenbourg et al. cooperation and collaboration differ in the way in which task is divided: "In cooperation the task is split (hierarchically) into independent subtasks, whereas in collaboration cognitive processes may be (heterarchically) divided into intertwined layers. In cooperation, coordination is only required when assembling partial results, while collaboration is a coordinated, synchronous activity that is the result of a continued attempt to construct and maintain a shared conception of a problem" [Dil95]. Nevertheless, in this study we will consider the terms as synonyms.

along with the reasons of its widespread adoption, insomuch that it has become the central point of work for many users. In Section 6.4 presence awareness is discussed as a means to reduce the 'social' losses due to the distance in geographically distributed collaborations. Finally, Section 6.5 presents an extension of Mozilla Thunderbird, which integrates the email client with presence awareness and instant messaging for providing both a means to ascertain and signal interruptibility, and a real-time communication channel distant to get in touch with collaborators in a more timely fashion.

## 6.2. History and Focus of CSCW and Groupware

The field of CSCW deals with cooperation within groups and tries to develop innovative computer technologies that support it. CSCW is not just an information management problem, but rather an interdisciplinary application domain in which methodologies of computer science, organizational theory, and sociology converge. Indeed, CSCW started as an effort by technologists to learn from economists, social psychologists, organizational theorists and anyone else interested in studying group activity [Gru94b]. According to Grudin, CSCW resulted from the convergence between previous trends of development: Single user application, influenced by research on Human-Computer Interaction (HCI), and organizational software, influenced by the research on Management Information Systems (MIS). More recently, a fourth area of development (Social Information Systems, SIS) has evolved from the research interests on social software for communities (see Figure 20) [Gru94a, Gru94b].

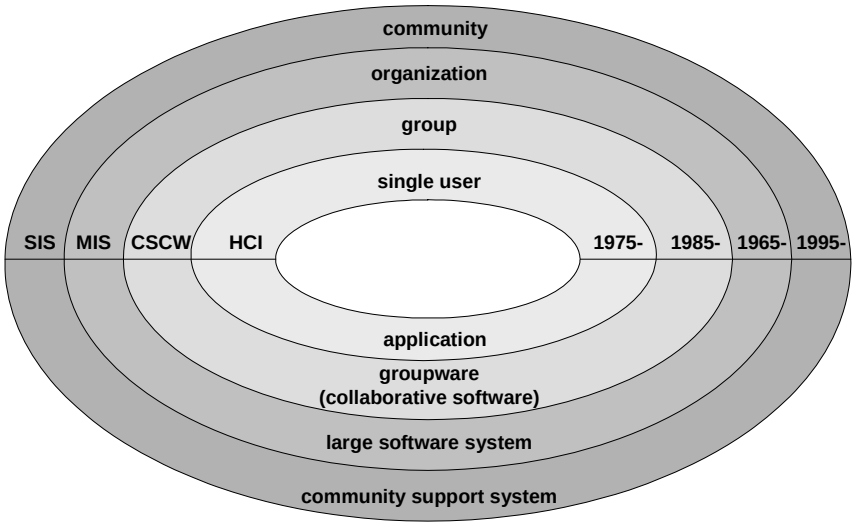


**Figure 20. CSCW in context of other research and development areas (adapted from [Gru94a])**

With regards to CSCW, work support involves four distinct, but interdependent, components shown in the Leavitt rhombus (see Figure 21). The distinct components are *task*, *organization*, *people*, and *technology*. Groupware was defined by Ellis et al. as "computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment" [Ell91]. Thus, groupware focuses on the technology component as an interface to task performing.

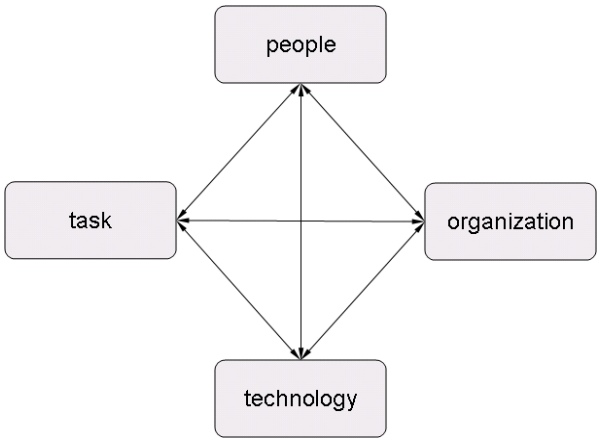


**Figure 21. Four perspectives in the development of groupware (adapted from [Bor00])**

A variety of groupware systems have been developed over the last two decades, and several taxonomies have been proposed for groupware categorization. The most widely used taxonomy is the *time/space matrix* (see Figure 22), proposed by Ellis et al. [Ell91]. This taxonomy classifies groupware by *when* and *where* participants are performing the cooperative work.

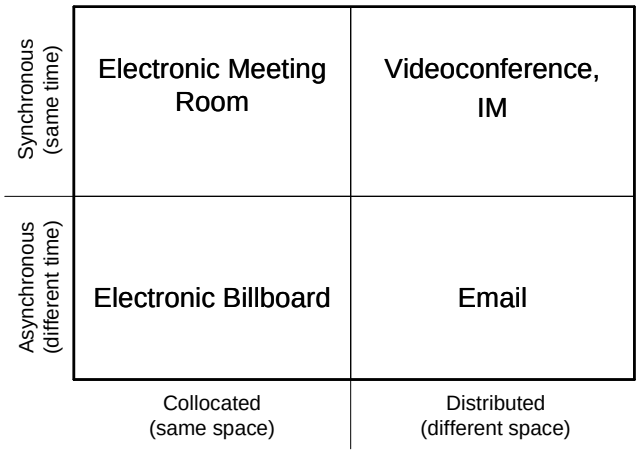


**Figure 22. Time/space matrix for groupware classification (adapted from [Ell91]).**

The time/space is useful shorthand to refer to the particular circumstances a groupware system aims to address. In literature, the axes are given different names. The space dimension, also called geographical dimension, is divided into *collocated* (same space) and *distributed* (different space). The time axis is typically divided into *synchronous* (same time) and *asynchronous* (different time). Groupware systems have been conceived to augment interaction of collocated groups who interact at the same time (i.e., face-to-face), as in the case of electronic meeting rooms, or at different time, as in the case of electronic billboards. However, groups need even more support when they are distributed and group members have to interact either synchronously, as in the case of videoconferencing or chat/instant messaging, or asynchronously, as in the case of email.

Groupware is said to be 'C-oriented' in that it supports group collaboration through three basic phenomena, namely communication, cooperation, and coordination.

*Communication* focuses on the mutual understanding of collaborators supporting information exchange. *Cooperation* between team members is needed to accomplish activities that require a joint effort, whereas *coordination* aims at finding the best way to arrange task-focused activities. The *3C model* taxonomy categorizes groupware systems with respect to the degree of the support given to the three basic phenomena. Thus, message systems, like emails and conferencing systems, are more oriented towards supporting communication, whether synchronous or asynchronous. In contrast, electronic meeting rooms focus more on cooperation, mostly needed in decision-making tasks, whereas workflow management systems focus on avoiding problems that arise from the lack of coordination between activities, mostly asynchronous.

Communication among team members is a fundamental aspect in cooperation. The concept of communication is strongly related to those of coordination and cooperation. The splitting of a cooperative task into independent (although cognitively intertwined) subtasks leads to a need for coordination and "the greater the need for coordination and cooperation, the greater the necessity for communication" [Ras91]. Hence, it is not surprising that most successful groupware systems are computer-mediated communication tools like email, or more recently, instant messaging (IM) and Internet-based telephony (VoIP).

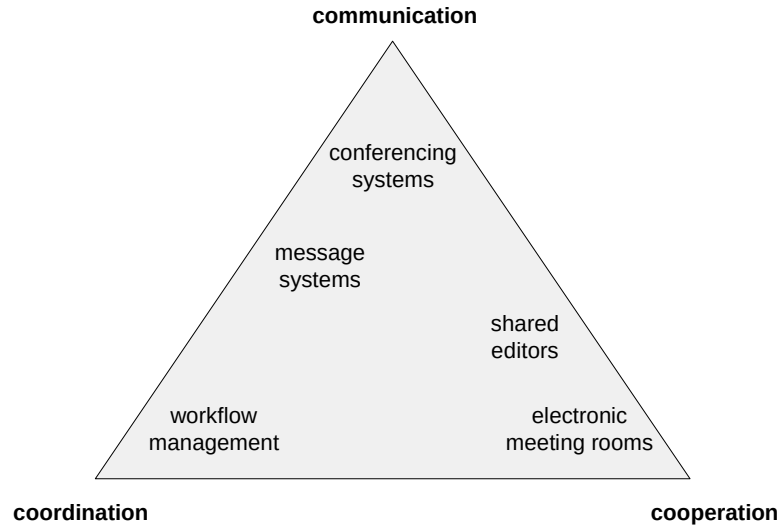


**Figure 23. 3C model for groupware classification (adapted from [Bor00])**

## 6.3. Email, the Only Successful Collaborative Tool?

Email is the most successful computer application yet invented and the form of computer-mediated communication in widest use today. It is used by millions of people to carry out their business each day. Already in 2001 nearly 12 billion email messages were sent every day and there are now as many as 170M corporate electronic mailboxes in use, growing 32% per year.[11] Over the last 10 years, it has changed the way that people work, and the ways that organizations operate. Many types of collaborative work would be unthinkable without it [Spr91]. Email has become ubiquitous even at home, and it is often the reason for purchasing a home computer [Kra00].

Kraut once said that "the only successful [collaborative] application was email" [Ens90]. In addition, several recent studies indicated that email is more and more often the place where collaboration emerges [Whi96, Duc01, Gey03]. In an early analysis of the success of collaborative applications, Grudin suggested that developers should have looked at email for inspiration on how to develop better tools [Gru94a]: First, email creates benefits for all members. Indeed, email provides an equitable advantage for both the message sender and the receiver. However, the primary beneficiary, that is, the person who sends the message, has a little more overhead due to the message composing. This situation is similar when using another successful communication media, the telephone: The caller has to pick up the phone and dial the number, whereas the callee only has to answer. Secondly, email is malleable. Studies of email usage have repeatedly documented a remarkable number of different purposes to which it is put. Thanks to its malleability (i.e., flexibility, lightweight, and ease of use), email can support conversations, but also operate as a task/contact manager [Mac98, Whi02, Bel03].

Yet, some of the same success factors contribute to the problems that are now endemic in email [Whi05]. Users do not only complain of the growing number of irrelevant, unsolicited emails (spam messages), but also about getting too many emails to keep track on (the *email overload* or *co-opting* phenomenon). Email overload was

[11] Source: http://www.jupiterresearch.com/

documented first by Whittaker & Sidner [Whi96] and later taken up by Ducheneaut & Bellotti [Duc01], and Kerr & Wilcox [Ker04] when they investigated how users used their email clients. Results showed that although email was originally intended as an asynchronous communication tool only, it is now being used for tasks other than conversation, including meeting scheduling, task management, document sharing, record keeping and file transfer. Another problem with the extensive use of email arises from the erroneously expected responsiveness. Email is generally used as an asynchronous communication medium because there is little expectation that users will read and respond to email promptly when received. However, when using email in the decision-making process, commonly problems arise because people tend to "expect everyone to be as responsive – or at least to overlook the possibility that they aren't – and to assign deadlines and decision points accordingly" [Coa04]. Furthermore, long and heated email discussions result stressful because, due to the serial nature of the medium, participants cannot interrupt each other until the whole message is dispatched. In such cases, asynchronicity fosters misunderstandings and *flamage* (i.e., harsh and emotional email outburst), which are usually more damaging than extending the deadline in the first place would have been.

When the email standard was defined, nobody thought that email would have been "abused" to the point of becoming the central point of work for many users. Yet, despite its success and problems, one paradox is how little email changed through the years. Early text based systems were replaced by graphical user interfaces, but aside from a few extensions (e.g., support for attachments, HTML, folders and address books), today's systems are remarkably similar to those introduced more than 15 years ago. The goal of our research effort with email, however, was not to address this paradox, but rather to extend the email client to begin to tackle a part of email problems. Solutions may vary depending on how one views email. When seen as a document archive, researchers seek to apply search or digital library frameworks; when seen as a task/contact manager, workflow analyses might be applied. Instead, we regarded email as a communication and collaboration tool.

### 6.3.1. Resolving Points of Friction in Collaborative Tools

Sometimes people get an email message and realize that they need more information, or need to get in touch with a collaborator right away [Whi05]. When that need arises, picking up phone, or opening an IM client to contact that person may cause a repetitive context switching. Context switching causes multiple applications workflow to interleave and provides an example of what Booch & Brown called "points of friction," because energy is wasted in the execution of repetitive, non-creative activities (switching back and forth between different applications), which can be eliminated from existing tools by integrating "creature comforts" [Boo03].

Despite what Kraut said, email is for sure the most successful collaboration tool, but it is by no means the only one. For instance, in the field of software development, Concurrent Versioning System (CVS) and its more recent descendants, like SVN and BitKeeper, have become indispensable tools for coordinating the interaction of distributed developers. Nowadays, source code management tools are always integrated within the most powerful IDEs, such as Eclipse, Netbeans and Visual Studio. Indeed, according to Booch & Brown, the integration of collaborative technologies within the development environment reduces the inherent friction of running and using different tools, thus increasing the developers' comfort and productivity, as everything stays in the workflow of a single application.

Although originally intended for sophisticated collaborative development environments, Booch & Brown's idea of reducing points of friction and context switching by the seamless integration of creature comforts contained a principle of general validity that can be also applied to single collaborative tools, such as the email client.

## 6.4. Reducing Distance with Presence Awareness

Previous research has shown that multi-site work presents considerable loss of opportunities for rich interaction and a very substantial reduction in frequency of communication [Her00]. When collaborators are geographically distributed, presence

awareness, that is the awareness of what distant colleagues are doing, their availability for interaction, and how they prefer to be reached, plays a key role for coping with the lack of physical proximity and improving multi-site work. Presence technology is believed to be important in collaborative contexts insomuch that it is has been covered by as US software patent assigned to Lucent Technologies Inc. [God00].

On the heels of a still growing popularity, synchronous IM systems have been increasingly brought into the workplace. Previous research [Han02, Her02] has shown that this form of near real-time communication has a number of uses in the workplace, including opportunistic interactions, broadcasting of information or questions, and a 'signaling' function in which people negotiate availability for interaction. Since synchronous messaging provides this lightweight way of ascertaining and negotiating availability, presence awareness has become synonymous with IM, or has been considered merely as its underlying technology. Presence awareness has made IM different from traditional form of communication in which users send out messages or calls without knowing whether the recipient is present to receive them. Yet, presence can be used not only in combination with IM, but also with other communication media, such as telephone, to make communication more timely and efficient: By ascertaining availability before placing a phone call one can substantially lower the cost of initiating a synchronous communication with someone, at a remote site, who is not available (e.g., avoid phone tag issues) [Vau03]. After all, as Turek noted, presence has always been part of how we work and the telephone's 'busy signal' is an unsophisticated presence indicator [Tur04]. IM has only been the first application to take advantage of presence awareness. Presence has gone well beyond IM and has become a killer app itself, which many tools will leverage on in the future. Today, it already spans a number of different applications, ranging from VoIP clients (e.g., Skype) to office suites (e.g., Microsoft Office). By bringing presence information in the context of the application workflow they are already in, users do not have to interrupt their work, leave the program they are using, and go to a separate communication application.

As an example of presence integration into existing applications, we developed a presence awareness plugin[12] for the wikis of our laboratory (Collab)[13] and research group (CDG)[14], in order to coordinate the online authoring of pages (see Figure 24). The plugin shows in the left sidebar who is online and what action (e.g., browsing, editing, adding an attachment) is performing on what page. The plugin uses AJAX technology to avoid frequent page refreshes. As of this writing, we are also extending the plugin to enable users to exchange instant messages from within the wiki.


**Figure 24. Collab Wiki augmented with the presence awareness plugin**

In the last years, a considerable research work has been targeted at providing cues about the presence and activity of users [Ben94]. There have been a number of approaches to awareness and availability over distance [Kra90a, Whi94], including various applications of video [Abe90, Dou92, Fis92, Fin97, Oba98], open audio

[12] http://www.pmwiki.org/wiki/Cookbook/PresenceAwareness
[13] http://collab.di.uniba.it/index.php
[14] http://cdg.di.uniba.it/index.php

channels [Hud02], and text [Chu99]. Collaborative platforms have also been used, but they proved effective only for intra-company collaboration, since they usually lose effectiveness as soon as collaboration goes beyond organization borders [Orl92]. Handel et al. and Herbsleb et al. found out from interviews conducted with members of distributed teams that they did not want an immersive virtual environment, but rather, a simple to use and unobtrusive application for signaling presence. Thus, they described the appropriate requirements for a presence system, using the metaphor of a car rear-view mirror [Han02, Her02]. In fact, a rear-view mirror provides a good example of presence awareness applied to the context of driving a car: It is a small, unobtrusive device, which effectively communicates at a glace who is around (e.g., other vehicles, road obstacles) and what they are doing (e.g., passing, turning). Drivers can use the rear-view mirror to get presence information on several levels: Quick, almost subconscious glances as part of the activity (driving), or a more intense monitoring of the others' status (e.g., making sure that someone is still following.)

While originally companies used their own proprietary protocols to provide presence capabilities in, for example, IM systems, now they are almost exclusively using two open Internet Engineering Task Force (IETF)[15] standards, namely the Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions (SIMPLE) and the Extensible Messaging and Presence Protocol (XMPP) [Vau03].

*SIMPLE*[16] is an IP-based protocol that sits on top of SIP, which is a signaling protocol for initiating, modifying, or terminating interactive user sessions between applications. Because it is IP-based, SIMPLE proponents argue it is better suited to work also with network-enabled applications such as videoconferencing and VoIP telephony. As of 2006, SIMPLE is still an IETF work in progress: Some parts have been standardized, whereas others, in particular IM sessions, are still under discussion. However, several implementations are already available, notably including the Microsoft Live Messenger (formerly MSN Messenger).[17]

[15] http://www.ietf.org
[16] SIMPLE RFC 3428 – http://www.ietf.org/rfc/rfc3428.txt
[17] http://get.live.com/messenger/features

*XMPP*,[18] which unlike SIMPLE is already an IETF standard, is a protocol for real-time messaging, presence, and structured data exchange. It is sponsored by the Jabber Software Foundation[19] and thus, it has been used for several years to build the Jabber IM network, an open and free alternative to closed IM and presence services like AIM, ICQ, MSN, and Yahoo.

The usual distinction made between XMPP and Jabber is that the former is used to refer to the protocol itself, whereas the latter refers to the application of the protocol for building the IM system. An accurate discussion of the XMPP protocol goes beyond the scope of this chapter and is given in Section 7.3.1. Unlike SIMPLE, which is IP-based, XMPP is purely XML-based, and thus its proponents argue it is better suited for extensibility. However, the two standards are not compatible with each other, which means that MSN and Jabber networks are not natively interoperable. In fact, "the presence-protocol standards conflicts have become religious wars" [Vau03]. Thus, the IETF has formulated the Common Presence and Instant Messaging (CPIM),[20] a framework that would let applications based on SIMPLE, XMPP, or any other future presence protocols, support cross-protocol messaging through gateways. XMPP RFCs already include the XMPP mapping to the IETF's CPIM specifications and thus, XMPP servers are already capable of bridging cross-protocol messaging.

With a still growing number of employees working away from headquarters, the widespread adoption of presence in the workplace is not a question of 'if,' but only of 'when.' Although interoperability issues exists among the existing standard presence protocols (i.e., SIMPLE and XMPP), the barriers are just partly related to technology. Knowing everyone's whereabouts and availability within your team or company is valuable, but it also raises severe privacy concerns. Also, management still fears it to be a productivity drain, rather than a boost. "Culture will have to change if the full benefits of presence and real-time collaboration are to be realized" [Tur04].

[18] XMPP RFC 3920-3923 – http://www.xmpp.org/rfcs/
[19] http://www.jabber.org
[20] CPIM RFC 3862 – http://www.ietf.org/rfc/rfc3922.txt

### 6.4.1. Adding Social Translucence to Collaborative Tools

Already in 1991, in a field studies analysis of electronic communication in large organizations, Perin noted the "invisibility of electronic social fields" as one possible cause of managerial suspicion and negativism [Per91].

As social creatures, we are immersed in a sea of social information. Thus, every day we make decisions based on the activities of those around us. However, software systems are generally opaque to social information. Thus, "in the digital world we are socially blind" [Eri00]. Erickson & Kellogg provided a real-world example of a "socially translucent" system. A door that can be opened in either direction is likely to slam into anyone if opened quickly. If a glass window is put in the door, as people approach they can see whether anyone is on the other side and, if so, adjust their actions appropriately. This solution works effectively for three reasons. Firstly, the glass window makes visible socially significant information which we, as humans, notice, interpret, and react to accordingly. Secondly, the glass window supports awareness. One does not open the door quickly because one knows that someone is on the other side. Our social rules come into play to govern our actions, as we have been raised not to slam doors into other people. Finally, awareness leads to the third, subtler reason, that is, accountability for actions: People who do not care about hurting others, will be held accountable for slamming the door on purpose, as a consequence of that awareness.

According to Erickson & Kellogg these three dimensions – visibility, awareness, and accountability – should be the building blocks of effective collaborative tools. Thus, in general, they defined a system as "socially translucent" when it enhances communications by making social information *visible*, and participants both *aware* of what is happening, and *accountable* for their actions. Collaborative tools should be socially *translucent* rather than *transparent* in the sense that there must be a vital tension between privacy and visibility (as the glass on the door only showed the information strictly necessary for the purpose).

In the last years, a considerable research work has been targeted at attempting to portray social cues into collaborative tools to reduce their social opaqueness by making the activities of users visible to others. In the review of these researches, Erickson & Kellogg identified two design approaches to representing social cues, namely the realist and the abstract.[21] The *realist* approach is the most straightforward since it tries to project social information directly from the physical world into or through the digital world, thus minimizing the difficulty of producing and interpreting social cues. The realist approach is typically adopted by videoconferencing systems. However, video conferencing systems have faced substantial pragmatic hurdles, such as their cost and the required infrastructure setup. Furthermore, such systems are deeply affected by the quality of sound and size of image, which may cause cues like gaze, facial expression, and intonation to go unnoticed [Ols00]. The *abstract* approach, instead, involves portraying social information in ways that are not closely tied to their physical analogues, such as the use of text and simple graphic representations. The abstract approach is typically adopted by IM systems (e.g., smileys). However, as compared to the realist approach, text and graphic have a limited potential at conveying social content [Sho76].

Although, the realist approach can be considered the most promising in the near future, we believe that the abstract approach still needs attention. Indeed, the abstract approach scales well thanks to the compactness of text and graphic representations (consider how well and how simply smileys portray social information). Furthermore, text and graphics have many useful characteristics: They are easy to produce, manipulate, and, above all, they persist over time. In particular, persistence renders text and graphics accessible at later times: "Persistent conversations may be searched, browsed, replayed, annotated, visualized, restructured, and recontextualized, with what are likely to be profound impacts on personal, social, and institutional practices" [Eri00].

[21] Actually, in their work Erickson & Kellogg discuss a third design approach, the mimetic, which tries to reproduce in digital the social cues from the real world as literally as possible (e.g., using avatars in virtual reality systems). The analysis of this approach was not of interest for this research though.

## 6.5. JabberPresence, a Mozilla Thunderbird Extension

Dourish & Bly defined awareness as "an understanding of the activities of others, which provides a context for your own activity" [Dou92]. Presence covers a wide range of information, including the simple physical presence, and the level of concentration or interruptibility of an individual. For collocated team members, this sort of information is usually gleaned through indirect communication channels, which can be signals such as walking through the halls and seeing closed doors, lights on or off, so it is much easier and takes much less time to contact a colleague. For remote teams instead, it is more difficult and time consuming to get hold of someone because there are less signals that inform whether colleagues are available, or if they are momentarily away from their desks, or away for vacation [Boy98].

Email is a socially-opaque collaborative tool: It has a very limited potential for signaling awareness and availability for interaction. Nevertheless, email still remains the preferred communication tool for collaborating. Due to its social opaqueness, however, people tend to use this asynchronous communication medium in a synchronous fashion, sending messages that need an immediate reply. Instead of using email when a prompt response is needed, one should rather use IM, first to ascertain availability and interruptibility, then to send an instant message. Furthermore, communication plays a critical role in project coordination, and facilitating less formal communication across sites can be expected to significantly reduce the problems of multi-site work [Gri99, Her99a, Her99b]. However, the continuous context switching between email and IM clients would cause "friction." Thus, taking a similar approach to that proposed by Booch & Brown for collaborative development environments, we seamlessly integrated IM and presence awareness capabilities with an email client, thus reducing the friction of also running an IM system for signaling one's own presence and availability, and ascertain the others'.

In particular, we developed JabberPresence, an extensions for the Mozilla Thunderbird mail client, which uses the Jabber/XMPP protocol for implementing the messaging and presence services [Cal06a]. JabberPresence has been designed to bridge

the gap between the asynchronicity of email, and the synchronicity of IM and chat. Running such a presence-enabled email client, there is no need for users to continuously switch between two clients and see what the others are doing, whether they are available, so as to choose the best and least obtrusive way to get in touch with them (i.e., sending either email or IM). With this extension, not only the friction of context switching was reduced, but also the versatility of the email client was increased by complementing asynchronous email (to read at convenient time) with synchronous, instant message (to read at present time). Among all the existing email clients and IM protocols we choose to integrate Mozilla Thunderbird and Jabber/XMPP because they are open and offer great extensibility with ease of development. Mozilla Thunderbird[22] is a free, open-source and cross-platform mail client for most operating systems. It is built on the Mozilla Cross Platform Front End (XPFE)[23] an extensible framework that enables the development of fully portable plugins using CSS, RDF, and JavaScript.

The JabberPresence extension was released as an open source project under the MIT license and then, source code can be accessed via the Collab CDE.[24] JabberPresence is also publicly available for install at the official Mozilla add-ons repository site.[25] The extension allows Thunderbird users to:

- Specify multiple Jabber accounts.
- Associate Jabber IM buddies to contacts in Thunderbird address book.
- Signal one's presence information.
- Display contacts' presence information.
- Send and receive instant messages from within Thunderbird.

Differently from other IM systems, in Jabber multiple clients can connect at the same time, using the same account. The XMPP protocol allows for Multiple Points Of Presence (MPOP) by assigning each client a resource (i.e., a specific identifier in the form userid@domain/*Resource*) and a priority number for each resource (i.e., an integer from 0 to 10). Incoming messages and presence information will be routed to the

[22] http://www.mozilla.com/thunderbird
[23] http://www.mozilla.org/xpfe
[24] http://cde.di.uniba.it/projects/presence-ext/
[25] https://addons.mozilla.org/thunderbird/867/

highest-priority resource, which is known as the primary resource, although messaging a specific client is still possible by explicitly addressing the associate resource. This mechanism enables a user to connect with the same account from multiple locations or devices, but only to receive messages at the selected resource. MPOP and resource priority were fundamental for our intent because they empower users to intelligently route communications: One can always send messages to start new discussions from within Thunderbird. Instead, incoming messages are processed by the plugin only when Thunderbird is either the primary resource or it is explicitly addressed as the recipient of the instant message (e.g., userid@domain/*Thunderbird*).

Another benefit of using JabberPresence is that most of the public XMPP servers offer bridges to other proprietary IM protocols, so that you need to run neither multiple IM clients, nor a single, multi-protocol client. For instance, you can have your MSN and Yahoo contacts visible at once in the JabberPresence roster, just as if you were running multiple clients over multiple protocols, whereas you are only connected via JabberPresence to the Jabber network, letting XMPP servers handle all the protocol conversions complexities.

In the following, a description of JabberPresence is provided in the form of a typical usage scenario.

*Fabio is a team manager and he daily uses a Jabber IM client to get in touch with his remote collaborators. He thinks it is useful to check at a glance if they are online and available, so as to choose whether to send an email, an instant message, or even place a phone/VoIP call, depending on his needs. Moreover, Thunderbird is his email client of choice and thus, he decides to try out the JabberPresence extension.*

*After installing the plugin, Fabio configures his Jabber accounts by either clicking on the Jabber Account Settings link in the start page of Thunderbird or selecting the corresponding option under the Tool menu (see Figure 25).*

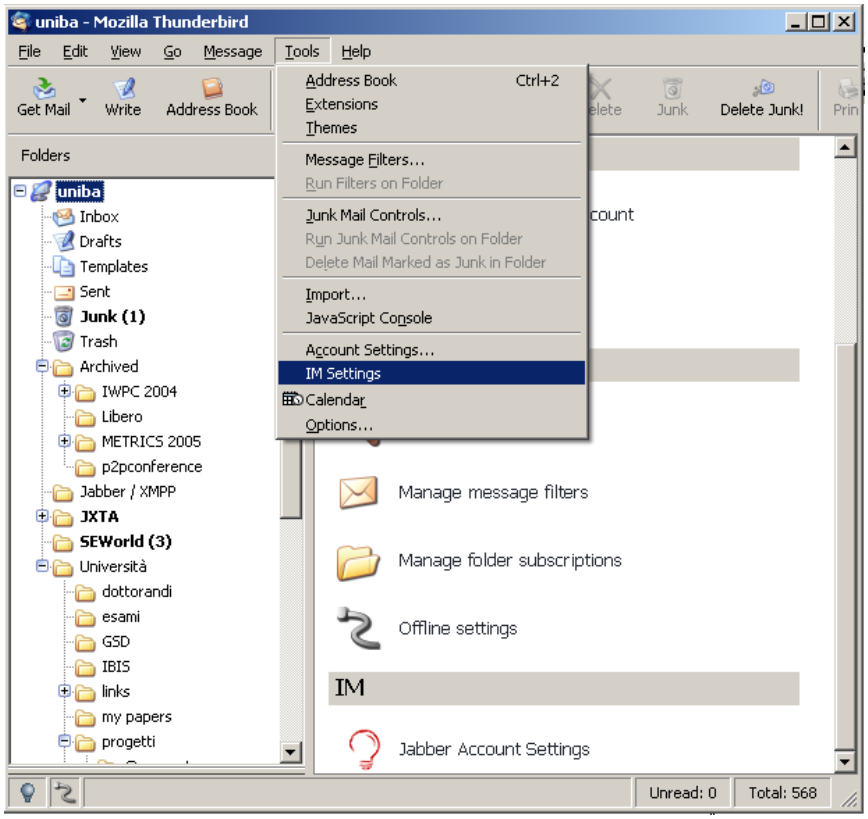


**Figure 25. Mozilla Thunderbird with JabberPresence installed**

*Afterwards Fabio fills out the fields in the dialog shown in Figure 26 to enter one or more of his existing Jabber accounts. Fabio edits the "resource" and "priority" fields to make sure that he will receive messages at the selected primary resource. From now on, when Fabio runs Thunderbird, the account flagged as default will automatically get connected.*

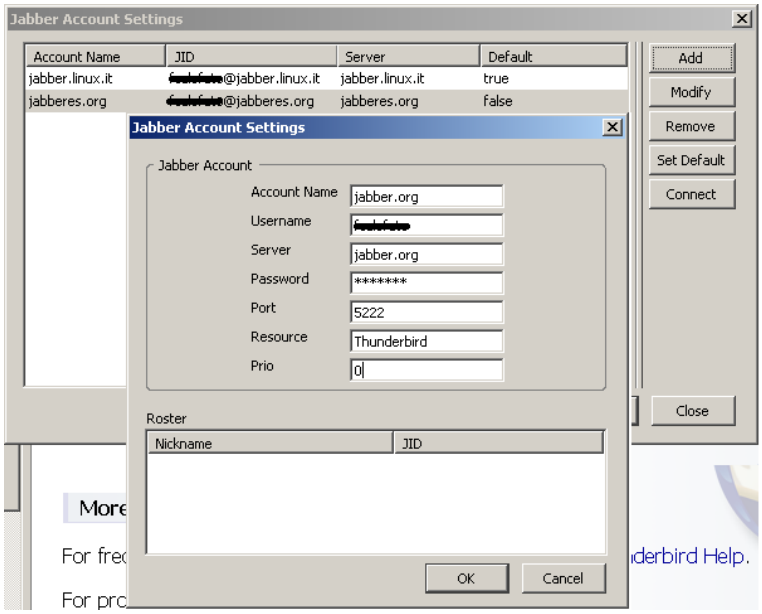


**Figure 26. Definition of Jabber accounts**

*At the first connection, the "roster import" dialog pops up, enabling Fabio to map his buddies in the Jabber roster to his contacts in the Thunderbird address book (see Figure 27), so that each buddy contact (the Jabber ID, or JID in short) is stored in the contact's vCard (see Figure 28). Obviously, the roster import dialog will also show up each time Fabio adds new buddies to the roster.*

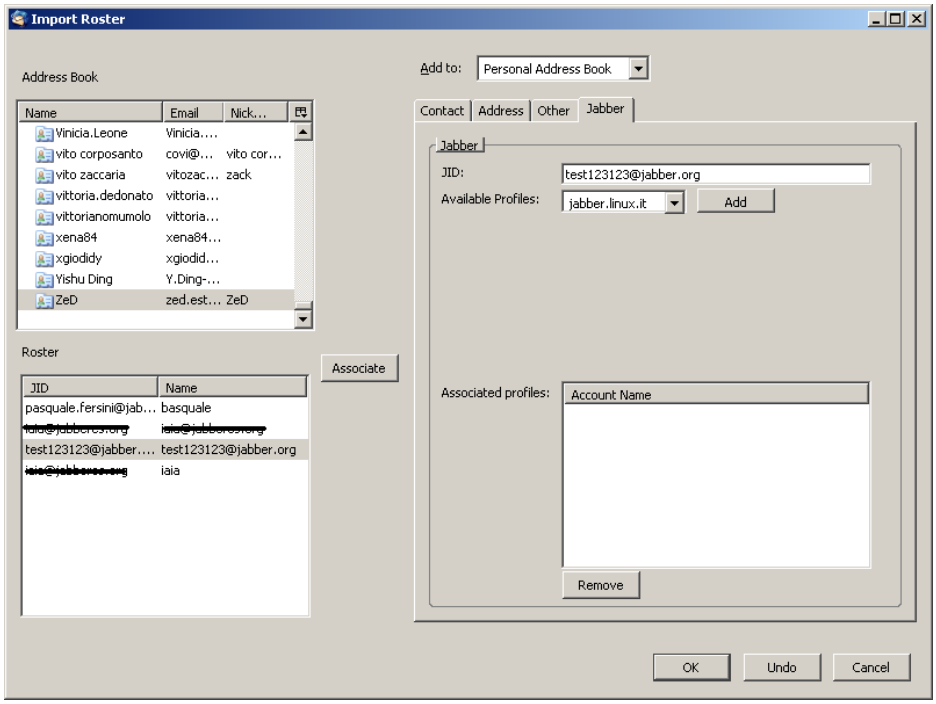


**Figure 27. The roster import dialog maps buddies to address book contacts**

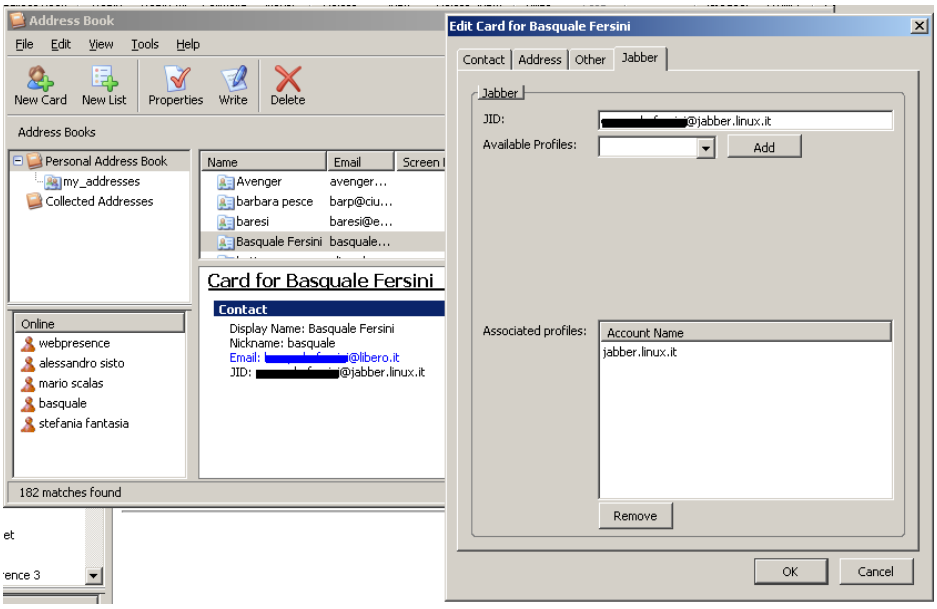


**Figure 28. JIDs are added to address book vCards**

*Once the contacts have been imported, Fabio can check the online status and availability of his buddies in a list placed in the bottom-left corner of the Thunderbird main window (see Figure 29). With a double-click, Fabio can send IM to online buddies directly from Thunderbird or, using the context menu, he can compose email when contacts are offline.*

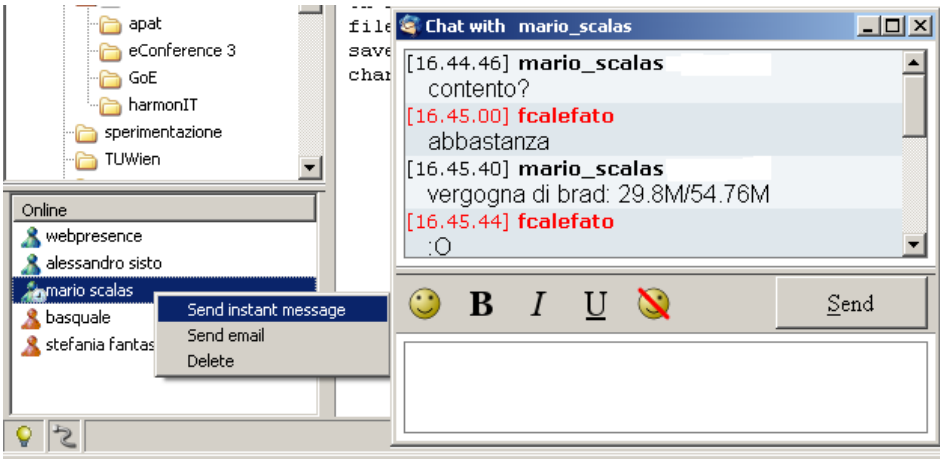


**Figure 29. Right-clicking on a buddy opens a context menu**

JabberPresence was by no means the first tool that embedded presence within an email client. Although other systems have communication features embedded (e.g. outlook with Live Messenger), not all of them integrate the message sending and receiving with the email client, or avoid the need to run the IM client as well. Usually, such systems only show presence status, but one should anyway switch context and open the IM client to send an instant message.

A similar plugin for signaling presence was developed for Kmail,[26] the open-source email client of the KDE desktop environment for GNU/Linux. This plugin integrates Kopete,[27] the KDE official multi-protocol IM client, with Kmail. Lately, two commercial email clients like IBM Lotus Notes[28] and Microsoft Outlook[29] have embedded presence awareness by integrating their own IM service, namely IBM Lotus

[26] http://kontact.kde.org/kmail/
[27] http://kopete.kde.org/
[28] www.ibm.com/software/lotus
[29] http://www.microsoft.com/outlook

Sametime[30] and Microsoft Live Messenger. In Microsoft Outlook when a user opens a message or views it in the reading pane, presence is shown through a smart tag, which is displayed beside the sender's name. This smart tag also appears in other places in Outlook, including the address book (when users rest the pointer on an email address) and new meeting requests (when users rest the pointer on an attendee's name). Users can right-click the smart tag to send an IM. However the IM is sent through the Live Messenger client, which users have to run as well. In Outlook Express, a simpler mail client embedded in Microsoft Windows, the full buddy list is also shown in the bottom-left corner. Despite the premise of integration, this solution still requires users to run the IM client anyway. With JabberPresence, instead, one can connect to Jabber and all these proprietary IM networks at once and send messages directly from Thunderbird.

IBM Lotus Notes is full-featured suite of collaborative including email client, calendar and personal information management, which also integrates IM. Apart from showing the status of contacts within emails and sending IM, the other notable features of Lotus Notes include the capability to start a group chat with all of the recipients and to add chat log to a mail body. The Lotus suite has only recently added support for Jabber/XMPP.

The comparisons made with the other free and commercial solutions available, suggested a number of ways to enhance JabberPresence. First and foremost, as a future extension, presence information will also be made visible in the From, To, CC and BCC fields, when reading or composing emails. JabberPresence does not support a multi-user chat (only 1-to-1 instant messaging). A useful enhancement will be enabling users to start a multi-user chat session inviting all the recipients of an email. Furthermore, chat contents can be a valuable source of knowledge to record and archive. Thus, at the end of each IM or multi-user chat session, users will be prompted to send an email to all the participants involved, with the conversation log added as an attachment. Finally, as security is always a major concern, we will add support for SSL secure connections.

Part of the features above have not been yet implemented due to the limits of the JavaScript library that we used: Jabberzilla[31] has lately been suffering from scarce

[30] http://www.ibm.com/software/sw-lotus/sametime
[31] http://jabberzilla.jabberstudio.org

development activity, and new compatibility issues arise as new updates of the Mozilla framework are released. XMPP4Moz[32] is a recently released JavaScript library, which has been used to develop Sameplace,[33] a Firefox add-on that adds IM and presence to the open source browser. XMPP4Moz is far more complete than Jabberzilla (e.g., supports SSL also under windows, multiple connections) and, above all, is now under very active code development. Currently XMPP4Moz only works with Firefox. However, Firefox and Thunderbird are both built on the Mozilla XPFE framework. As of this writing we are collaborating with the maintainer of XMPP4Moz to port the library to Thunderbird as well, so that to re-implement JabberPresence on top of the library. We do not exclude that in the future JabberPresence features will be merged with those of Sameplace, so as to realize a unique IM extension for Thunderbird.

## 6.6. Summary

In this chapter we have reviewed the history and focus of the CSCW field. We have focused on reviewing the benefits and pitfalls of email, the most-widely used and successful groupware application. One of the drawbacks of email is that, due to its success, people tend to use for a variety of purpose (e.g., scheduling, archiving, file sharing), and often in a synchronous fashion. In addition, email is 'socially blind' in that it does not enable users to signal their availability. Nevertheless, before becoming an indispensable tool ubiquitous in every workplace, email was initially used by the niche of research community and opposed by management. Nowadays, chat and IM are following a similar evolution path. At first mostly used by teenagers for exchanging 'social' messages, these synchronous tools have been recently spreading more and more in the workplace, although looked upon with suspicion by management as a source of continuous interruptions. While email is socially blind, these tools, in contrast, provide a lightweight means to ascertain availability and interruptibility of coworkers and contact them in a timely manner. In this chapter we have presented JabberPresence, a Mozilla Thunderbird plugin designed to merge the asynchronicity of email with the synchronicity of IM and chat, so as to better coordinate collaboration at a distance.

[32] http://dev.hyperstruct.net/xmpp4moz
[33] https://addons.mozilla.org/en-US/firefox/addon/3633

# Chapter 7:

# THE ECONFERENCE TOOL: HISTORY AND EVOLUTION

## 7.1. eConference

eConference is a text-based, distributed meeting system. Electronic Meeting Systems emerged in the early 90s to support group work by improving discussion [Nun91]. The primary functionality provided by eConference is closed groups chat with agenda, augmented with meeting minutes editing and typing awareness capabilities. Around this basic functionality, other features have been built to help organizers structure the discussion during distributed meetings. Indeed, eConference is structured to accommodate the needs of a meeting without becoming an unconstrained on-line chat discussion. The inceptive idea behind the eConference tool is to reduce the need for face-to-face meetings, using a simple collaboration tool that minimizes potential technical problems and decreases the time it would take to learn it. Our prototype has evolved through the years, first changing the underlying communication framework, from the JXTA P2P platform to the XMPP client/server protocol, which has proved to be a more robust and reliable solution to develop an extensible tool for distributed meetings. Then, in the latest version, eConference has evolved from a conferencing system to a pure-plugin collaborative framework, built on top of the Eclipse Rich Client Platform.

This chapter presents the development history of the eConference tool in terms of three major lessons learned, which may be of help when making strategic decisions that have the potential to affect the evolution process of communication tools. The remainder of the chapter is structured as follows. In Section 7.2, we first describe the first generation of our tool and then we discuss the problem encountered with JXTA. Section 7.3 illustrates the second generation of our prototype, along with XMPP and the impacts of its adoption. Section 7.4 describes the pilot experience conducted and its results, which have been used to further improve the tool. The third tool generation is discussed together with Eclipse RCP in Section 7.5. Finally, in Section 7.6 we show the forthcoming project for the next (fourth) generation of the tool.

## 7.2. 1st Generation: P2PConference

The first generation of our tool was named P2PConference[34] and was developed by the researcher as master thesis project [Cal04a, Cal05b]. The initial prototype was developed with the inceptive idea in mind of building a distributed meeting system that provided the basic features needed for supporting smooth discussion and facilitating meeting creation and execution, so as to maximize the tool effectiveness while minimizing complexity.

The tool GUI had six main areas: agenda, input panel, message board, hand raising panel, edit panel, and presence panel (see Figure 30). The agenda indicated the status of the meeting (‘started,’ ‘stopped’), as well as the current item under discussion. The input panel enabled participants to type and send statements during the discussion. The message board was the area where the meeting discussion took place. The edit panel was used to synthesize a summary of the discussion. The presence panel showed participants currently logged in and the role played.

[34] http://p2pconference.jxta.org

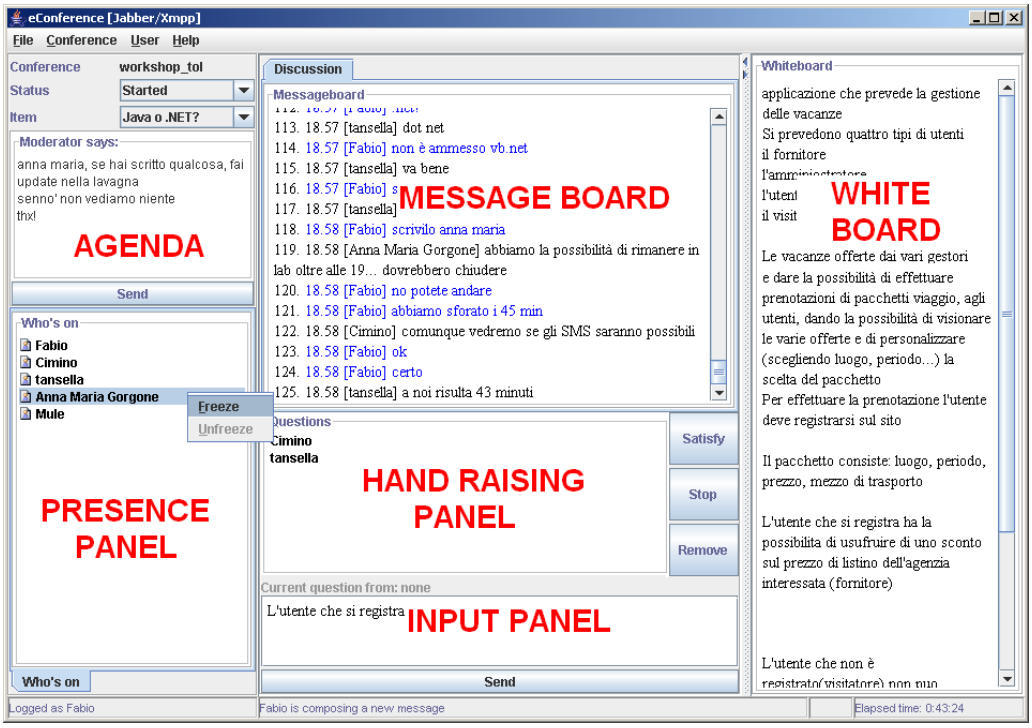


**Figure 30. A screenshot of P2PConference**

The organization of a meeting followed a strict protocol. Table 3 summarizes participant roles available and their rights.

**Table 3. Roles and rights in P2PConference**

| Role | Can speak | Must raise hand (presentation and panel only) | Can be frozen |
|---|---|---|---|
| Organizer | Yes | No | No |
| Moderator | Yes | - | - |
| Scribe | Yes (optional) | No | No |
| Speaker | Yes | No | Yes |
| Participant | Yes | Yes | Yes |
| Observer | No | - | - |

The meeting organizer was guided by a wizard through a few steps (see Figure 31) in order to:

1. Define the main topic and the agenda of the meeting.
2. Specify participants invited and their roles;
3. Schedule the conference and training sessions, if necessary.

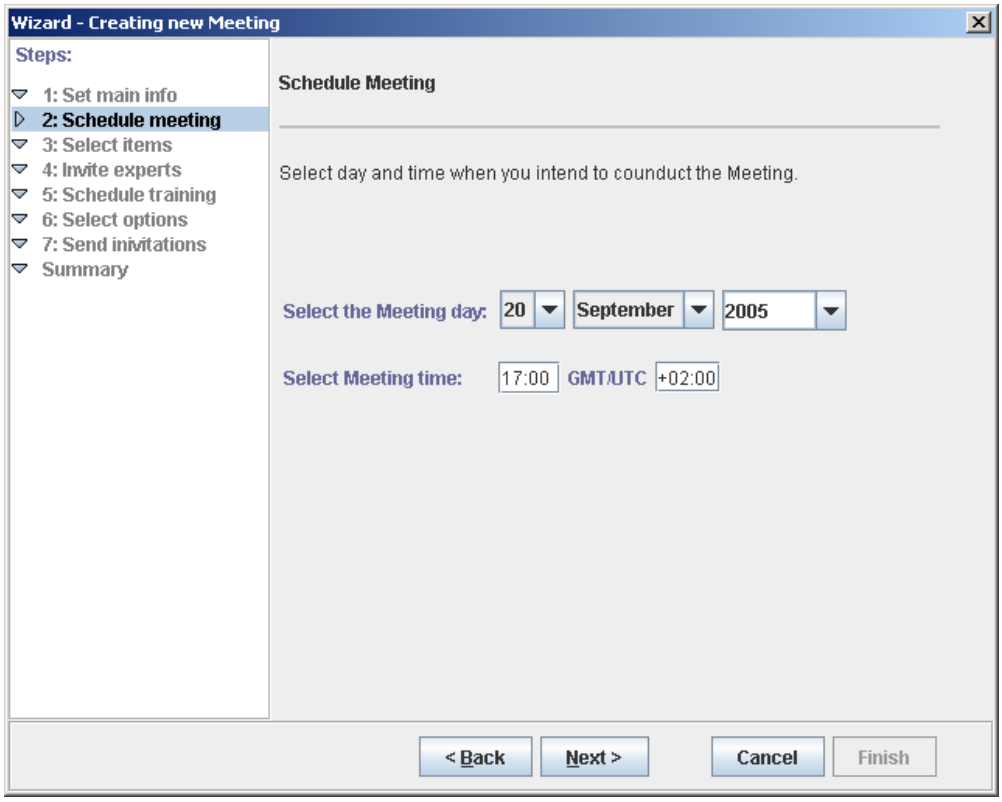


**Figure 31. The wizard for event organization**

Among the participants invited, the meeting organizer selected who would act as moderator and scribe. The moderator was supposed to facilitate the meeting and had control over participants, whereas the scribe captured and summarized the discussion in the edit panel. Thus, the content of the panel became the first draft of the meeting minutes (see Figure 32).

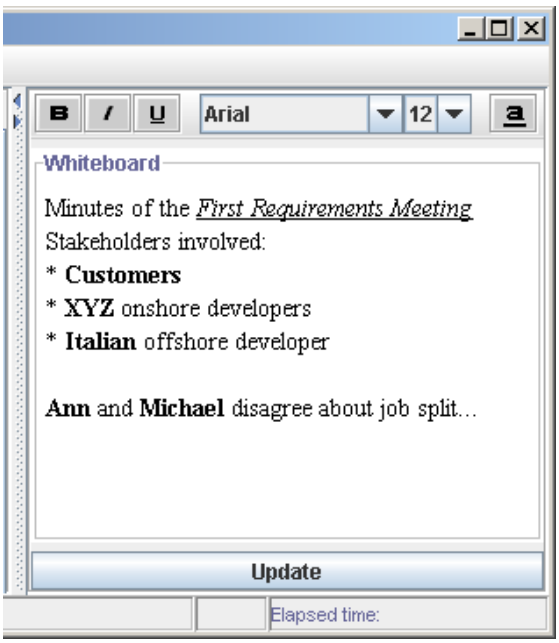


**Figure 32. Minutes being edited by the scribe**

Some participants might also be invited as observers, in that they would attend the meeting, but would not be able to actively contribute to the discussion. The selection of the event type was a fundamental step in the event creation process, as it would affect the interaction model for the meeting. In P2PConference there were three types of distributed meetings available, with an increasing level of control asserted by the moderator over participants during the discussion:

- *Brainstorm*: In a brainstorm, every participant could freely contribute to the discussion at any time. It ensured a limited control power since the moderator could only 'freeze' participants (i.e., frozen participants become observers, as they had been forbidden to type and send statements).
- *Presentation*: This event modeled a formal conference. One special invited participant, the speaker, delivered his/her own text-based speech, while the other participants could ask questions upon 'raising their hands.'
- *Panel*: It was a generalization of presentation, since there were more speakers, who acted as key participants with the aim of fostering the discussion.

During presentation and panels, the interaction of participants was driven by the use of the hand-raise feature, a mechanism to coordinate discussion and turn taking. In distributed meeting systems, turn-taking, or floor control, is a mechanism and policy under which users exchange possession of the floor, that is the temporary right to safely access shared objects or resources [Kos06]. In conferencing systems, the floor identifies the user(s) who can contribute to the discussion and the floor control policy provides a way to ask for the right to do it. Table 4 shows some common types of explicit floor control mechanisms for serial turn-taking, used by groupware so far. Lauwers et al. [Lau90] and Greenberg [Gre91] have recommended that collaborative systems should support a broad range of floor control policies to suit the users' needs. They believe there is no 'best floor-control policy' in absolute, and that the only certainty about floor control is that no single policy will suffice for all groups, in all situations. Thus, according to them, groupware needs to provide an alternative between different floor policies.

**Table 4. Summary of floor control protocols (adapted from [Gre91])**

| Policy | Description |
|---|---|
| Free floor | Any participant can enter input at any time. |
| Pre-emptive | Any participant can pre-empt control away at any time from the floor holder |
| Explicit release | The floor holder must explicitly release the floor before another participant may claim it |
| FIFO queue with explicit release | Participants line up to take turns, where the floor, once explicitly released by the floor holder, is given to the person at the front of the line |
| Central moderator | A moderator oversees all activity and decides who should hold the floor, usually by monitoring requests for the floor by other participants |
| Pause detection | The floor is made available to any participant only after the system detects a suitable pause of activity by the floor holder |

Not all of these policies are relevant for controlling turn-taking in conferencing applications (e.g. the pre-emptive policy). Thus, in P2PConference we implemented the free floor policy, available in brainstorm events, and the central moderator policy,

available in both presentation and panel events. During presentations and panels the moderator managed the queue of the questions/speaking requests from participants. The hand-raise feature also allowed the moderator to preview queued questions, showing a tooltip when hovering the mouse pointer over them (see Figure 33).

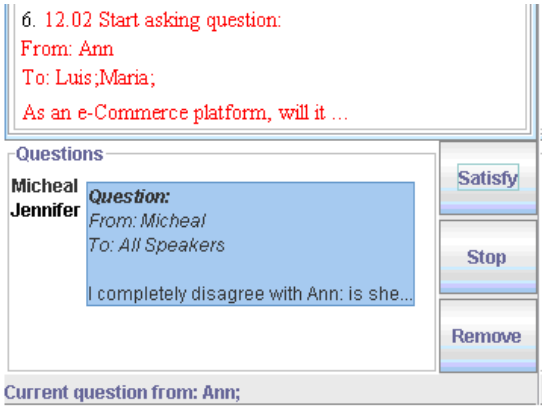


**Figure 33. Preview of a question as tooltip**

### 7.2.1. Project JXTA

P2PConference was developed using the Java binding of Project JXTA.[35] Project JXTA (in short JXTA henceforth) is an open-source project that provides a general purpose, language independent middleware for building P2P applications.

JXTA was originally conceived by Sun Microsystems and designed with the participation of a small number of experts from academic institutions and industry. In April 2001 JXTA 1.0 was released as an open-source project under the Sun Project JXTA Software License. All the contributions from volunteering developers are hosted at the JXTA community portal1 under the same license. JXTA is not an IETF standard. Although a first draft was submitted to IETF in June 2002, it expired at the end of 2004, since IETF declined to start a working group for JXTA. IETF, instead, referred JXTA to their sister organization IRTF to become a part of the IRTF Peer-to-Peer working group.[36]

[35] www.jxta.org
[36] www.irtf.org/charter?gtype=rg&group=p2prg

JXTA defines an XML-based suite of protocols that build on top of the existing physical network a virtual overlay network, with its own addressing and routing mechanisms [Gon01]. Figure 34 depicts the architecture of JXTA. The building blocks of the JXTA network are advertisements and peers. *Advertisements* are special XML documents that announce the presence of any JXTA resource and entity (i.e., other peers, services). *Peers* are any networked, digital device capable of running the JXTA protocols. In JXTA two categories of peers are defined: super peers and edge peers. *Super peers*, namely rendezvous and relay peers, deal respectively with the resources discovery and message routing. *Edge peers*, instead, are usually peers that reside on the border of the Internet, with transient, low bandwidth network connectivity, often hidden behind corporate firewalls. In JXTA, peers self-organize into peergroups. All peers belong to the NetPeerGroup, the default peergroup joined when booting the JXTA network, although they can create and join multiple custom subgroups, called user peergroups. Peergroups compartmentalization helps to set a scope for delimiting the search horizon when looking for resource, barriers for security policy hardening, and boundaries to message propagation.

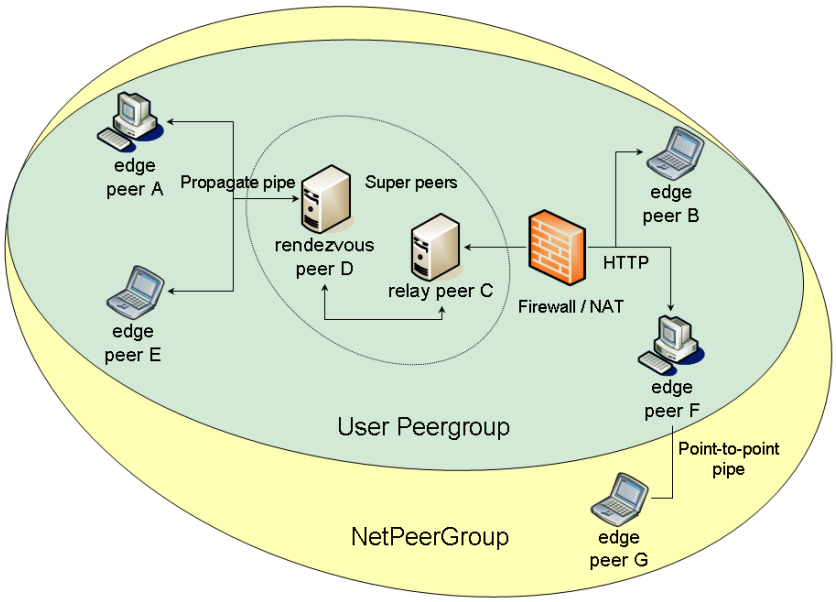


**Figure 34. Project JXTA messaging architecture**

A peer must belong to a peergroup before it can communicate with other peers, and it can communicate only with peers that have joined the same peergroup. In JXTA messaging architecture the fundamental abstraction used for inter-peer communication is the pipe, a virtual channel that consists of input and output ends. There are different pipes available JXTA. The core pipe services, available since JXTA 1.0, include unicast, secure and propagate pipes. *Unicast* and *secure pipes* serve for point-to-point communication, connecting two peers in unicast mode. *Propagate pipe*, instead, operates in one-to-many mode, leveraging either IP multicast on the subnet, or rendezvous peers. Core pipes are unreliable by definition and thus, they cannot guarantee ordered message delivery. Non-core pipe services, namely *bidirectional pipe* and *JXTA Socket*, provide bidirectional communication. Bidirectional pipe is available since JXTA 1.0, but has become reliable only since the second stable release of JXTA. JXTA Socket, available only since JXTA 2.0, is the only pipe reliable by design, and is fundamentally a reimplementation of the standard Java socket API upon the JXTA pipe infrastructure.

### 7.2.2. The Impact of JXTA

The choice of adopting a fully-decentralized, P2P approach stemmed from our intent of building a distributed meeting system easy to use and set up, with administration costs kept at minimum. JXTA seemed a promising technology because, by exploiting its virtual network, we aimed at using existing resources that live on the edge of the Internet infrastructure (e.g., bandwidth, storage). No central server to maintain and no a single point of failure is what the platform promised. JXTA did not deliver on all of its promises though. The remainder of this section discusses the problem encountered with JXTA.

**Low level API & End User Complexity.** The development of P2PConference started in March 2002 using the Java binding of JXTA. The first useable version of P2PConference was released at the end of 2002. The project was active during the year 2003, when file and browser sharing features were added, but was completely discontinued in 2004. Eight different releases of the platform were used for the

development of P2PConference (see Table 5). All the experiences reported and judgments expressed here refer to versions of the platform up to JXTA 2.3.

**Table 5. Impact estimation of JXTA platform changes (source [JXTA])**

| Version | Release date | Impact (compared to previous release) |
|---|---|---|
| 1.0 build 49b | 2002/02/08 | Low |
| 1.0 build 65e | 2002/07/08 | None |
| 1.0 final | 2002/09/24 | None |
| 2.0 | 2003/03/01 | High |
| 2.1 | 2003/06/09 | Low |
| 2.1.1 | 2003/09/16 | None |
| 2.2 | 2003/12/15 | Medium |
| 2.2.1 | 2004/03/15 | Medium |

None = No changes to API, bug fixes, other improvements

Low = New APIs

Medium = New APIs, APIs changes (deprecations, methods/classes removed, signature changes)

High = New APIs, APIs and Protocol changes (no backward compatibility)

One of the main disadvantages of JXTA was its overly low-level API, which made API users (developers) subject to frequent changes. Sometimes, as in the case of release 2.1, although the impact on developers was assessed as low, there were some platform incompatibilities that actually obliged us to update the tool. Indeed, as soon as the super peers were updated to the latest available release, we used to experience erratic behaviors (e.g., failure of resource discoveries, high rate of lost messages. Thus, not upgrading to the most recent release meant that we could not properly use fundamental services like routing or discovery, and run our distributed meeting system over the Internet in a truly distributed mode, but only in our subnet, using IP multicast.

A low level API ended up adding considerable amount of extra code and complexity. Furthermore, JXTA did not come with some basic services for a collaborative application, like a robust presence awareness mechanism. Thus, we had to develop from scratch a presence-broadcasting feature.

JXTA was not only complex for developers, but even for end users. In JXTA 1.0, the first time peers were started and each time their network configuration changed, users had to manually set up the platform through a wizard. The wizard was overwhelming to users because a plethora of settings were provided, not only about the network, but also about the JXTA platform itself (see Figure 35). Furthermore, it did not try to make any automatic setup (e.g., use of HTTP tunneling rather then TCP, behind a firewall, NAT). However, in JXTA 2.0 manual configuration could be bypassed trying to apply template configuration settings for well-known kind of peers (e.g., HTTP-firewalled edge peers or TCP rendezvous peers), but only with the release of version 2.2.1, configuration became fully automatic and sophisticated enough to work well without manual tuning most of the times.

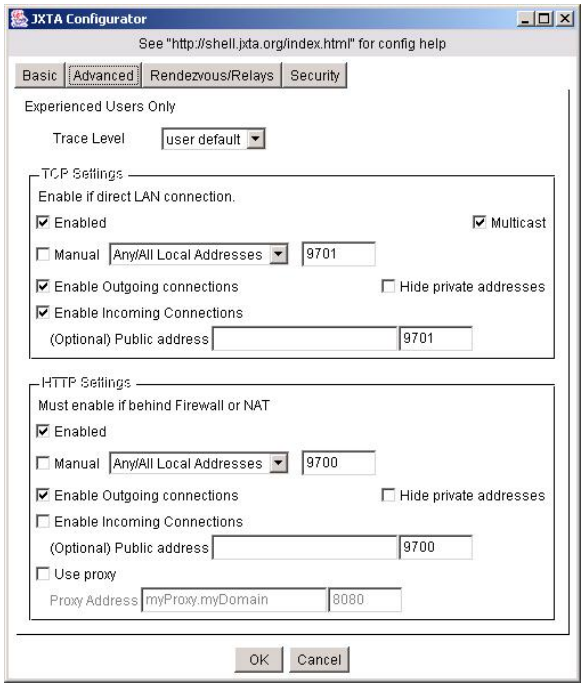


**Figure 35. JXTA configurator**

**Lack of reliable messaging mechanisms.** The main issue that forced us to abandon the P2P platform was the inadequateness of the JXTA messaging service. In JXTA the fundamental abstraction used for inter-peer communication is the pipe, a virtual channel

that consists of input and output ends. JXTA offered different alternatives to implement group communication in our prototype (see Table 6).

**Table 6. Alternative JXTA pipe services evaluated**

| Pipe service | Since | Type | Needs a server for group communication | Reliable |
|---|---|---|---|---|
| Unicast | v 1.0 | 1-to-1 | Yes | No |
| Secure | v 1.0 | 1-to-1 | Yes | No |
| Propagate | v 1.0 | 1-to-M | No | No |
| Bidirectional | v 1.0 | 1-to-1 | Yes | Yes (v 2.3+) |
| JXTA Socket | v 2.0 | 1-to-1 | Yes | Yes |

We chose to use the propagate pipe service in our prototype because of its one-to-many communication mode was the most apt for implementing group communication. Despite its unreliability, propagate pipe was actually the only practical solution, as all the other communication services were meant for point-to-point communication. Indeed, the use of any one-to-one service would have entailed the need to set up in the peer group a super peer that behaved very similar to a server (i.e. receive a message from a peer, then route it to all other known peers). This solution would have defeated any motivation for experimenting with a P2P approach, as it would have been equivalent to using a traditional client/server solution, but on a P2P platform and with much more complexity. Unfortunately, propagate pipes and discovery on rendezvous peers were unreliable, except when all the peers were in the same subnet using multicast. Instead, when peers were dispersed over the Internet, results were discouraging, with high message drop rate and low resource discovery recall. In [Osa06] Osais et al. encountered similar problems in the development of a collaborative tool (a shared web-browser), using a more recent release of JXTA (ver. 2.3). Due to the unreliability of propagate pipe, they had to develop a server for implementing group messaging, and concluded that JXTA is not an ideal solution for applications that require prompt and reliable group communication.

Although we have not collected data from formal tests or benchmarks, other research studies have somewhat confirmed the problems of the JXTA messaging architecture in general. Benchmarking JXTA is a hard challenge, and, hence, these studies are not complete and test results show a high variance because of the many different network settings and peer configurations to take into account (e.g., using multicast or rendezvous discovery, relay peers or direct connection, TCP, UDP or HTTP) [Ant05]. In their analysis of pipe services performance in versions 1.0 build 49b and 1.0 build 65e, Seigneur et al. found that unicast pipes behaved reliably only using TCP in local/LAN test scenario, whereas an extremely high message-drop rate was found when using HTTP [Sei02]. Halepovic & Deters tested performances of core and non-core pipe services for three JXTA releases (1.0, 2.0 and 2.2) in both LAN and WAN [Hal03]. Results reported somewhat confirmed that propagate pipes perform better in LANs, when UDP multicast is available, than in WANs. However, these tests are performed considering only one sender and an increasing number of receivers (1, 2, 4, and 8). Instead, complete tests on propagate pipe scalability should take into account the complex, realistic case of multiple senders and receivers in a large peer group over the Internet, messaging through relays and performing discovery on rendezvous.

### 7.2.3. Lesson Learned: Stability as a Key Aspect

Our experience with JXTA was not positive and a far cry from what we expected. JXTA was released in 2001 and, after having developed with it for over a year and a half, our feeling was that it had been released in a yet too-early stage, not mature enough, probably just on the heels of the growing popularity and hype of P2P. Although it aimed at addressing a real problem, fragmentation and redundancy of services offered by the plethora of existing P2P systems, JXTA failed at delivering a robust, general-purpose platform that can serve as the building blocks for P2P applications. Furthermore, JXTA proved to be not the ideal communication framework for developing a distributed meeting system. Paradoxically, its messaging services proved inappropriate for implementing group communication without using a client/server-like approach. Developing a spike (in agile development terminology, a brief proof-of-concept experiment to learn about an area of an application) would have probably

showed that JXTA pipe services were not suitable for many-to-many communication in pure P2P approach, and that the platform API was too low level and complex. The spike, however, would have never spotted the API instability issues along the releases.

When building a new application from existing components you make implicit assumptions or have expectations, which often turn out to be wrong or just do not match the actual environments [Gar95]. Stability is a key aspect of any API to guarantee the promised independence between API producers (software developers who write the API implementation) and API consumers (software developers who write code with method calls to the API). Changes in the API itself require changes in the API consumers' code because this code uses services provided by the API [deS04]. Although not the ideal scheme, to avoid these risks some component technologies, such as COM, use immutable interfaces, which cannot be changed once published. As API consumers, we did not expect the JXTA API to change often and we assumed the platform not to have backward compatibility issues as well. However, this is a more general issue in software platforms, because "so far, platforms have not sufficiently understood that software is supposed to be soft and thus, needs facilities that allow change" [Fow02].

## 7.3. 2nd Generation: eConference

Given the several issues we encountered during the development of P2PConference, we decided to port the tool onto a different communication framework. Our choice fell onto the Jabber/XMPP protocol, an IETF standard for messaging and presence services. P2PConference, refactored to use Jabber/XMPP as network backend, was renamed eConference.

### 7.3.1. Jabber/XMPP

The Jabber project started in 1999 to create an open alternative to closed instant messaging (IM) and presence services. In 2002 the Jabber Software Foundation[37] (JSF) contributed the Jabber core XML streaming protocols to the IETF, which approved them in early 2004 with the name XMPP, eXtensible Messaging and Presence Protocol

[37] www.jabber.org

[XMPP]. JSF continues to publish XMPP Extension Proposals (XEPs, henceforth), which define extensions on top of core XMPP technology. Lately, JSF and the development team of Google Talk, which uses XMPP for implementing its IM functionality, have defined the Jingle XEP-0166 [Lud06], a set of protocol extensions for adding real-time multimedia data exchange to XMPP, thus enabling Jabber clients to initiate audio/video chat.

Under the IM hood, Jabber/XMPP distributed architecture is capable of exchanging any data that can be represented in XML. Nowadays, Jabber/XMPP is being used to build not only a large and open IM network, but also and mostly, to develop a wide range of XML-based applications. The guiding principle behind the Jabber/XMPP development has been to keep most of the existing complexity at the server side, thus making easier to write both clients and other client-like entities that connect to servers. Jabber/XMPP is an example of hybrid client/server model, similar to email or DNS, in which entities are identified by a unique Jabber ID (JID), which usually takes the form `user@server`. As in the case of email, to exchange messages with each other, all that end users need to know is their unique identifiers. All that happens back-to-back between the servers for delivering the messages is entirely transparent to end users.

The building blocks of Jabber/XMPP are stanzas, XML elements sent over streams, which are connections established between clients and servers. When a client connects to a server, a stream is open in both directions (i.e., one from the client to the server and one from the server to the client). In addition, when a client (e.g., `romeo@montague.lit`) addresses a stanza to a non-local entity (e.g., `juliet@capulet.lit`), its server first negotiates a server-to-server stream with the foreign domain and then, sends the stanza over that stream for delivery to the non-local entity. Differently from other IM protocols, Jabber/XMPP supports Multiple Points-Of-Presence, that is, it allows a user to connect with the same account from multiple locations or devices: A resource name (e.g., `romeo@montague.lit/office`) and a priority number have to be specified for each connection. Thus, for instance, incoming messages and presence information sent to `romeo@montague.lit` will be routed to the highest-priority resource, the PDA in this case, although messaging a specific resource is still possible by explicitly using the form `romeo@montague.lit/office`.

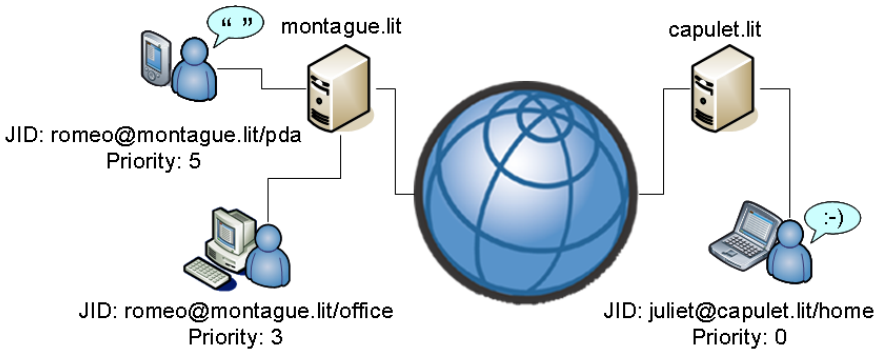


**Figure 36. Jabber/XMPP architecture**

Jabber/XMPP defines three core stanza types, namely `<message/>`, `<presence/>`, and `<iq/>`. The `<message/>` stanza is the messaging mechanism through which one entity pushes information to another in near real-time. The `<presence/>` stanza is a publish-subscribe mechanism through which status information about a given entity are propagated to other entities that have subscribed to it. Finally, the `<iq/>` (info/query) stanza provides a general purpose request/response mechanism, used for everything that is neither message, nor presence (e.g. contact-list management). These stanzas provide the transport layer, whereas the content of any given stanza, which is all pure XML, is specified by its child elements. More details on XML streaming with Jabber/XMPP can be found here [StA05].

Although Jabber/XMPP is all about XML, developers do not have to use raw XML directly. One of the reason for the success of Jabber/XMPP is the availability of high-level API for almost every programming language (ranging from Java, JavaScript and C# to PHP, Ruby, and Perl), which make the development even easier.

### 7.3.2. The Impact of XMPP

Compared to JXTA, Jabber/XMPP (simply XMPP henceforth) offered us three clear advantages: First, XMPP provided by design a robust, extensible, secure, and scalable architecture for near real-time presence, messaging, and structured data exchange. The second advantage was simplicity: XMPP has been conceived to delegate complexity to the servers as much as possible, so that developers can keep focused on the application

logic, and the clients can stay lightweight and simple. Furthermore, the intrinsic extensibility allowed to leverage and extend the services already available (e.g., presence, group communication). Third, the IETF standardization of the core XMPP protocols has generated a plethora of high level XMPP APIs, available for a number of programming languages. XMPP programmers do not even need to know the protocol details: All the raw XML exchanges are hidden by the use of any of these APIs.

At a first glance, replacing a P2P framework with XMPP might look somewhat contradictory. However in the hybrid, client/server architecture of XMPP, there are public interconnected servers that form the XMPP federation[38] (although running a corporate XMPP server, not part of the federation, is still possible). Hence, the XMPP federation allowed us to develop a client/server distributed meeting system, but without abandoning the goal of developing a tool that exploits resources already available, thus keeping at minimum the infrastructure costs.

### 7.3.3. Lessons learned: Complexity on server side Vs. Extensibility on client side

In our experience XMPP proved to be more stable, easy-to-use, and reliable than JXTA. Overall, XMPP is a good choice for applications that need an extensible messaging and presence framework. Indeed, its intrinsic extensibility has allowed us to easily expand the MUC capability, adding the extra functionality we needed to build eConference.

The use of the extremely high-level SMACK[39] library made programming XMPP even easier. Furthermore, the hybrid architecture of XMPP relieved us of many burdens and complexities, as compared to the P2P solution. However, centralization is no silver bullet and has its flipside. XMPP is capable of exchanging any data that can be represented in XML. To date, thanks to XML intrinsic extensibility, the XMPP transport layer has proved to be flexible enough to sustain our needs. But if you build your application on a protocol extension like the MUC XEP and you are not satisfied

[38] https://www.xmpp.net
[39] http://www.igniterealtime.org/projects/smack/index.jsp

with something – in our case, how it handles chat-history logging and its synchronization – there is not much you can do on the client side. The MUC XEP ensures persistency, delegating to servers the tasks of history logging and dispatching, but you can by no means alter it (e.g., avoid something to be logged, organize history by content, and retrieve parts one by one). To completely overcome it, we have to tackle the MUC history problem from the server side. That is, to develop a comprehensive solution we should either submit an extension proposal for the existing MUC XEP, or rather write on top of it a new XEP for a "structured MUC" that handles history synchronization at lower level. Writing a new XEP is a neat solution, in line with the XMPP philosophy (i.e., to move the complexity away from the client side). Nevertheless, to be accepted, any new XEP submitted has to go through the XEP standards process, which involves discussions on mailing list, formal reviews, voting by the Jabber Council, and, eventually, the approval as protocol extension. Thus, in the worst case, a new XEP submitted can be rejected at the end of the process, otherwise, in the best case, it will take several months and revisions before the draft becomes mature enough for public servers to decide to implement it, if at all.

With the adoption of XMPP, we traded the complexity and complete customizability of the JXTA solution with simplicity and reliability of the messaging and presence services, partially sacrificing extensibility.

## 7.4. Pilot Study with eConference

A pilot study with eConference was conducted in 2005 at the University of Bari. The tool was used to organize and run sixteen distributed requirements elicitation workshops. The participants were graduate students in computer science, attending a web engineering course. As final course assignment they were required to work in groups of three to five people and develop a web application, including both analysis and design documentation. All the workshops were conducted during the course, in a time frame of five weeks. The participants received one demo presentation of the tool. To provide further help, a detailed usage scenario was made available online. To simulate the geographical dispersion of the stakeholders, the students were allowed to

use the tool from home, as well as from laboratories in our department. Each group of students role-played a developer team, whereas the client team was the same for all the workshops and was constituted by one research assistant and two PhD students, who were also in charge of facilitating in turn each event. For each workshop, one of the developers was selected to act as scribe. Unlike JAD sessions, the scribe was also free to contribute information to the workshop. The meeting minutes edited by the scribe were the main outcome of a workshop. They contained a general description of the application to develop, a high-level list of the features to implement, all the decisions taken and the constraints, both technical and functional, imposed by the clients. Afterwards, the minutes were first used by the developers team, who edited a full requirements document for their own application, and then by the client team, who used it to crosscheck the same requirements document with IBIS, a web based inspection tool developed by our research group [Lan03b].

The goal of this pilot study was threefold. The first goal (G1) was assessing the general feasibility of conducting distributed requirements elicitation workshops using a synchronous, text-based communication channel. The second goal (G2) was understanding which type of meeting (controlled or open-format) is more successful for structured meetings like requirements workshops. Finally, the third goal (G3) was gathering useful suggestions to enhance the eConference tool and better support distributed meetings in general.

### 7.4.1. Data Collection and Analysis

We analyzed information collected from multiple sources and obtained results that were used to evolve the tool. The main sources of information were conversation logs and questionnaires, integrated with informal communication with the study participants. In particular, log analysis helped us to deepen the design flaws in features implementation, whereas the informal communication with the students allowed us to confirm our interpretations of data. In the next subsections we discuss findings with respect to the three goals defined.

#### 7.4.1.1. G1: Feasibility of Synchronous, Text-Based Requirements Workshops

Table 7 shows the duration, the number of actual participants and utterances exchanged, and the type of floor control policy for each requirements elicitation workshop. The duration was computed considering the time-span between the first and last utterance sent.

**Table 7. Characterization of the requirements workshops (workshops that employed central-moderator floor-control policy with hand raising are shown in grey)**

| Workshop | Duration (in min.) | Actual participants | Utterances | Floor control policy |
|---|---|---|---|---|
| W1 | 55 | 5 | 208 | Central moderator |
| W2 | 60 | 5 | 333 | Central moderator |
| W3 | 39 | 6 | 201 | Central moderator |
| W4 | 66 | 5 | 314 | Free floor |
| W5 | 63 | 5 | 250 | Free floor |
| W6 | 47 | 4 | 230 | Free floor |
| W7 | 47 | 5 | 268 | Central moderator |
| W8 | 36 | 4 | 138 | Free floor |
| W9 | 47 | 5 | 143 | Free floor |
| W10 | 43 | 4 | 134 | Free floor |
| W11 | 53 | 5 | 157 | Free floor |
| W12 | 45 | 6 | 301 | Central moderator |
| W13 | 48 | 5 | 154 | Central moderator |
| W14 | 54 | 5 | 378 | Free floor |
| W15 | 46 | 5 | 241 | Free floor |
| W16 | 38 | 5 | 203 | Free floor |

The longest workshop went on for 66 min., whereas the shortest for 36 min. Given the small standard deviation (8.7 min.), we can state that a workshop lasted in general for a little less than one hour (mean of 49.2 min.). The numbers of utterances exchanged in the workshops exhibit a high variance (mean=228, std. dev=75.3). The number of utterances exchanged was not influenced by the number of participants in the workshop. Indeed, we calculated the Spearman rank order correlation, but the result was not significant ($r=.39$, $p>.05$). Instead, useful understandings were obtained from the

informal interaction with the students, taking into account the composition of the groups. The two workshops with the largest number of utterances exchanged, namely W14 (378) and W2 (333), involved groups of client whose members had already been involved together in other previous course projects. In addition, they also were from different cities and thus, used to keep in touch by text chat. Hence, they actually behaved like an established group. Media Synchronicity and Channel Expansion theories (see Section 4.5 and 5.3) postulations confirm that communication effectiveness varies between ad hoc and established groups (i.e., newly formed groups and groups with an existing history of collaboration), since the latter are better able to communicate on a given channel. Being just a pilot study, group composition was not controlled and the students were left free to form the groups. Group composition and history are factors to be controlled in follow-up, empirical studies.

*In the next experiment we shall control the group history factor by managing group composition.*

After completing all the workshops, we sent by email a questionnaire to all the 46 students who participated. We received 43 replies, for a response rate of 91.3%. Among the questions asked, one aimed at understanding whether synchronous, text-based communication can be effectively used for running requirements elicitation workshops. The students were asked to rate on a 4-point Likert scale (4 = fully agree, 1 = strongly disagree) their agreement with the following statement: "Synchronous, text-based communication can be effectively used for conducting distributed requirements elicitation workshops." The breakdown of the responses is shown in Figure 37 and reveals that the majority of students (25, 62.5%) moderately agreed with the effectiveness of using a synchronous, text-based channel. The responses were subjected to the chi-square goodness of fit test, which showed that the students' moderate agreement was statistically significant at the 1% level ($\chi^2$=33.2, p<.01). This results show that, despite the limitation due to the leanness of text-based communication, the students perceived the tool as a useful means to conduct a requirements elicitation workshop.

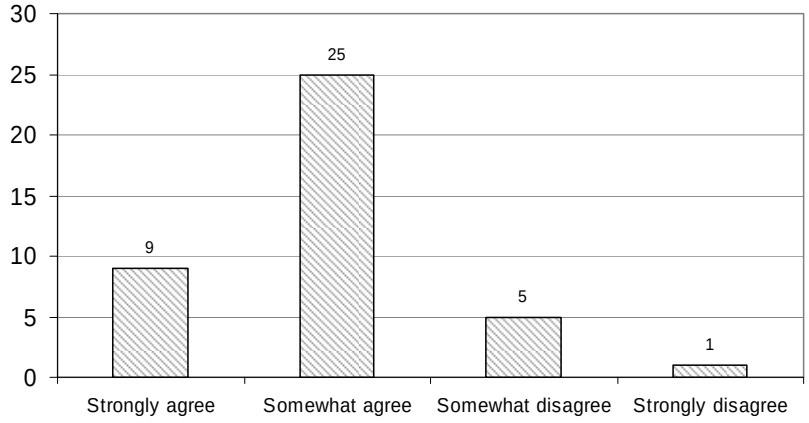


**Figure 37. "Synchronous, text-based communication can be effectively used for conducting distributed requirements elicitation workshops"**

#### 7.4.1.2. G2: Structured Vs. Open-Format Requirements Workshops

In [Mar99] Mark et al. reported an analysis of distributed meeting conducted at Boeing using NetMeeting. The authors found that the meetings were most successful when they had a formal structure or were facilitated. While the idea of facilitating workshops is widely accepted, especially in distributed requirements workshop (e.g. JAD), there is controversy about the need of imposing formal structure processes or a model of interaction to group in distributed, computer-mediated meetings [Nun91, Ros92, Boy93, Kos02, Yan04].

In our tool we implemented both the free floor policy (available for brainstorms) and central moderator policy for coordinating turn taking (implemented in presentation and panel events through the hand-raising feature). One of our intents was to examine how well the two policies worked in our context, that is, (1) how well the free floor and the central moderator floor supported the interaction of ad-hoc distributed groups for the given purpose; (2) whether and how imposing formal control influenced meeting execution.

When each requirements workshop was organized, we chose *a priori* the type of the event, and, thus, also which floor control policy would be used. Of the sixteen events, ten were organized as open-format meetings (brainstorm), and only six as structured

meetings (either presentation or panel). This discrepancy was due to the fact that the implementation of the hand raising proved not to work as expected already after the first six structured meetings. Indeed, the interaction driven by hand raising was ineffective for the type of interaction occurring in the workshops because it too strictly enforced a model of conversations based on questions and replies. Hence, we decided to re-schedule all the remaining events as open-format brainstorms.

After running the first six controlled meetings, we already understood that we had two major flaws in the design of our tool. Afterwards, the log analysis only confirmed our initial intuition. Looking at the conversation logs from these six events, we noticed that the hand-raising feature was deliberately forced into a free-floor policy by the moderator, and, hence, presentations and panels were actually turned into brainstorm-like events. Questions were treated as bare speaking requests and, at the very beginning of the event, the moderator allowed every participant one by one, thus leaving anybody free to participate at any time. As a matter of fact, from the analysis of six structured meetings logs examined resulted that a very few questions were asked (a mean of 4 questions per event), from a minimum of 2 questions in W13 (1.2% of all utterances), to a maximum of 9 in W1 (4.3%). W1 was somewhat of an outlier, though: Due to a network interruption, the event was split into two sessions, thus conversation had to be resumed and questions asked again. Furthermore, on average the questions asked within the first 8 minutes since the start of the discussion accounted for more than 70% of the overall questions.

The first flaw turned out to be selection of the type of meeting during the event creation. This solution did not prove to be flexible because choosing the event implicitly forced the use of the floor-control policy associated with that event type. It would have been a desirable feature for the facilitator to change the floor-control policy on the fly. For instance, the need of coordinating discussion using hand raising might have emerged also during a brainstorm.

*In the next release, the tool shall not distinguish among brainstorm, presentation and panel. The type of event shall be unique and all the floor-control policies implemented shall be available and interchangeable on the fly.*

The second design flaw of the current implementation of the hand raising feature was that it allowed only the speakers to talk freely, whereas the audience was only allowed upon requests accepted by the central moderator. This interaction model did not apply very well to the requirements workshops scenario, where there is a rich, bi-directional interaction among all the parties involved. Presentations or panels fit best a context where there is a neat separation between speakers and audience (e.g., education). Instead, this separation is not marked when the interaction occurs among customers, software analysts and other stakeholders, trying to capture requirements.

The limit of our hand-raising feature was that the implementation was trying to supplant the hand-raising social protocol, used in real meetings for polite turn-taking. Social protocols, defined as "standards of a polite behavior" in [Mor04], are a set of rules used as coordination policies by groups that need to manage collaboration and prevent/resolve conflicts. In contrast to technological protocols like our implementation of hand raising, social protocols are left to the control of participants. As Reagle pointed out, the important factor for collaborative tools that want to leverage social protocols is the degree to which the semantics and operation of social behavior/structure are captured within the data structures and protocol of the application [Rea99]. In our implementation of hand raising, the feature too strictly enforced a model of conversations based on questions and replies, and prevented anybody to jump in the conversation and interrupt somebody else. In real life, instead, it is still possible, though limited by the social etiquette and politeness of participants themselves.

In previous research there was controversy between researchers who believed that social protocols should be determined only by the group members, by the software, or by the hybridization of the two approaches [Gre91]. Both technological and social protocols have advantages and disadvantages. On the one hand, technical protocols ensure that a process is followed, providing more structure to group's activity. However, as we noted, technical protocols tend to be overly restrictive, since different groups perform similar activities in very different ways. On the other hand, leaving process to social protocols encourages collaboration, as the group must develop its own protocols. However, social protocols require collaborative tools to be very adaptive.

We decided that the next version of the tool should leverage social protocols for turn-taking, using an approach similar to the one proposed in [Yan04]. Yankelovich et al. presented an audioconferencing system, Meeting Central, which does not provide any floor control features for turn-taking (control channel, in their terminology). They claimed those characteristics to be intrinsic to humans and inherent in social protocols ruling any conversation in which people are involved, whether face-to-face or computer-mediated. Indeed, "conversation is a fundamentally social process" and "software control channel must be provided to facilitate the social mediation, not supplant it." Their philosophy is to let humans do what they are good at – mediating the social situation – and use technology to solve only those problems that people find difficult or impossible. Thus, Meeting Central counts on the power of social convention and socially defined roles to prevent users from usurping control of a presentation when it is inappropriate to do so.

Despite the differences between the communication media used by eConference and Meeting Central, namely text-based and audio, we decided to adopt the same approach for the evolution of our meeting system.

*Instead of imposing a model of interaction trying to supplant the real counterpart, the hand-raising feature shall foster the same social protocol used by people in real contexts to coordinate a discussion. That is, the hand-raise feature shall be re-implemented as a "polite turn-taking," and participants shall always be allowed to contribute to the discussion. All the user interface complexity that comes from imposing states and roles shall be eliminated (e.g., there will be no concept of a presenter or speaker).*

#### 7.4.1.3. G3: eConference Enhancement

One of the goals of this pilot study was also gaining insights on how to improve eConference. With electronic meeting systems complexity and usability are key problems. People want to focus on the content of their meeting, not on the meeting tool. Hence, features have to be chosen carefully to maximize the tool effectiveness while

minimizing complexity. In the first implementations of our tool, we decided to initially focus on core features, that is, on easing meeting creation and execution, and fostering communication. Thus, in the questionnaire sent to the students we asked them to rank on a 4-point ordinal scale (4 = most important, 1 = least important) four future extensions of the tool (namely, presentation sharing, web-browsing sharing, vote, and freehand drawing). We did not include file sharing because we considered it an indispensable feature, although not yet implemented in eConference. Analyzing the answers received (see Figure 38), we found that the students indicated a strong preference for the web-browsing (mean rank of 3.00), followed by presentation sharing (2.83) features, voting support (2.13), and freehand drawing (2.05). The rankings were subjected to the Friedman test, a non parametric alternative to two-way analysis of variance. The null hypothesis for this test is that the rankings given by each subject are random and thus, there is no systematic difference in the mean ranks of the features. The Friedman test result showed instead a significant difference in the mean ranks, at the 1% confidence level ($\chi^2$=16.77, $p < .01$). We also assessed the inter-subject concordance computing the Kendall's coefficient, which showed a somewhat little concordance between the students (W=.14). In [Yan04], Yankelovich et al. conducted an internal study at Sun Microsystems on top problems afflicting distributed meetings. They collected about 1800 questionnaires filled out by employees who mostly had to report about the typical problem they had experienced in distributed meetings. Employees were also asked to rank by importance the features they used. The highest rated were web-browsing sharing, freehand drawing, presentation and application sharing. Compared to results of the previous study, our findings confirm the importance of browser and presentation sharing, whereas data are conflicting with respect to the importance of freehand drawing, which was ranked by our students as the least useful feature.

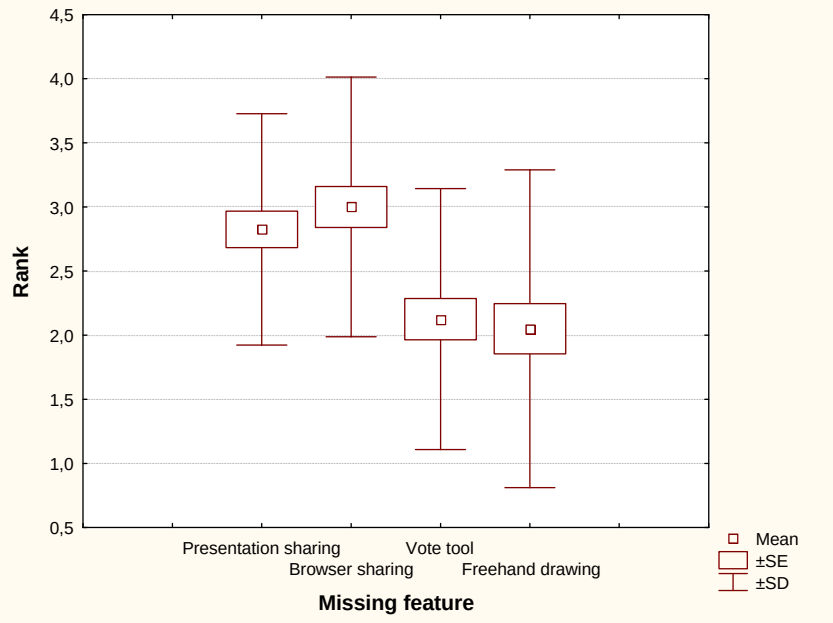


**Figure 38. Box-plot of the ranks for the most important features missing in eConference**

Besides rating the aforementioned missing features, we also asked the students to report any other feature they wanted to see implemented in a future version of the tool, although only a few students answered. The most common feature requests were about being able to add/edit/remove agenda items, draw UML diagrams in the edit panel, and send private messages to a single conferencing participant or whole groups (e.g., to the developer or client team). We thought that editing agenda when the meeting is going on would be useful for granting a greater flexibility. Drawing UML diagrams is certainly useful for some technical meetings, but useless for others. This feature was considered a serious candidate for being developed as a plugin. Instead, we were a bit skeptical whether private message could actually be a useful feature. The students motivated their request, making good points:

– *"I think that a separate chat shared only with same-group members could be useful when you don't want to make something public to others and vice versa."*

– *"Sometimes there were points I wanted to share only with my group and I could not do it."*

Nevertheless, we were anyway worried that this feature, if implemented, could be abused to the detriment of the discussion itself, especially in the case of private group communication. We decided for a tradeoff, and accepted only to implement one-to-one private messaging.

*The next version of the tool shall enable the moderator to edit the agenda while the meeting is running, and also provide one-to-one private messaging capability.*

The only technical problem that some students reported about in questionnaires was related to the scrolling of the message board panel when a new message was received. Talking informally to students about this annoyance allowed us to spot and deepen another issue that had not been revealed by questionnaires. Students perceived that the item based discussion helped to stay focused on the item currently at hand, but, sometimes, they needed to switch back to one of those previously discussed. In such cases, students found awkward to scroll up the message board, looking for the lines about that item among the very many contained in it. Moreover, the message board automatically scrolled down again as soon as a new message was received:

– *"I found the automatic scrolling of the message board very annoying. Particularly, when we needed to switch back to an item previously discussed and had to scroll up, looking for the lines about that item."*

To some extent, this issue should have been mitigated by having always at hand the minutes draft in the edit panel. However, our course was not on requirements engineering techniques and, hence, it was likely that students designated as scribes lacked training, and that the minutes draft did not always report all the decisions taken. Nevertheless, the feedback on this issue allowed us to understand that, to ease communication flow in eConference, we needed to have separate threads of discussion for each item available in the agenda. Such a feature would have avoided having a cluttered message board, with utterances about items interleaved with each other.

*Instead of mixing all the discussions for each agenda item in a single message board view, the next version of the tool shall provide a way to show a separate thread discussion per item.*

## 7.5. 3rd Generation: eConference RCA

From the pilot study we collected many useful requests of feature extensions, although some were specific for the requirements engineering context. However, it is overly challenging to foresee all the possible features needed to make a distributed meeting system flexible enough to be apt for all contexts. In addition, when we ported our meeting system from JXTA to XMPP, we lost some features (namely file and browser sharing), because they could not be easily adapted and needed to be rewritten from scratch. We wanted to avoid again all the effort spent in adapting the tool to support another communication platform.

These concerns led us to think about evolving eConference from a simple collaborative application to a collaborative platform. Our intention was to have a platform that offered as core functionality a reliable, extensible, and scalable messaging framework, on the top of which new features could be added as plugins.

### 7.5.1. Eclipse Rich Client Platform

To support the composition of a larger system that is not pre-structured or to extend it in ways that cannot be foreseen, an architecture that fully supports extensibility is needed. We decided to build another prototype exploiting the Eclipse Rich Client Platform (RCP) [McA05]. While mostly known as a powerful Java IDE, since the release of version 3.0, Eclipse has evolved towards an open and fully extensible "platform for creating other platforms." Thus, the Eclipse IDE is now only one of the Rich Client Application (RCA) that can be built on Eclipse RCP.

Eclipse RCP is a pure-plugin system, fully extensible by architectural design. This new modular architecture looked very attractive to us because it promised to help us in developing with a focus on modular functionality and writing new plug-ins for missing

functions. In *traditional-plugin* systems (Figure 39a), plugins are mere add-ons to extend the functionality of a host application, that is, binary components not compiled into the application, but linked via well-defined interfaces and callbacks. Instead, in *pure-plugin* systems (Figure 39b) plugins are the building blocks of the architecture, as almost everything is a plugin. In other words, the host application becomes a runtime engine with no inherent end-user functionality, as every application behavior is provided by a federation of plugins orchestrated by the engine [Bir05a].

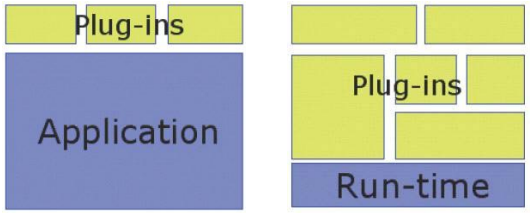


**Figure 39. Traditional (a) vs. pure-plugin (b) architectures (adapted from [Bir05a])**

The runtime engine of RCP is the Eclipse implementation of the specifications of a service platform for dynamic management of components, provided by the OSGi Alliance[40] [Gru05a].

### 7.5.2. The impact of Eclipse RCP

The 3rd and latest generation of eConference is a RCA, built upon Eclipse RCP. Besides all the benefits that come from using SWT/JFace and native GUI widgets, our tool has inherited all the capabilities of the RCP, in terms of extensibility and classical concepts from the Eclipse world, like views and perspectives. Figure 40 shows the "orthogonal architecture" [Raj96] of the 3rd generation of the eConference RCA. The communication protocol supported was again XMPP. We also kept using the SMACK library, which we already mastered and whose event-based model easily fitted into our architecture.

[40] www.osgi.org

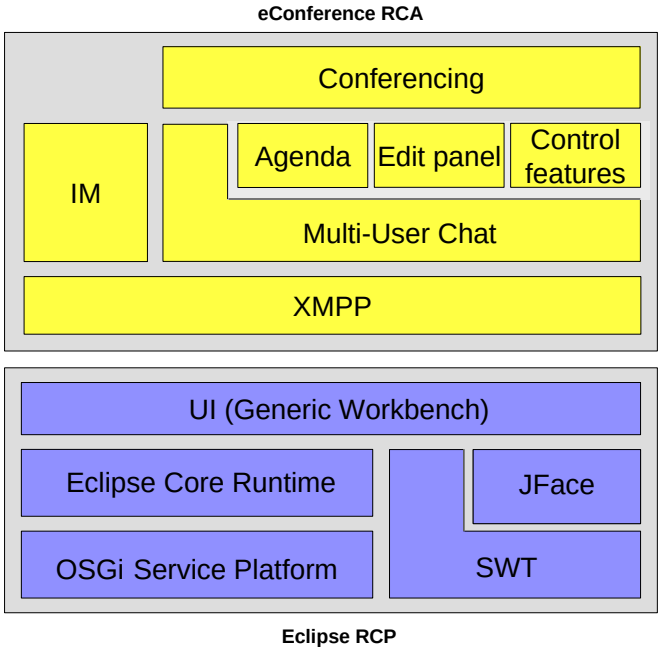


**Figure 40. The eConference RCA architecture**

eConference RCA was developed incrementally, using a story-driven agile process [Coh04]. In the following, we describe some of the epics, that is, the high-level, long stories that were later split into smaller and testable user stories, used for the implementation.

**Epic 1: A user can see presence status of contacts and send instant messages.** We started building a feature (i.e., a collection of plugins in Eclipse terminology) to provide both instant messaging (IM) and presence awareness capabilities. Both presence and IM are at the core of XMPP and thus, the mapping was almost effortless. Figure 41 shows the perspective associated with this feature.

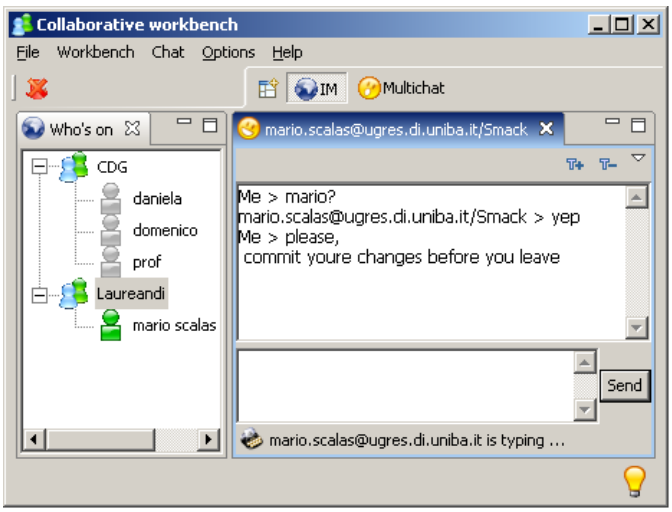


**Figure 41. The IM perspective with presence awareness**

**Epic 2: A user can create and join a chat room.** The second major step was to extend the existing feature to implement Multi-User Chat (MUC) for group communication. Unlike presence and IM, MUC is not a core functionality of XMPP. Instead, it is available as a XEP. The Multi-User Chat XEP-0045 [StA06] is the protocol extension proposed for managing chat rooms. Though not in the final stage, the MUC XEP draft is already supported by all the hundreds public servers belonging to the XMPP federation. One limit we found with the MUC JEP was that it did not handle typing awareness. We tackled this problem leveraging the intrinsic extensibility of XMPP and creating a custom typing-notification packet, sent whenever a participant in the room starts to type.

**Epic 3: A user can create and join an eConference.** Finally, leveraging the functionality already provided by the MUC feature, we developed new key plugins, namely the agenda, edit panel and hand raising, so as to obtain the overall "eConference feature" (see Figure 42). Indeed, rather than an application, in Eclipse terminology eConference is now just a feature of the RCA, with its own perspective. Similarly, when new features for web-browsing and presentation sharing have to be developed, we will build onto the existing features and plugins, and create new perspectives to optimize the arrangements of the UI views.

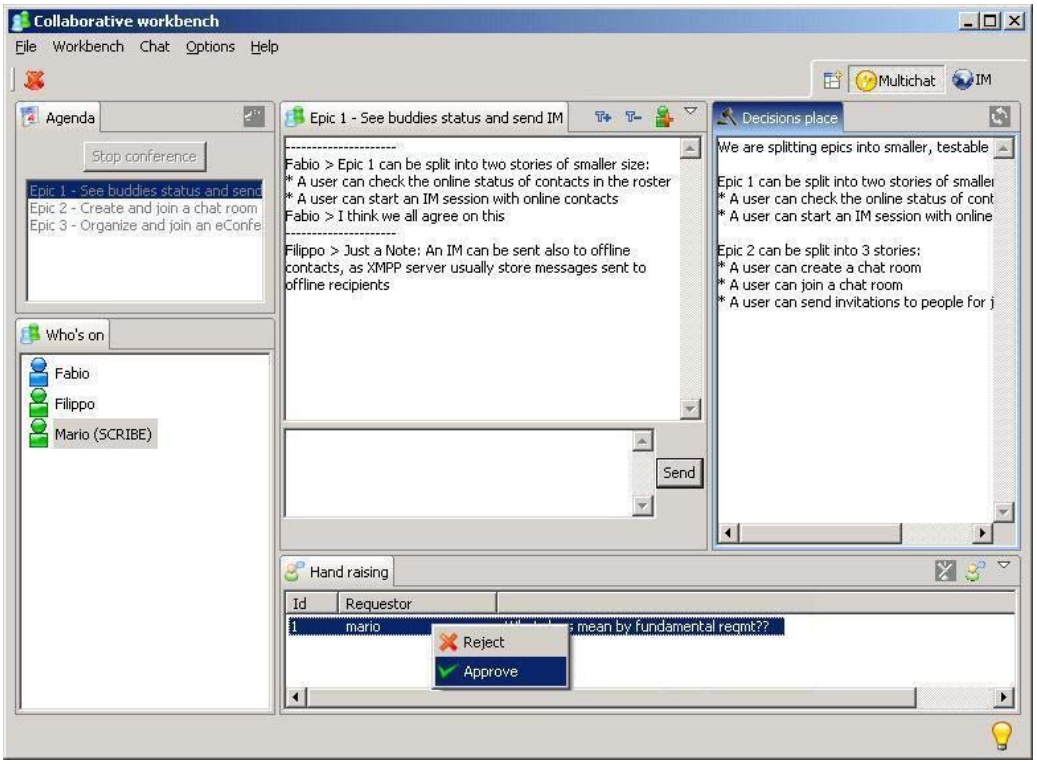


**Figure 42. The eConference perspective**

To implement the eConference RCA, we also took into account the feedback and suggestions gathered from the pilot study. Thus, we made the agenda editable by the moderator, when the meeting is already started, and added support for one-to-one private messaging. Finally, we also implemented the item-based discussion threads, so that all the utterances related to an item are grouped together. As soon as the moderator selects the first item in the agenda, say 'Epic1 - See buddies status and send IM', the meeting topic is changed accordingly (see the tab name in Figure 43a).

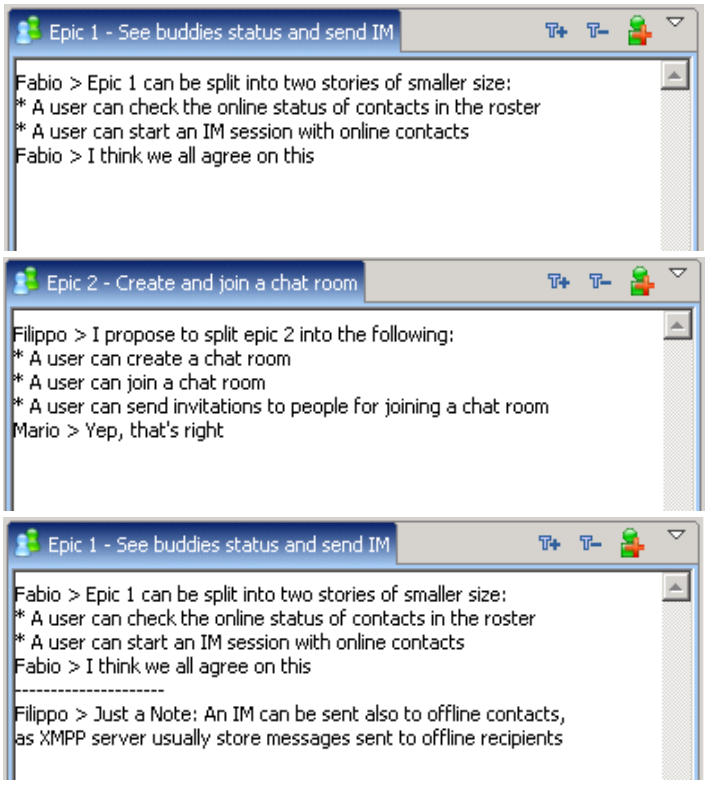


**Figure 43. The item-based discussion threads**

When it is time to move to the next item (say 'Epic 2 - Create and join a chat room'), the moderator selects it in the agenda and all the utterances about the previous item ('Epic 1') are hidden away from the messageboard, so as to show only the newly-entered utterances about the item at hand (Figure 43b). Suppose, for instance, that a note has to be added to 'Epic 1'. As soon as the moderator selects it back in the agenda, all the utterances previously exchanged will appear in the messageboard again. The dashes indicate a new session in a thread (Figure 43c).

### 7.5.3. Lesson Learned: Size Does Matter

Eclipse RCP is a platform for building other platforms. Its rich framework offers to developers a platform to build very advanced desktop applications, without reinventing the wheel. With a little more coding, this framework not only offered us a pure-plugin architecture, but also standard functionalities out of the box, such as a windowing

system, update functionality, help system, branding concepts, and all the other countless benefits seen in the Eclipse IDE. The only, but negligible, problems we encountered were some erratic behaviors during the process of product export (i.e., making the application executable outside of the Eclipse workbench) and the final size of the product itself. The application gets bloated because of all the Eclipse RCP libraries to be included, even if not all of its services are utilized. The size of the product for our prototype is almost 9 Megabytes, when the custom plugins developed, plus all the other third-party libraries we used, account for only 980 Kilobytes. This limitation is already known [RCP-BUG] and the Eclipse community is now working to reduce the minimal set of libraries needed.

## 7.6. Next Generation: eConference over ECF

To date three generation of eConference have been developed. Several iterations have been necessary to find good frameworks and build a flexible, collaborative tool. It is not unusual that a number of iterations are required to get a collaborative system right, and fully realize the benefits of frameworks [Boe01a].

The experience gained in developing the first two generations of our prototype helped us in identifying the basic features to provide in our tool and the characteristics that a communication protocol must have to work well in a distributed meeting system. In the third generation, the choice of Eclipse RCP gave us a means to build a system with greater flexibility and maintainability, capable of coping with change. However, RCP provides no facilities that allow for change also at the level of the communication protocol to be employed. When we implemented the second generation of eConference, the change from JXTA to Jabber/XMPP was not painless, but it cost the loss of the file and browser sharing. For the implementation of the next generation of our tool, we wanted more network protocols to be available with as little effort as possible. More specifically, we wanted a mechanism to support all at once multiple communication protocols through pluggable network backends, so as to make it possible to add support for a new protocol at any time, by only writing the specialized code for its integration. Hence, the next (fourth) generation of eConference will be built on the basis of the

Eclipse Communication Framework[41] (ECF). ECF is a set of plugins built on top of Eclipse RCP and designed for developing RCA that need cross-protocol support all at once. ECF provides a high-level abstraction layer that hides the supported protocol internals, which are handled by specific communication containers. Among the currently supported protocols there are JXTA, XMPP, Yahoo, IRC and JMS, but more are on their way (e.g., SIP/SIMPLE, Jingle). Figure 44 shows the orthogonal architecture for the ***eConference over ECF*** project, in which support for the browser and presentation sharing features will be added, as they the add-ons most requested by participants in the pilot study. This upcoming project has been one of the 43 projects in the world to receive the ***IBM Eclipse Innovation Award*** for the 2006 competition [EIA06]. The Eclipse Innovation Award[42] is a program sponsored by IBM to encourage teaching, research or community building around Eclipse and Eclipse-based technologies.

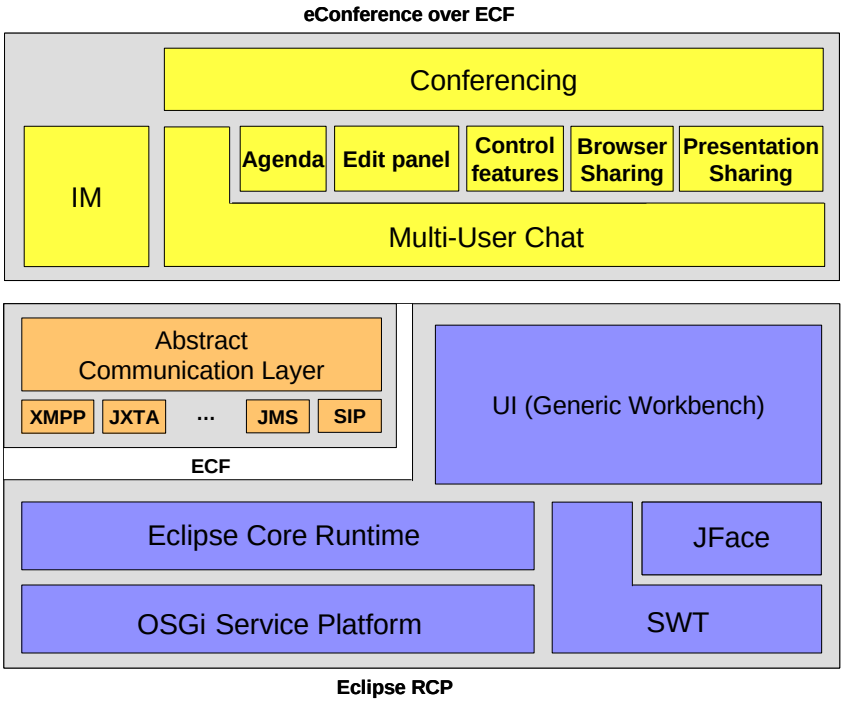


**Figure 44. Orthogonal architecture of eConference over ECF**

[41] www.eclipse.org/ecf
[42] http://www-304.ibm.com/jct09002c/university/scholars/products/eclipse/eig.html

## 7.7. Summary

This chapter has presented the eConference tool and its evolutions through three different generations. eConference started in 2002 as a P2P application (1st generation), based on the JXTA P2P platform. Then, due to the limits encountered with JXTA, in 2004 eConference was refactored to use the XMPP protocol (2nd generation). In 2005 eConference became a Rich Client Application, i.e., a fully-extensible, pure-plugin application, built on the Eclipse Rich Client Platform (3rd generation). Finally, the chapter has presented the project for the next (4th) generation of eConference, which will use the Eclipse Communication Framework for seamlessly offer the opportunity to use our tool, leveraging any of the many existing communication protocols. Using the Eclipse Communication Framework will make eConference an even better tool for supporting the synchronous communication of ad hoc groups, since they will be able to use our tool with their protocol of choice. This upcoming project has received the *IBM Eclipse Innovation Award* for the 2006 competition.

# Chapter 8:

# EVALUATING THE SUPPORT OF SYNCHRONOUS TEXT-BASED MEDIA IN DISTRIBUTED REQUIREMENTS WORKSHOPS

## 8.1. Goal of The Empirical Study

Over the past three decades, and particularly in the mid-1990s, many experimental studies on deployments of both desktop and classroom videoconferencing have been published. Some of these studies report about the successful interaction among remote sites, with no losses compared to F2F interaction [Ols97, Mar99], whereas others describe failures due to technical and behavioral issues [Tan93, Isa94, Ols95]. Today, despite the recent advances in video and audio technology and the increasing ability to create a rich medium for distributed meetings, the practicality of organizing videoconferences still remains low, due to the considerable overhead. The necessary infrastructure is expensive, awkward to setup and maintain at remote sites, and its coordination across organizational boundaries is often problematic [Pol05].

While there is an interesting body of knowledge about the comparison between F2F and audio/video technology, although with mixed results, past research on media effects

has not given the same attention to the comparison between F2F and synchronous, text-based interaction. Such disregard is probably due to the many theories of CMC (reviewed in Chapter 4) which recommended the use of rich media for complex tasks as the only possible solution. However, prominent theories such as Media Richness and Social Presence have strong face validity, but empirical evidence is rather equivocal [Den98a, Den99]. A number of studies of media use have provided evidence that runs counter to the predictions, particularly when media other than F2F communication are utilized, thus pushing researchers to theorize that media selection is also affected by factors beyond richness [Car99b]. Such theories fall short when considering context and task complexity for media selection. GSS-related literature has often reported about distributed groups who, while interacting on text-based channels, outperformed collocated groups in idea generation tasks and were instead outperformed in problems solving tasks [Mur00]. More recently, Birnholtz et al. proved the existence of collaboration settings, characterized by reduced information loads, where synchronous, text-based communication was adequate to achieve common ground among conversational participants unknown to each other [Bir05b]. These results suggest that to get the best out of media, an in-depth analysis of contextual factors (e.g., group typology, task complexity) is needed, and that all the CMC theories cannot be accepted or considered valid *tout court*, without giving due regard to the context. Instead, a common limit of both past and more recent studies is the evaluation of media effects on the execution of generic tasks, whereas executing realistic tasks requires individuals to apply known techniques or recall specialized knowledge to be performed [Mur00].

The goal of the empirical investigation described in this chapter is to evaluate the effectiveness of synchronous, text-based media support, as compared to F2F and in a well defined context: Ad hoc groups of stakeholders with no previous experience, involved in distributed workshops for eliciting and negotiating software requirements.

Requirements Engineering (see Section 2.3) is an appropriate domain for this study for a couple of reasons. First, it involves a complex set of communication-intensive tasks. Requirements elicitations and negotiations are among the most challenging and communication-intensive practices in software engineering [Mac96]. Further, requirements elicitation and negotiation are complex tasks that require a constant

interplay between idea generation, decision making, and conflict resolution activities, although in different measure (elicitation is more a generative task, whereas negotiation is more oriented to decision making). Secondly, recent research in the field has compared the support of audio [Llo02] and video [Dam03a] to F2F, but it has not given the same attention to synchronous, text-based interaction.

The remainder of this chapter is structured as follows. In Section 8.2 we develop the research questions for the empirical investigation, described in detail in Section 8.3. Section 8.4 provides a description of the empirical design, whereas Section 8.5 describes the instrumentation, training, and execution of the experiment. Section 8.6 depicts the data collection procedures. The measures and results are discussed in Sections 8.7 and 8.8, respectively. Finally Sections 8.9 and 8.10 discuss the findings and the threats to validity.

## 8.2. Development of the Research Questions

One of the commonly posed challenges by past research on media effects is to find evidence of the effectiveness of CMC interaction as compared to F2F, testing the following null hypothesis:

*$H_0$ – Computer-mediated groups perform as effectively as groups interacting face-to-face.*

However, any hypothesis similar to $H_0$ is ill-defined. Since no medium can now replace F2F communication, then the correct view of the challenge is that CMC represents an alternative with unique characteristics, which can be preferable in some situations to minimize or eliminate certain inconveniences. In other words, the correct challenge is to apply the most appropriate solution for the given context, defined by (1) the technology that supports group interaction, (2) the typology of the task that the group has to accomplish, and finally, (3) the typology of the group itself.

The context of this research is provided by its overall goal: To evaluate the support of (1) synchronous text-based communication (2) for conducting distributed

requirements elicitation and negotiations, (3) involving ad hoc groups of stakeholders. The figure sotto graphically depicts the context of this empirical study through the causal model suggested by Nunamaker et al. [Nun91].

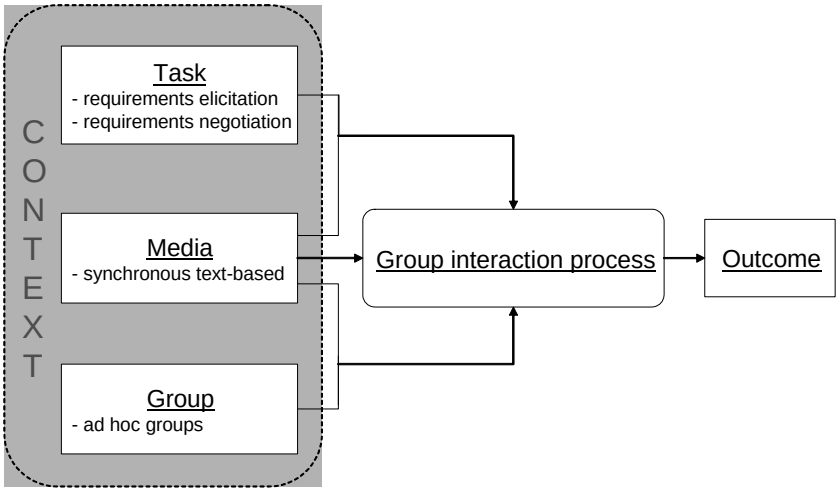


**Figure 45. The context of the empirical study shown in the causal model suggested by Nunamaker et al. (adapted from [Nun91])**

Given the scarce existing literature on the empirical evaluation of synchronous text-based communication in distributed requirements workshops, our investigation is somewhat exploratory in nature, since we could only rely on the theoretical background reviewed. Hence, in line with the background reviewed, we developed the following research questions:

- *$RQ_1$: How do synchronous, text-based requirements workshops vary from F2F requirements workshops, when involving ad hoc groups of stakeholders?*
- *$RQ_2$: Do synchronous, text-based requirements elicitation and synchronous, text-based requirements negotiation both represent an appropriate task/technology fit?*

We have investigated these research questions by means of an empirical study, described in the following sections.

## 8.3. Empirical Investigation

We conducted an empirical study of six academic groups, playing the role of stakeholders involved in requirements engineering activities. The six groups observed (Gr1-6) were attending a Requirements Engineering course held at the University of Victoria (Canada), from January to April, 2006. The study subjects were forty undergraduate students who volunteered to take part in the experimentation, after giving informed consent. Each team was constituted by five to eight randomly-selected students.[43] Randomization was the means used to form six ad-hoc teams of stakeholders. Furthermore, the projects were randomly assigned to groups before group membership was determined. Each of the six software projects was developed through the interaction of a client and a developer team. Table 8 shows the student groups, assigned to the six project teams.

**Table 8. Groups and allocation to projects**

| Project | Client team | Developer team |
|---|---|---|
| Project1 | Gr1 | Gr2 |
| Project2 | Gr2 | Gr3 |
| Project3 | Gr3 | Gr4 |
| Project4 | Gr4 | Gr5 |
| Project5 | Gr5 | Gr6 |
| Project6 | Gr6 | Gr1 |

As an educational constraint imposed by the course, the project assignment was done so that each student was involved in two projects at the same time, as either client or developer. For instance, students belonging to Gr1 acted as clients in Project 1, and as developers in Project 6. A description of each project can be found in Appendix D.

The goal of each project team was to develop a Requirements Specification (RS) document as a negotiated software contract between the developer team and the client team. The project work did not contemplate the writing of any code for the developer groups. Figure 46 illustrates the workflow of the requirements development process

[43] The terms students, stakeholders, and study participants are used interchangeably henceforth.

over a period of about ten weeks. It comprises ten phases of continuous requirements discovery and validation, through which the understanding and documentation of requirements was improved. Each of these stages consists of tasks for either one of the client/developer groups, or both groups (project tasks). The developers, together with the clients, created several versions of the Requirements Specification document (as a response to a Request for Proposals, RFP), while applying techniques of requirements elicitation and negotiation. The students were graded on the final version of the Requirements Specification document (RS 2.0), which reflects the shared understanding of the project that the clients and the developers built over the iterative process.

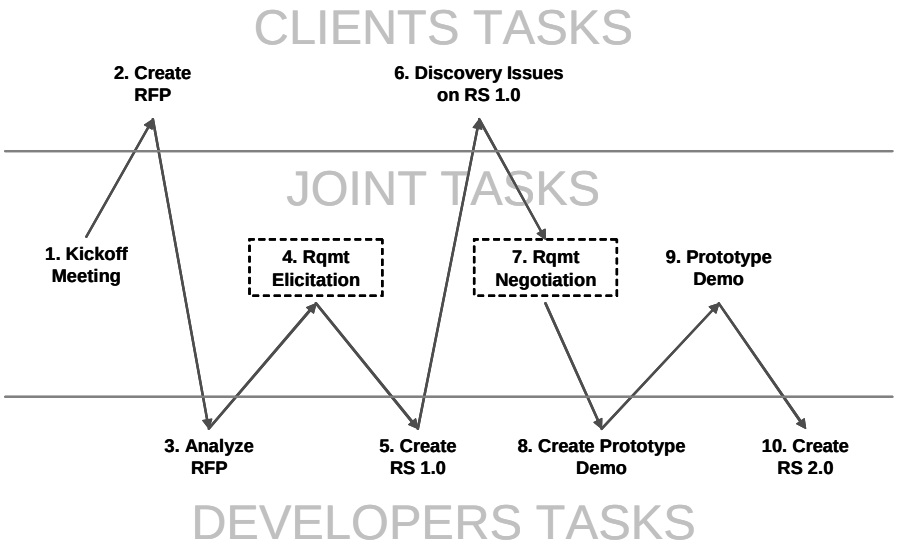


**Figure 46. Workflow for the development of the RS document**

A detailed description of the activities performed during each of the ten phases is given below.

1. ***Kickoff meeting.*** It was held as a one-hour F2F meeting, and aimed at giving the stakeholders of both groups the opportunity to start building social relationships with each other.
2. ***Create Request For Proposals.*** From the assigned project topic, the client group created a RFP document in order to call for developers' suggested solutions to the declared needs.

3. ***Analyze Request For Proposals.*** In response to the Request For Proposals, the developer group had three days to analyze it and define a list of clarifications about clients' need.
4. ***Requirements elicitation.*** The client and developer groups held a scheduled one-hour requirements elicitation workshop, *either F2F or using synchronous, text-based communication*. During the workshop, the developers clarified the clients' needs and elicited more requirements.
5. ***Create Requirements Specification RS 1.0.*** A week after the requirements elicitation workshop, the developers delivered an initial Requirements Specification document (RS 1.0). This document followed the IEEE standard for requirement specification and described in detail the scope and the features of the project.
6. ***Discovery issues on RS 1.0.*** Upon receiving the RS 1.0 document, the clients had a week to asynchronously carry on an inspection in order to identify the gaps in the understanding of requirements for the designer team. Each member of the client team participated in this phase, individually reading the RS 1.0 document available, and recording a description of the issues found.
7. ***Requirements negotiation.*** The issues identified in the Discovery phase were discussed during a scheduled requirements negotiation workshop, developers and clients communicated *either F2F or using synchronous, text-based communication*.
8. ***Create prototype demo.*** After the requirements negotiation, the developer group had one week to develop a prototype of the system to reflect the results of the negotiation. This prototype was a computer-based mock-up (i.e., did not have to contain any working code).
9. ***Prototype demonstration.*** A prototype demonstration was done through a one-hour F2F demo in order to let the developers show their understanding of both the project and the needs of the clients, who in turn gave their feedback to the developers, thus reaching a consensus on the project.
10. ***Create Requirements Specification RS 2.0.*** A last step, three weeks after the prototype demos, the developers submitted a final version of the

Requirements Specification document (RS 2.0). The RS 2.0 version integrated RS 1.0 with all feedback obtained from the clients through the requirements negotiation and prototype demo.

## 8.4. Experimental Design

As stated earlier in this chapter, the intent of the experiment conducted is twofold: To evaluate (1) the use of synchronous, text-based communication in distributed requirements workshops, as compared to F2F, and (2) the effects of CMC with respects to the different tasks of distributed requirements elicitation and negotiation. Table 9 shows the experimental plan, which corresponds to a $2^3$ factorial design [Mon96]. The three factors, each having two levels, are:

1. **communication mode** (levels: *F2F* and *CMC*);
2. **requirements workshop** (levels: *elicitation* and *negotiation*);
3. **role** (levels: *client* and *developer*).

**Table 9. Experimental design: $2^3$ factorial design with subjects as unit of analysis**

| | *A*<br>**Communication Mode** | *B*<br>**Requirements Workshop** | *C*<br>**Role** | **Subjects** |
|---|---|---|---|---|
| *(1)* | F2F | elicitation | client | Gr1, Gr3, Gr5 |
| *a* | CMC | elicitation | client | Gr2, Gr4, Gr6 |
| *b* | F2F | negotiation | client | Gr2, Gr4, Gr6 |
| *ab* | CMC | negotiation | client | Gr1, Gr3, Gr5 |
| *c* | F2F | elicitation | developer | Gr2, Gr4, Gr6 |
| *ac* | CMC | elicitation | developer | Gr1, Gr3, Gr5 |
| *bc* | F2F | negotiation | developer | Gr1, Gr3, Gr5 |
| *abc* | CMC | negotiation | developer | Gr2, Gr4, Gr6 |

The stakeholder-related observations, shown in groups for better readability, are the unit of analysis for this empirical design. The lowercase letter labels shown in the left-most column represent the notation chosen to identify the treatment combinations of the three factors, labeled instead, with the uppercase letters *A*, *B*, and *C*.

In the experiment, the *communication mode* and *requirements workshop* factors vary within subjects, whereas *role* factor varies between subjects. For instance, subjects in

*Gr1* interacted as *clients* in *F2F elicitation* workshop (treatment combination *(1)*), and in *CMC negotiation* workshop (treatment combination *ab*). Conversely, they participated in *CMC elicitation* and *F2F negotiation* as *developers* (treatment combinations *ac* and *bc*, respectively). Albeit in different roles, with this experimental design we obtained data from the subjects for comparing CMC to F2F communication for the purpose of conducting requirements elicitations as well as negotiations.

If we focus on the intent of comparing exclusively the support of the CMC mode to the two distinct types of requirements workshops, then the experimental plan will be able to be represented by an alternative design, obtained from the first design by dropping the communication mode variable, and retaining the requirements workshop and the role factors. Table 10 shows the alternative experimental plan, which corresponds to a 2x2 factorial design.

**Table 10. Alternative, 2x2 factorial design**

| | *B* **Requirements Workshop** | *C* **Role** | **Subjects** |
|---|---|---|---|
| *a* | elicitation | client | Gr2, Gr4, Gr6 |
| *ab* | negotiation | client | Gr1, Gr3, Gr5 |
| *ac* | elicitation | developer | Gr1, Gr3, Gr5 |
| *abc* | negotiation | developer | Gr2, Gr4, Gr6 |

## 8.5. Instrumentation, Training, and Execution

The requirements workshop sessions were instructed so that all the workshops would be held in parallel and be completed within an hour. F2F workshops (both elicitations and negotiations) were held in parallel, in the same classroom. Also the CMC workshops were all held in parallel, but the students interacted from three different laboratories, so as to simulate geographical dispersion. Each student was assigned to a given seat, so that to avoid whole teams to stay in the same laboratory, and some participants in the same workshop to sit side by side. The three laboratories were equipped with PCs running either Windows or Gnu/Linux OS, with Java 5.0 installed. Due to a course constraint, F2F and CMC requirements elicitations session were instructed so that each workshop would typically involve 2 developers and the whole

client group. In contrast, both F2F and CMC negotiations session were instructed to involve the whole project teams (i.e., all the clients and developers).

CMC workshops were run using the eConference tool. A dedicated XMPP server was installed on a machine available at the University of Victoria. Then, an account (JID) was manually created for every participant and sent by email, along with a randomly generated password. The server was dedicated to the study in that it was not part of the XMPP federation of publicly available servers and the registration of other accounts was disabled. No other instrument was utilized to run the experiment.

To let participants gain familiarity with the eConference tool, a one hour demo was given at class time. In addition, a user manual was created by the researcher and made publicly available on the course web site. Furthermore, to reduce the risks of technical problems, a training session was instructed a week before each CMC workshop session. During the training sessions the students installed the tool, checked their accounts (many of them changed the random password), and familiarized with the interaction style and features of eConference. The training was useful in particular to the students selected to act as moderators and scribes. The moderators were instructed by the researcher on how to use the agenda and the item-based discussion, whereas the scribes, who were required to log decisions taken, were instructed on the use of the decisions place editor. For the CMC elicitations, both the moderators and the scribes were randomly selected among the developer teams. For the CMC negotiations we asked the students who acted as moderator and scribe in the CMC session, to role-play again the same roles. One of the moderators refused, so the researcher instructed another volunteering student.

During the executions of the CMC sessions, the researcher, a teaching assistant, and a Ph.D. student stayed in each laboratory to provide technical support, and to ensure that no participant verbally interacted with the others. It was fundamental to the study that the participants of the CMC sessions did not have access to any visual or verbal cues, unavailable in text-based communication. Furthermore, since the tool also supports IM, we decided to disable the roster management, so that the students were not able to add buddies to the contact list and chat ‘off-topic’ with their friends during the workshops.

The CMC workshops were executed as follows. The moderators arrived in the lab 10 minutes ahead of schedule to populate the workshop agenda with the items they had prepared beforehand. Then, they waited for all participants to join the workshop. Upon selecting the assigned student as scribe, each moderator started the workshop with the discussion of the first item in the agenda.

## 8.6. Data Collection

The data sources for the experiment are the post-elicitation and post-negotiation-questionnaires, administered to the students about one week after each requirements workshop session. The students received the two post-hoc questionnaires in both electronic form (sent by email) and printed form (handed out during class). Students who returned the post-elicitation questionnaire were 20 out of 24 total participants (83%), whereas the response rate for the post-negotiation questionnaire was lower (19 out of 38, 50%). The post-elicitation and post-negotiation questionnaires are integrally reported in Appendix B and Appendix C, respectively.

The questionnaires were formulated taking into account the communication issues commonly experienced and already acknowledged by previous research in the requirements engineering field [AlR96], but also the issues informally reported by the students after each requirements workshop session. The questionnaires contained both open- and closed-ended questions. Closed questions, in turn, include multiple-choice questions and 4-point Likert scales. Some of the Likert scales in the questionnaires also contained matched-pair items, for direct comparisons of the treatments.

The most commonly used Likert scales in literature usually range between 4 and 7 points. Since the reliability of the Likert scale tends to increase with the number of items used, large scales (beyond 10-/13-points) are used as well. However, research has proved that while Likert-scale reliability largely increases from 2- to 5-points, the gain is limited and gradual beyond 7-points [Lis75, Jen77, Cic85, Cum00]. Thus, a 5-point Likert scale seemed appropriate, in particular to evaluate agreement when administering a satisfaction questionnaire (e.g., ‘5=strongly agree’ to ‘1=strongly disagree’). Cummins et al. reported of the benefits of adopting scales with an even number of

choices, which do not include a neutral mid-point, or central category [Cum00]. For instance, in a hypothetical 5-point agreement scale the midpoint would be '3=neither agree, nor disagree.' Adopting scales with an even number of choices is known as 'forced choice' method, and is intended to avoid respondents 'to sit on the fence' (i.e., the central tendency bias) by forcing an either positive or negative answers. Conflicting conclusions about the theoretical and psychometric necessity of the midpoint are reported [Har97]. Typically, midpoint responders have been classified as 'ambivalent,' 'indifferent,' and 'uncertain.' Theoretically, a midpoint response should represent ambivalent attitudes – an equal feeling of agreement and disagreement. Item Response Theory research has not supported the utility of the midpoint [ibid.]. Compared to 5-point scales, Garland found that 4-point scales (obtained by eliminating the mid-point from the previous one) mitigated the social desirability bias, arising from respondents' desire to please the interviewer, appear helpful, or avoid what they perceive to be a socially unacceptable answer [Gar91]. Thus, we decided to adopt 4-point Likert scales, anchored with '4=strongly agree,' and '1=strongly disagree.' Given that the study of synchronous, text-based communication in distributed requirements engineering is rather exploratory in nature, this choice was also consistent with the suggestion made by Johns who argued that omitting the use of mid-point on controversial topics improves validity [Joh05].

The questionnaires were formulated as follows. The day after the elicitation session, during the class, the students who participated in F2F and CMC elicitations were asked to write down their thoughts, ideas, and insights. Then, the researcher collected the sheets and created a transcript, grouping contributions on their content. Specifically, the contributions were tool improvements or deficiencies, considerations on the elicitation workshops, and comparisons between F2F and text-based communication, either in general or specifically for the purpose of eliciting software requirements. Considerations within each category were first collated and then merged into a single unit. Finally, the post-elicitation questionnaire was obtained by adding to the basic elicitation questionnaire, previously created by the researcher, those contributions thought to be relevant. By reformulating the students' contributions as closed questions

and adding them to the questionnaire, we could measure the level of agreement among all the study participants.

The formulation of the post-negotiation questionnaire followed a similar process: First, a basic questionnaire was formulated by the research, and then augmented with new questions, resulting from the contribution written by the students in class the day after the negotiation session.

In the remainder of this section we present the scales and the other closed-question groupings employed in the two post-hoc questionnaires. Henceforth, the following schema is used to refer to a specific a scale or question set in the questionnaires: *E-X* refers to the scale/set *X* in the post-*E*licitation questionnaire. Likewise, *N-Y* refers to scale/set *Y* in the post-*N*egotiation questionnaire. The scales E-A and N-A are two Likert scales that evaluate tool support during requirements elicitations and negotiations, respectively. Scales E-B and N-B compare, respectively, CMC elicitation to F2F elicitation, and CMC negotiation to F2F negotiation. Scales E-C and N-C are Likert scales with paired items for a more accurate comparison of F2F and CMC during elicitation and negotiation. Finally, E-D and N-D are two question sets containing several three-choice closed questions that aim at discovering which communication mode better supports a given task.

## 8.7. Dependent Variables and Measures

Satisfaction questionnaires are the only data sources in the investigation considered in this dissertation. Subjects' responses were then, coded to perform quantitative analysis.

To evaluate the differences between the requirements workshops and the communication modes through the subjects' perception, we conceptualized the comparisons in terms of two constructs, (1) satisfaction with performance and (2) comfort with communication mode, adapted from a similar investigation by Murthy & Kerr [Mur00]. *Satisfaction with performance* deals with the degree of satisfaction with performance perceived by subjects during the requirements workshops process. *Comfort with communication mode*, instead, deals with the degree of contentment with

either medium perceived by subjects during the requirement workshops. To operationalize these constructs, several questions were defined and added to the questionnaires.

With respect to the construct of satisfaction with performance, questions aimed at weighing both *discussion quality* and *consensus attainment*. We chose these two criteria because they strongly affect idea generation and consensus attainment which are the dominant activities executed, respectively, when performing the tasks of eliciting and negotiating software requirements. Discussion quality-related questions measured how focused, structured, in-depth, open, and understandable the discussion was. Consensus attainment-related question assessed instead, the extent to which decisions were made with group consensus, the ability to draw conclusions, the visibility and traceability of ideas generated.

With regard to the construct of comfort with communication mode, questions aimed at weighing the degree of *discussion contentment* and *engagement level* of the stakeholders. We selected these criteria because we wanted to assess how media affect the opportunity to actively participate in the discussion and openly discuss conflictual issues. The questions related to discussion contentment measured the appropriateness of interaction pace, the ability to express, spontaneity and familiarization with other individuals. The questions related to the engagement level measured instead, motivation to participate of self and others, participation equality.

We evaluated consensus even for elicitations workshops because, while not as relevant as for conflictual tasks like negotiations, some consensus is still needed in generative tasks. However, the two post-hoc questionnaires were slightly different, since they contained exclusive, task-specific questions (e.g., a question that evaluated the ease in requirements prioritization was asked in the post-negotiation questionnaire, but not in the post-elicitation questionnaire).

To ensure the validity of the constructs defined above, principal component analysis was performed. Principal component (or factors) analysis is a procedure that discards poorly-correlated questions and retains only those that account for a large amount of the total variance in the components data set, thus confirming the existence of the hypothesized components [Hat94]. We also performed scale reliability analysis to

further determine the internal construct validity by assessing the extent to which a set of questions measures a single latent variable. We used the Cronbach's alpha coefficient, the most-widely used index of internal consistency in social sciences [Cro51].

## 8.8. Results

We received 20 post-elicitation questionnaires, 17 of which were from participants in both F2F and CMC elicitation workshop. The post-negotiation questionnaires were 19, and all of them were from subjects who participated in both F2F and CMC negotiation. Finally, on the overall 39 questionnaires received, 10 were from subjects who, albeit in different roles and for different projects, participated in all the four kinds of requirements workshops (i.e., F2F elicitation, CMC elicitation, F2F negotiation, and CMC negotiation), and 13 were received from participants in both CMC elicitations and CMC negotiations. We employed nonparametric statistics, due to the limited sample size and the impossibility to rely on the assumption of normal data distribution.

In the next two sections we present the results from the quantitative analysis of data collected from the questionnaires.

### 8.8.1. F2F vs. Synchronous, Text-based Requirements Workshops

The results from the analysis of questionnaires are presented here, distinguishing between data collected from Likert-type scales and multiple-choice closed questions.

#### 8.8.1.1. Results from Likert-type Scales Analysis: Satisfaction with Performance

Figure 47 and Figure 48 show the breakdown of the responses to scales E-C and N-C, employed to assess subjects' perceived satisfaction with performance in terms of the extent to which the decisions made in the workshops were consensus-based (paired item

E-C.1 and E-C.2)[44] and all the generated information was not missed (paired item N-C.1 and N-C.2).

The two figures below show the responses given by the ten stakeholders who participated in all the four combination of requirements workshops (i.e., F2F and CMC elicitations, F2F and CMC negotiations) and returned both post-hoc questionnaires.

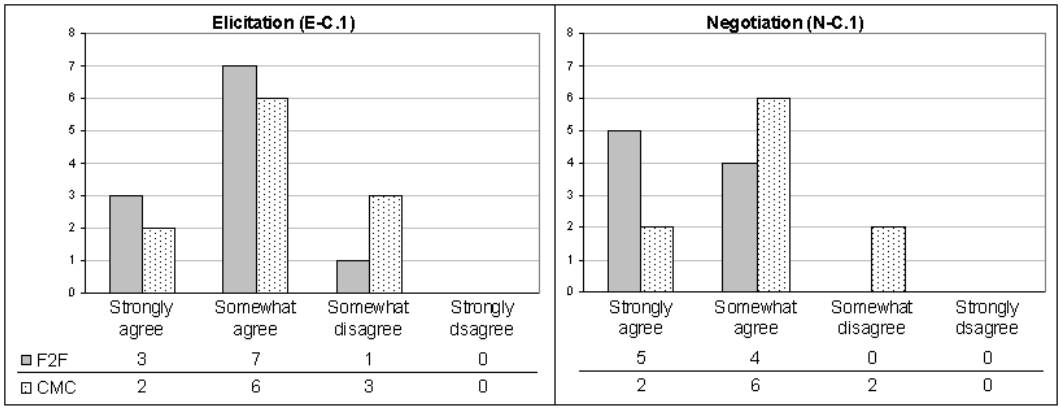


**Figure 47. "During the requirements workshop, decisions were made with a group consensus"**

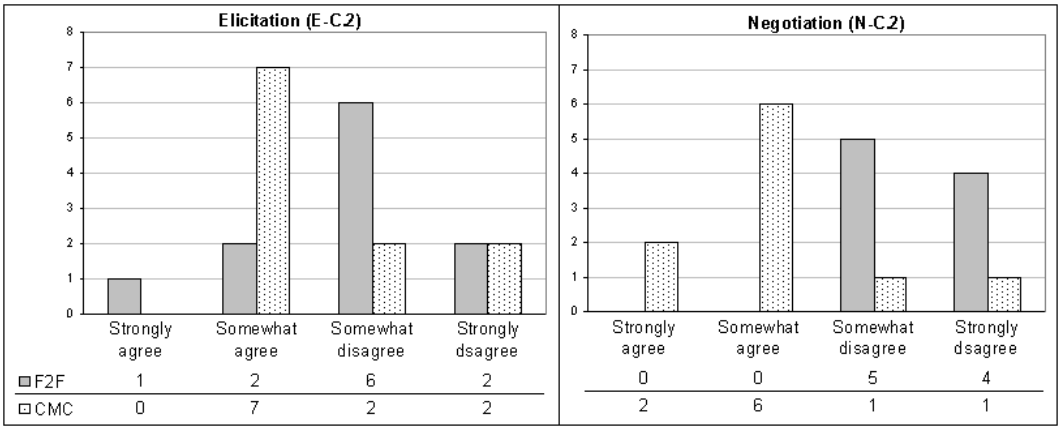


**Figure 48. "During the requirements workshop, the discussion was too fast and information was missed"**

We executed the Friedman test on the response set of scales E-C and N-C, as a non-parametric alternative to the within-subjects analysis of variance for multiple dependent samples [Con80]. The purpose of applying this statistic is to determine (1) whether

[44] E-C.1 indicates the 1st question of the scale E-C. Likewise, N-C.2 refers to the 2nd question of the scale N-C, and so on.

there are significant differences in the level of subjects' satisfaction with performance between the four workshop-medium combinations, and (2) whether subjects' responses are in agreement.

Before executing the Friedman test, responses to items E-C.2 and N-D.2 were reversed (i.e., strong agreement became strong disagreement, and so on), since this paired item had been formulated in a way it captured dissatisfaction for missing information generated. After that, for each subject, the responses were first summed, so as to obtain an overall score of the personal level of satisfaction with performance during the requirements workshops. Then, the ranks of the four workshop/medium fits were calculated on each per-subject summed scores (see Table 11).

**Table 11. Summed scores for comparing the four combinations of requirements workshop**

| Subject | F2F Elicitation | CMC Elicitation | F2F Negotiation | *CMC Negotiation* |
|---|---|---|---|---|
| S1 | 7 | 7 | 8 | *4* |
| S2 | 6 | 7 | 6 | *6* |
| S3 | 7 | 5 | 7 | *6* |
| S4 | 6 | 6 | 7 | *4* |
| S5 | 6 | 4 | 8 | *4* |
| S6 | 5 | 4 | 6 | *5* |
| S7 | 5 | 6 | 7 | *5* |
| S8 | 6 | 4 | 7 | *5* |
| S9 | 6 | 5 | 6 | *5* |
| S10 | 7 | 7 | 7 | *7* |
| *Total* | *61* | *55* | *69* | *51* |

For this scenario, Friedman test determines if there is a difference in the scores between the four workshop-medium fits. The role factor is confounded with the interaction between the other two independent variables, communication mode and requirements workshop. For each subject, the ranks of the four workshop kinds were calculated on the summed scores ($4^{th}$ rank corresponds to the highest score, $1^{st}$ rank to the lowest). The box plot in Figure 49 shows F2F negotiation to exhibit the highest, or best, mean rank (3.5) followed by F2F elicitation (2.75). CMC elicitation and CMC

negotiation have the lowest average ranks (2.15 and 1.6, respectively). In addition, F2F and CMC negotiations exhibit a smaller rank variability compared, respectively, to F2F and CMC elicitations. The null hypothesis for the Friedman test is that the distribution of the ranks for each combination is the same. The test result indicates a statistically significant difference between the ranks at the 5% significance level ($\chi^2$=14.54, p=.002) and, consequently, the null hypothesis is rejected.

Along with Friedman test, we also assessed the concordance level in subjects' agreement, using the Kendall's W coefficient of concordance [Ken47]. In this case, the coefficient was measured to be .49, a halfway result between the complete disagreement (W=0) and the complete agreement (W=1). Hence, we cannot establish whether the scores are correlated more than it would be expected by chance.

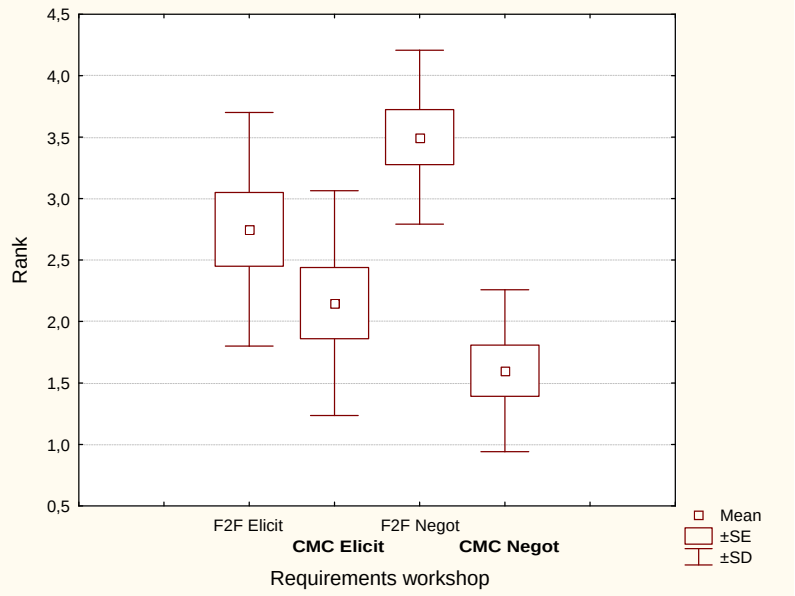


**Figure 49. Box plot of ranks computed on subjects' evaluation of consensus-based discussion and appropriate information generation pace in requirements workshops (the higher the rank, the better the workshop/medium fit)**

To further assess the differences between the ranks of the four workshop/medium fits, we applied a series of statistics to these scores to perform matched-pair comparisons between (I) F2F elicitation and F2F negotiation, (II) F2F elicitation and CMC elicitation, (III) F2F negotiation and CMC negotiation, and, finally, (IV) CMC

elicitation and CMC negotiation. The comparisons were performed by applying the Wilcoxon signed-rank test, as a nonparametric alternative to the t-test for two dependent samples [Con80]. We might have also executed the Friedman test for only two treatments, which is however, equivalent to performing the Sign test. Nonetheless, we preferred the Wilcoxon test, since it represents a more powerful alternative to the Sign test [ibid.].

The results, shown in Table 12, report for each matched-pair comparison (e.g. F2F elicitation vs. CMC elicitation), positive ranks (e.g., how many subjects preferred F2F excitation over the CMC counterpart), negative ranks (e.g., how many subjects preferred CMC elicitation over the F2F counterpart), and ties (e.g., how many subjects perceived F2F and CMC workshops to be equal).

**Table 12. Results of the Wilcoxon signed-rank test for matched-pairs comparisons (N=10)**

| | Matched-pair comparison<br>A vs. B | Positive ranks<br>A > B | Negative ranks<br>A < B | Ties<br>A = B | Wilcoxon signed-ranks test |
|---|---|---|---|---|---|
| I. | F2F elicitation vs. F2F negotiation | 0 | 6 | 4 | **Z=2.27** |
| II. | F2F elicitation vs. CMC elicitation | 5 | 2 | 3 | Z=1.56 |
| III. | F2F negotiation vs. CMC negotiation | 8 | 0 | 2 | **Z=2.54** |
| IV. | CMC elicitation vs. CMC negotiation | 4 | 3 | 3 | Z=.88 |

Results significant at the 5% level are shown in bold

The Wilcoxon test for the first pair (I) resulted significant at the 5% level (Z=2.27, p=.023), showing a significant preference of subjects for F2F negotiations over F2F elicitations. The second and third Wilcoxon tests show that, while subjects significantly prefer F2F negotiation over CMC negotiation (III, Z=2.54, p=.011), no statistically significant difference was found in the comparison between F2F elicitation and CMC elicitation (II, Z=1.56, p=.119). Finally, the comparison between CMC elicitation and CMC negotiation was not found statistically significant as well.

Given the results of Wilcoxon test and Friedman test, we can conclude that study subjects perceived F2F negotiations as the best-fitting task/technology match in terms

of the extent to which discussion was consensus-based and the information generated not missed.

#### 8.8.1.2. Results from Likert-type Scales Analysis: Comfort with Communication Mode

Besides satisfaction with performance, the other construct adopted for evaluating the requirements workshops is the subjects' perception of comfort with communication mode. Two 4-point Likert scales of 3 items (scale E-B) and 5 items (scale N-B), were used to compare the level of participation during elicitation and negotiation workshops., With respect to scale E-B defined in the post-elicitation questionnaire, we directly computed Chronbach's alpha coefficient to assess scale reliability, since it is made of only three items. The coefficient was .82, which is above the threshold of .70, suggested by Nunnally to affirm scale reliability [Nun78].

With respect to scale N-B defined in the post negotiation questionnaire, before computing the alpha index, we first performed principal component analysis to confirm that the five items were effectively measuring the construct of comfort with communication mode. The analysis, executed with varimax rotation and a cut-off point of .70, extracted two components: The first three items (N-B.1, N-B.2, and N-B.3) loaded on the first component, question N-B.4 loaded on the second one, and finally the question N-B.5 did not load on either component. Thus, we retained only the first component, which accounts for the 48% of the total variance. Cronbach alpha computed on the component extracted was .75. Henceforth, when ambiguity is not an issue, we will refer to the components extracted altogether (i.e., items N-B.1, N-B.2, and N-B.3), using the name of the whole scale (N-B). The descriptive statistics for the responses to the two Likert scales are presented in Table 13 and Table 14. We point out that three items of the component extracted from scale N-B coincide with the three items defined in the scale E-B.

**Table 13. Descriptive statistics for responses to scale E-B in the post-elicitation questionnaire**

| *In comparison to F2F elicitation, CMC elicitation…* | | **Valid N (clients / developers)** | **Mean** | **Median** | **Std. dev.** |
|---|---|---|---|---|---|
| **E-B.1** | *offered increased opportunity to participate in the discussion* | 17 (10 / 7) | 2.82 | 3 | 1.02 |
| **E-B.2** | *encouraged to more openly discuss conflicting issues with same group members* | 17 (10 / 7) | 2.77 | 3 | .753 |
| **E-B.3** | *encouraged to more openly discuss conflicting issues with other group members* | 17 (10 / 7) | 3.06 | 3 | .83 |

**Table 14. Descriptive statistics for responses to scale N-B in the post-negotiation questionnaire**

| *In comparison to F2F negotiation, CMC negotiation…* | | **Valid N (clients / developers)** | **Mean** | **Median** | **Std. dev.** |
|---|---|---|---|---|---|
| **N-B.1** | *offered increased opportunity to participate in the discussion* | 19 (11 / 8) | 2.53 | 3 | .77 |
| **N-B.2** | *encouraged to more openly discuss conflicting issues with members of the same group* | 19 (11 / 8) | 2.47 | 3 | .90 |
| **N-B.3** | *encouraged to more openly discuss conflicting issues with members of the other group* | 19 (11 / 8) | 2.53 | 3 | .61 |

In order to assess the statistical significance of subjects' level of agreement, we executed a chi-square goodness of fit test on both response sets. With regard to the elicitation workshops (see Table 15), the chi-square test results show that the subjects' moderate agreement with the fact that CMC elicitations encourage to more openly discuss conflicting issues with same and other group members (item 2 and 3, respectively) is significant at the 5% level ($\chi^2$=11.48, p=.009, and $\chi^2$=9.12, p=.028, respectively). With respect to the negotiation workshops (see Table 16), the chi-square test results show that subjects' moderate agreement with having increased opportunity to participate in the discussion and being encouraged to more openly discuss conflicting issues with same group members during CMC negotiations (item 1 and 3, respectively) is significant at the 5% level ($\chi^2$=10.68, p=.014, and $\chi^2$=8, p=.018, respectively).

**Table 15. Comfort with communication mode in CMC and F2F elicitations**

*In comparison to the F2F elicitation workshop, the CMC negotiation workshop…*

| | E-B.1 | | E-B.2: | | E-B.3: | |
|---|---|---|---|---|---|---|
| | **Valid N (clients/developers)** | **%** | **Valid N (clients/developers)** | **%** | **Valid N (clients/developers)** | **%** |
| **Strongly agree** | 5 (3 / 2) | 29.4 | 2 (1 / 1) | 11.8 | 5 (3 / 2) | 29.4 |
| **Somewhat agree** | 6 (3 / 3) | 35.3 | **10 (5 / 5)** | **58.8** | **9 (5 / 4)** | **52.9** |
| **Somewhat disagree** | 4 (3 / 1) | 23.5 | 4 (3 / 1) | 23.5 | 2 (1 / 1) | 11.8 |
| **Strongly disagree** | 2 (1 / 1) | 11.8 | 1 (1 / -) | 5.9 | 1 (1 / -) | 5.9 |
| **Chi-square $\chi^2$** | 2.06 | | **11.48** | | **9.2** | |

Significant results at the 5% level are shown in bold

**Table 16. Comfort with communication mode in CMC and F2F negotiations**

*In comparison to the F2F negotiation workshop, the CMC negotiation workshop…*

| | N-B.1: | | N-B.2: | | N-B.3: | |
|---|---|---|---|---|---|---|
| | **Valid N (clients/developers)** | **%** | **Valid N (clients/developers)** | **%** | **Valid N (clients/developers)** | **%** |
| **Strongly agree** | 1 (1 / -) | 5.3 | 2 (1 / 1) | 23.5 | - - | **-** |
| **Somewhat agree** | **10 (6 / 4)** | **52.6** | 8 (6 / 2) | 42.1 | **11 (7 / 4)** | **57.9** |
| **Somewhat disagree** | 6 (3 / 3) | 31.6 | 6 (3 / 3) | 31.5 | 7 (4 / 3) | 36.8 |
| **Strongly disagree** | 2 (1 / 1) | 10.5 | 3 (1 / 2) | 15.8 | 1 (- / 1) | 5.3 |
| **Chi-square $\chi^2$** | **10.68** | | 4,79 | | **8** | |

Significant results at the 5% level are shown in bold

In general, the results of the goodness of fit tests show the subjects tending to somewhat agree that, compared to F2F requirements workshops, in CMC elicitations

and negotiations they had increased opportunity to participate and more openly discussed about conflicting issues with the other participants. These statistics, however, compare F2F elicitation to CMC elicitation, and F2F negotiation to CMC negotiation through subjects' responses, regardless of the fact that they participated in either requirements workshop playing different roles. Hence, we performed two t-tests to verify whether being client or developer influenced subjects' perception of comfort with communication mode in both paired comparisons.

As a nonparametric alternative to t-test on independent samples, we applied the Mann-Whitney U test to both response sets [Con80]. The results reveal no significant differences in the responses given by clients and developers to the scales E-B and N-B (see Table 17). Hence, we can generally state that, whatever the role played, the subjects perceived an increased chance of participating in a more open discussion during CMC requirements elicitations and negotiations, as compared to F2F workshops.

**Table 17. Results of the Mann-Whitney U tests for differences between the responses given by clients and developers**

| | Role | N | Mean Rank | Mann-Whitney U test |
|---|---|---|---|---|
| **E-B.1** | Clients<br>Developers | 10<br>7 | 8,85<br>9,21 | U=33.5 Z=.15 p=.15 |
| **E-B.2** | Clients<br>Developers | 10<br>7 | 8,05<br>10,36 | U=25.5 Z=1.05 p=.36 |
| **E-B.3** | Clients<br>Developers | 10<br>7 | 8,85<br>9,21 | U=33.5 Z=.87 p=.87 |
| **N-B.1** | Clients<br>Developers | 11<br>8 | 10.73<br>9 | U=33 Z=.73 p=.47 |
| **N-B.2** | Clients<br>Developers | 11<br>8 | 11<br>8.63 | U=36 Z=.96 p=.34 |
| **N-B.3** | Clients<br>Developers | 11<br>8 | 10.73<br>9 | U=33 Z=.76 p=.45 |

Afterwards, we performed a matched-pair comparison to discover differences in the perceived comfort with communication mode between CMC elicitations and CMC negotiations. The purpose of this comparison was establishing whether the stakeholders perceived either type of CMC requirements workshop to provide a higher equality of participation and a more open discussion. For each of the ten subjects involved in both CMC workshop sessions, the responses were first summed to create the overall score on

both scale E-B and scale N-B (see Table 18). Then, being dependent samples, the two sets of overall scores were subjected to the Wilcoxon signed-rank test (see Table 19), which shows that in four cases, the overall score for comfort in CMC elicitation outweighs CMC negotiation (positive ranks, E-B > N-B). In other words, four subjects felt more comfortable with CMC during elicitation workshops than during the negotiation workshop. Only one subject felt more comfortable with CMC during the negotiation workshop than during the elicitation workshop (negative ranks, E-B < N-B), and the remaining five participants perceived no difference (ties, E-B = N-B). Since these results are not statistically significant (Z=1.63, p=.102), we can conclude that one type of CMC requirements workshop is not better then the other one at ensuring a more equal and open discussion, as compared to its own F2F alternative.

**Table 18. Summed scores of comfort with communication computed on E-B and N-B**

| Subject | E-B summed score | N-B summed score |
|---|---|---|
| S1 | 9 | 10 |
| S2 | 7 | 7 |
| S3 | 9 | 9 |
| S4 | 9 | 9 |
| S5 | 6 | 6 |
| S6 | 11 | 8 |
| S7 | 6 | 4 |
| S8 | 3 | 3 |
| S9 | 10 | 8 |
| S10 | 11 | 10 |
| Total | 81 | 74 |
| Mean | 8.1 | 7.4 |
| Median | 9 | 8 |
| Std. dev. | 2.56 | 2.41 |

**Table 19. Results of the Wilcoxon signed-rank test applied on the overall scores for E-B and N-B (N=10)**

| E-B summed score vs. N-B summed score | N | Mean rank | Sum of ranks |
|---|---|---|---|
| E-B > N-B. | 4 | 3.38 | 13.5 |
| E-B < N-B | 1 | 1.5 | 1.5 |
| E-B = N-B | 5 | | |
| Wilcoxon signed-rank test | Z=1.63, p=.102 | | |

### 8.8.1.3. Results from Multiple-choice Closed Questions Analysis

Besides the use of Likert-type scales, we also employed closed-ended questions with three choices. The three categorical (nominal) variables employed were *F2F*, *CMC*, and *About the same*. The closed questions aimed at capturing the subjects' preference for

either communication medium when performing some of the typical activities executed during requirements workshops. In the next two subsections we report the results from the analyses applied on closed questions, distinguishing between the two constructs of satisfaction with performance and comfort with communication mode.

#### 8.8.1.4. Results from Multiple-choice Closed Questions Analysis: Satisfaction with Performance

For the post-elicitation questionnaire, a set of fifteen items was defined (E-D). The first six questions (E-D.1 to E-D.6) are meant to capture subjects' preference in terms of satisfaction with performance.

The responses to these six questions are reported in Table 20. The total score computed for each categorical variable shows a general preference for CMC. Hence, in order to statistically assess the significance of subjects' preferences, responses were subjected to the chi-square goodness of fit test. The results from the chi-square test show that the subjects significantly prefer CMC over F2F for having a structured discussion (E-D.2, $\chi^2$=14.59, p=.001) and documenting the decisions made (E-D.3, $\chi^2$=14.59, p=.001) during the elicitations. Conversely, the subjects prefer F2F conversation over CMC for having an in-depth discussion (E-D.5, $\chi^2$=10.71, p=.001). It is also interesting to note that the stakeholders did not perceive any difference between the media with respect to consensus-based decisions.

**Table 20. Results of the goodness of fit test on satisfaction with performance in requirements elicitation workshops (N=17)**

| *Which medium better supported…* | | F2F | CMC | About the same | Chi-square $\chi^2$ |
|---|---|---|---|---|---|
| **E-D.1** | *consensus-based decisions* | 7 | 6 | 4 | .82 |
| **E-D.2** | *structured discussion* | 1 | **13** | 3 | **14.59** |
| **E-D.3** | *documentation of decisions made* | 1 | **13** | 3 | **14.59** |
| **E-D.4** | *workshop facilitation* | 0 | 12 | 5 | 2.88 |
| **E-D.5** | *in-depth discussion* | **12** | 2 | 3 | **10.71** |
| **E-D.6** | *visibility of decisions made* | - | 12 | 5 | 2.88 |
| | Total | 21 | 58 | 23 | |

Significant results ate the 5% level are shown in bold

The post-negotiation questionnaire contained a larger set of twenty-two multiple-choice closed question (N-D), which includes the same items contained in the set E-D of the post-elicitation questionnaire, plus other specific items.

Table 21 shows the results of the goodness of fit test applied on the first twelve closed questions (N-D.1 to N-D.12), which evaluate subjects' preference for either medium in terms of satisfaction with performance during requirements negotiations. While the total scores indicate no general preference for either medium during the negotiation workshops, the stakeholders significantly prefer F2F interaction to have an in-depth discussion (N-D.5, $\chi^2$=8.32, p=.003) and CMC to document the decisions made (N-D.3, $\chi^2$=11.79, p=.003).

These results are similar to the stakeholders' preferences found significant for requirements elicitation workshop. Beside, the stakeholders also found that CMC better supports facilitation of the workshop (N-D.4, $\chi^2$=8.9, p=.016) and visibility of the decisions made (N-D.6, $\chi^2$=11.84, p=.001).

**Table 21. Results of the goodness of fit test on satisfaction with performance in requirements negotiation workshops (N=19)**

| *Which medium better supported…* | | F2F | CMC | About the same | Chi-square $\chi^2$ |
|---|---|---|---|---|---|
| **N-D.1** | *consensus-based decisions* | 5 | 7 | 7 | .42 |
| **N-D.2** | *structured discussion* | 3 | 11 | 5 | 5.47 |
| **N-D.3** | *documentation of decisions made* | 1 | **13** | 5 | **11.79** |
| **N-D.4** | *workshop facilitation* | - | **16** | 3 | **8.9** |
| **N-D.5** | *in-depth discussion* | **12** | 2 | 5 | **8.32** |
| **N-D.6** | *visibility of decisions made* | - | **17** | 2 | **11.84** |
| **N-D.7** | *keeping participants on task* | 10 | 6 | 3 | 3.9 |
| **N-D.8** | *reaching an agreement* | 7 | 4 | 8 | 1.37 |
| **N-D.9** | *prioritizing requirements* | 5 | 5 | 9 | 1.68 |
| **N-D.10** | *resolving conflicts* | 10 | - | 9 | .05 |
| **N-D.11** | *drawing conclusions* | 10 | 5 | 4 | 3.26 |
| **N-D.12** | *getting all the work done* | 8 | 2 | 9 | 4.53 |
| | Total | 71 | 88 | 69 | |

Significant results at 5% level are shown in bold

Again, communication mode did not affect the extent to which decisions were made with a consensus during the negotiations. In addition, it is important to point out that many of the activities relevant to requirements negotiations, such as reaching an agreement prioritizing, requirements, resolving conflicts, and drawing conclusions, are not significantly affected by the medium used to conduct the workshop.

#### 8.8.1.5. Results from Multiple-choice Closed Questions Analysis: Comfort with Communication Mode

The nine remaining items in group E-D are meant to further assess the degree of comfort with the two communications mode in requirements elicitation workshops. Table 22 shows subjects responses to questions E-D.7 to E-D.9. In this case, the total scores show a general, strong preference for F2F. In fact, the goodness of fit test results shows that, for the elicitation workshops, the stakeholders significantly prefer F2F communication mode to facilitate familiarization with other participants (E-D.12, $\chi^2$=13.24, p=.000), and better support the ability to express complex ideas (E-D.10, $\chi^2$=23.06, p=.000) and understand others' opinions (E-D.13, $\chi^2$=7.12, p=.000). Also, the chi-square test results reveal no significant preference for CMC elicitations in terms of comfort with communication mode. Despite the general preference for F2F, we can also note that no differences were perceived by the students in terms of the sense of involvement and motivation to participate in workshops.

**Table 22. Results of the goodness of fit test on comfort with communication mode in requirements elicitation workshops (N=17)**

| *Which medium better supported…* | | F2F | CMC | About the same | Chi-square $\chi^2$ |
|---|---|---|---|---|---|
| **E-D.7** | *articulation of ideas freely* | 7 | 5 | 5 | .47 |
| **E-D.8** | *spontaneous discussion* | 10 | 3 | 4 | 5.06 |
| **E-D.9** | *ability to express basic ideas* | 7 | 4 | 6 | .82 |
| **E-D.10** | *ability to express complex ideas* | **15** | 1 | 1 | **23.06** |
| **E-D.11** | *control of etiquette and professionalism* | 7 | 5 | 5 | .47 |
| **E-D.12** | *gaining familiarity with participants* | **16** | - | 1 | **13.24** |
| **E-D.13** | *ability to understand participants' opinions* | **14** | 0 | 3 | **7.12** |
| **E-D.14** | *sense of involvement in the workshop* | 6 | 2 | 9 | 4.35 |
| **E-D.15** | *motivation to participate in discussions* | 4 | 3 | 10 | 5.06 |
| | Total | 86 | 23 | 44 | |

Significant results at the 5% level are shown in bold

Finally, Table 23 shows the results of the goodness of fit test applied on the responses to questions N-D.13 to N-D.22, which evaluate subjects' preference in terms of comfort with communication mode during requirements negotiations. Similarly to the case of the post-elicitation questionnaire, the total scores show a strong preference for F2F communication and the goodness of fit test shows again no statistically significant preference in favor of CMC. The stakeholders significantly prefer F2F interaction to better express complex ideas (N-D.16, $\chi^2$=9.58, p=.008), understand others' opinions (N-D.19, $\chi^2$=11.79, p=.001), familiarize with workshop participants (N-D.18, $\chi^2$=18.11, p=.000), and control professionalism (N-D.17, $\chi^2$=11.42, p=.002).

**Table 23. Results of the goodness of fit test on comfort with communication mode in requirements negotiation workshops (N=19)**

| *Which medium better supported…* | | F2F | CMC | About the same | Chi-square $\chi^2$ |
|---|---|---|---|---|---|
| **N-D.13** | *articulation of ideas freely* | 6 | 5 | 8 | .74 |
| **N-D.14** | *spontaneous discussion* | 6 | 5 | 8 | .74 |
| **N-D.15** | *ability to express basic ideas* | 7 | 2 | 10 | 5.16 |
| **N-D.16** | *ability to express complex ideas* | **12** | 1 | 6 | **9.58** |
| **N-D.17** | *control of etiquette and professionalism* | **17** | - | 2 | **11.84** |
| **N-D.18** | *gaining familiarity with participants* | **15** | 1 | 3 | **18.11** |
| **N-D.19** | *ability to understand participants' opinions* | **13** | 1 | 5 | **11.79** |
| **N-D.20** | *sense of involvement in the workshop* | 8 | 5 | 6 | .74 |
| **N-D.21** | *motivation to participate in discussions* | 6 | 5 | 8 | .74 |
| **N-D.22** | *look up relevant info from existing documents* | 6 | 8 | 5 | .74 |
| **N-D.23** | *avoiding somebody to dominate discussion* | 2 | 10 | 7 | 5.16 |
| | Total | 98 | 43 | 68 | |

Significant results at the 5% level are shown in bold

Similarly to the results obtained from the post-elicitation questionnaire, also in this case the communication mode did not significantly influence the sense of involvement and motivation to participate in negotiation workshops.

### 8.8.2. Tool Support Evaluation for Synchronous, Text-Based Elicitations and Negotiations

To evaluate the differences between F2F and CMC requirements workshop, we defined two identical 8-item, 4-point Likert scales in the post-elicitation and post-negotiation questionnaires (scales E-A and N-A, respectively) to measure the support provided by the eConference tool features.

With regard to the analysis of scale E-A, we performed an exploratory component analysis on the responses to the scale E-A to extract the latent construct(s) measured by the questions. The principal component analysis with varimax rotation and a cut-off of .70 extracted three components. However, scale reliability can be only computed when components have at least three items. Hence, we retained only one component, containing the three items E-A.1, E-A.3, and E-A.3 (see Table 24), which account for the 37% of the total variance. Scale reliability was assessed by calculating coefficient alpha (.78). The three items in the extracted component consistently measure the tool support for an effective discussion during CMC elicitations.

Table 25 shows the results of the chi-square goodness of fit test, applied to the extracted component, all of which reveal significant differences between the responses. The test showed that the significant majority of the stakeholders strongly agree that having complete logs of discussions and decisions made helps increasing elicitation effectiveness (E-A.1, $\chi^2$=10.9, p=.004), and that the decisions place content helps reaching a consensus faster (E-A.3, $\chi^2$= 12.79, p=.019). The subjects' moderate agreement with the fact that the decisions place content helps clarifying ambiguities is also statistically significant (E-A.3, $\chi^2$= 12.79, p=.005).

**Table 24. Descriptive statistics for the component 'discussion effectiveness in CMC elicitations,' extracted from scale E-A (N=20)**

| | | Valid N (clients/developers) | Mean | Median | Std. dev. |
|---|---|---|---|---|---|
| **E-A.1** | *Having a complete log of both chat and decisions place at the end of the workshop improved the effectiveness of the elicitation* | 20 (13 / 7) | 3,60 | 4 | 0,60 |
| **E-A.2** | *Viewing the decisions place content being edited by the Scribe during the elicitation was useful in reaching a consensus faster* | 20 (13 / 7) | 3,25 | 3 | 0,85 |
| **E-A.3** | *Viewing the decisions place content being edited by the Scribe during the elicitation was useful in clarifying ambiguities* | 19 (12/ 7) | 3,05 | 3 | 0,78 |

**Table 25. Results for the goodness of fit test applied to component 'discussion effectiveness in CMC elicitations'**

| | E-A.1 | | E-A.2 | | E-A.3 | |
|---|---|---|---|---|---|---|
| | **Valid N (clients/developers)** | **%** | **Valid N (clients/developers)** | **%** | **Valid N (clients/developers)** | **%** |
| **Strongly agree** | **13 (9 / 4)** | **65** | **9 (5 / 4)** | **45** | 5 (4 / 1) | 26.3 |
| **Somewhat agree** | 6 (3 / 3) | 30 | 8 (6 / 2) | 40 | **11 (6 / 5)** | **57.9** |
| **Somewhat disagree** | 1 (1 / -) | 5 | 2 (1 / 1) | 10 | 2 (1 / 1) | 10.5 |
| **Strongly disagree** | - - | - | 1 (1 / -) | 5 | 1 (1 / -) | 5.3 |
| **Chi-square $\chi^2$** | **10.9** | | **10** | | **12.79** | |

Significant results at the 5% level are shown in bold

As stakeholders, the participants in the CMC elicitations played different roles. Hence, we performed a Mann-Whitney U test to verify whether being client or developer influenced subjects' perception of tool usefulness during CMC elicitations. The test results show no significant difference in the responses given by clients and developers (see Table 26).

**Table 26. Results from Mann-Whitney U test applied to component 'discussion effectiveness in CMC elicitations'**

| | Role | Valid N | Mean rank | Mann-Whitney U test |
|---|---|---|---|---|
| **E-A.1** | developers | 7 | 9.93 | U=41.5 Z=-.38 p=.705 |
| | clients | 13 | 10.81 | |
| **E-A.2** | developers | 7 | 11.64 | U=37.5 Z=-.68 p=.491 |
| | clients | 13 | 9.88 | |
| **E-A.3** | developers | 7 | 9.21 | U=36.5 Z=-.52 p=.601 |
| | clients | 12 | 10.46 | |

The scale N-A in the post-negotiation questionnaire was subjected to the same statistics applied to the scale E-A. The principal component analysis with varimax

rotation and a cut-off of .70 extracted three components, but, again, only one component with three items was retained (i.e., N-A.2, N-A.3, and N-A.4). The retained component accounts for the 43% of the total variance and coefficient alpha of scale reliability is .84. This component measures the same construct of tool support for an effective discussion, this time, during requirements negotiations.

**Table 27. Descriptive statistics for component 'discussion effectiveness in CMC negotiations' extracted from scale N-A (N=19)**

| | | Valid N (clients/developers) | Mean | Median | Std. dev. |
|---|---|---|---|---|---|
| **N-A.2** | *Viewing the decisions place content being edited by the Scribe during the negotiation was useful in reaching a consensus faster* | 18 (10 / 8) | 3,00 | 3 | 1,08 |
| **N-A.3** | *Viewing the decisions place content being edited by the Scribe during the negotiation was useful in clarifying ambiguities* | 17 (8 / 9) | 3,05 | 3 | 0,83 |
| **N-A.4** | *The item-based discussion was useful in keeping the negotiation workshop on track* | 16 (7 / 9) | 2,69 | 3 | 0,87 |

Table 28 shows the breakdown of subjects' responses to the three questions. The goodness of fit test results reveals that only the moderate agreement with the usefulness of viewing the decisions made to clarify ambiguities was statistically significant at the 5%level (N-A.3, $\chi^2$= 9.1, p=.028).

**Table 28. Results for the goodness of fit test applied to component 'discussion effectiveness in CMC negotiations'**

| | N-A.2 | | N-A.3 | | N-A.4 | |
|---|---|---|---|---|---|---|
| | **Valid N (clients/developers)** | **%** | **Valid N (clients/developers)** | **%** | **Valid N (clients/developers)** | **%** |
| **Strongly agree** | 7 (5 / 2) | 38,9% | 5 (2 / 3) | 29.4 | 7 (6 / 1) | 38.9 |
| **Somewhat agree** | 7 (4 / 3) | 38,9% | **9 (6 / 3)** | **52.9** | 8 (3 / 5) | 44.4 |
| **Somewhat disagree** | 1 (- / 1) | 5.6 | 2 (1 / 1) | 11.8 | 3 (1 / 2) | 16.7 |
| **Strongly disagree** | 3 (1 / 2) | 16.7 | 1 (- / 1) | 5.9 | - - | - |
| **Chi-square $\chi^2$** | 6 | | **9.12** | | 4.5 | |

Significant results at the 5% level are shown in bold

These responses were subjected to the Mann-Whitney U test to assess the influence of the role played on the subjects' perception of tool support during CMC negotiations. The results are shown in Table 29 and reveal no statistically significant influence of the role on subjects' responses.

**Table 29. Results from Mann-Whitney U test applied to component 'discussion effectiveness in CMC negotiations'**

| | **Role** | **Valid N** | **Mean rank** | **Mann-Whitney U test** |
|---|---|---|---|---|
| **N-A.2** | developers | 8 | 7,75 | U=26.0 Z=-1.32 p=.185 |
| | clients | 10 | 10,90 | |
| **N-A.3** | developers | 8 | 9,06 | U=35.5 Z=-.05 p=.958 |
| | clients | 9 | 8,94 | |
| **N-A.4** | developers | 7 | 8,07 | U=28.5 Z=-.33 p=.737 |
| | clients | 9 | 8,83 | |

Finally, to investigate the differences between CMC elicitations and CMC negotiations, for each of the participants who returned both post-hoc questionnaires, we computed the summed score was of the valid responses to the items in common

between the two components extracted from scales E-A and N-A (namely E-A2, EA.3 and N-A.2, N-A.3). These overall scores (see Table 30) measure the level of tool support provided by the decisions place feature to participants during CMC elicitations and CMC negotiations. A matched-pair comparison was then performed applying the Wilcoxon signed-rank test. The test results are reported in Table 31 and indicate that three subjects perceived decisions place support to be better in the CMC elicitation than in CMC negotiation workshop (CMC elicitation > CMC negotiation). In contrast, two subjects indicated a lower support during the CMC elicitation workshop (CMC elicitation < CMC negotiation). Finally, three subjects perceived no differences in the support provided (CMC elicitation = CMC negotiation). The signed rank test result are not significant (Z=.71, p=.48). We also applied the Sign test to the same data and found again no statistical difference. Hence, we conclude that there is no statistically significant difference support between the perceived support provided by the decisions place feature during CMC elicitations and CMC negotiations.

**Table 30. Summed scores computed on the items in common between the two components extracted from scales E-A and N-A**

| Subjects | CMC elicit summed score (E-A.2+E-A.3) | CMC negot summed score (N-A.2+N-A.3) |
|---|---|---|
| S1 | 2 | 2 |
| S2 | 7 | 6 |
| S3 | 8 | 8 |
| S4 | 6 | 8 |
| S5 | 6 | 6 |
| S6 | 4 | 5 |
| S7 | 6 | 7 |
| S8 | 7 | 6 |
| Total | 46 | 48 |
| Mean | 5.75 | 6 |
| Median | 6 | 6 |
| Std. dev. | 1.91 | 1.93 |

**Table 31. Results of Wilcoxon signed rank test performed to evaluate differences in the support provided by decisions place to CMC workshops**

| CMC elicitation Vs. CMC negotiation | N | Mean rank | Sum of ranks |
|---|---|---|---|
| CMC elicit > CMC negot | 3 | 3.33 | 10 |
| CMC elicit < CMC negot | 2 | 2.50 | 5 |
| CMC elicit = CMC negot | 3 | | |
| Wilcoxon signed-rank test | Z=.71, p=.48 | | |

## 8.9. Discussion

In this study we compared the use of F2F and synchronous, text-based communication (CMC) for supporting ad hoc groups of stakeholders involved in distributed requirements workshops. We investigated two research questions.

### 8.9.1. $RQ_1$

Research question $RQ_1$ was intended to investigate whether the leanness of text-based media significantly impairs the requirements workshop with respect to the stakeholders' perceptions of performance and the comfort with the communication mode.

To answer $RQ_1$, in comparing F2F workshops to CMC workshops, we evaluated the levels of *comfort with communication mode* and *satisfaction with performance* perceived by stakeholders. Because the role factor was not found to significantly affect our findings, it is ignored in this discussion. Table 32 summarizes, for the sake of readability, the statistically significant differences found in the analysis of subjects' perceptions, with respect to the two high level constructs.

**Table 32. Statistically significant differences found in the subjects' perceptions when comparing F2F and CMC requirements workshops**

| | Satisfaction with performance | Comfort with communication mode |
|---|---|---|
| **F2F** | * more in-depth discussion | * higher comfort with spoken language<br>* better for gaining familiarity, express and understand complex ideas, control professionalism |
| **CMC** | * more structured discussion<br>* decisions made more visible and better documented | * increased opportunity to participate in more open discussions with same and other group members |

With regard to the level of communication comfort perceived by subjects, our findings confirmed in general the predictions of both media richness theories, and socio-psychological theories on the media effects. On the one hand, the prominent theories of Media Richness and Social Presence (see Section 4.2 and 4.3) have already

acknowledged the general individual preference for rich interaction, regardless of any context. In line with these predictions, we found that the subjects perceived a higher comfort with F2F interaction during the requirements workshops (see Table 22 and Table 23). In addition, the subjects perceived F2F interaction to be significantly better than CMC communication at expressing complex ideas and understanding the others' opinions, thus confirming that, as tasks are thought to be more complex, individuals tend to prefer rich interaction. Further, F2F workshops were significantly more useful in terms of the possibility to gain familiarity with other participants. This is a typical problem of newly-formed, distributed groups in general and hence, also of ad hoc groups of stakeholders.

Altogether, these findings show that individuals' perception of comfort with communication mode in task/technology fits depends more on channel richness characteristics *per se*, rather than on the appropriateness of match with task characteristics. It is also interesting to note that the communication mode factor did not affect stakeholders' motivation to participate and the sense of involvement in the requirements workshops, i.e., no statistically significant differences resulted from the analyses executed, showing that subjects had similar level of commitments in the execution of both F2F and CMC tasks.

With regard to the level of satisfaction with performance, compared to CMC workshops, the subjects were more satisfied with F2F workshops performance in terms of depth of discussion (see Table 20 and Table 21). An explanation for the lack of depth in CMC discussion is again provided by Media Synchronicity theory. The slower the interaction pace of text-based communication strengthened the time constraint imposed on the stakeholders, thus causing a shallower discussion during the CMC workshops. Conversely, CMC elicitations and CMC negotiation were preferred over F2F workshops to ensure better documentation and visibility of decisions made. Besides, the subjects significantly perceived to have increased opportunity to participate and more openly discuss conflicting issues with other participants during CMC workshops than during F2F workshops (see Table 15 and Table 16). Finally, they were also significantly more satisfied with the structured discussion they had during CMC workshops. These benefits are somewhat expected because they are intrinsic to the usage of the agenda

and decisions place features available in the tool employed. This is further confirmed by results presented in Section 8.8.2 (see Table 25 and Table 28), which show that the decisions place editor and the availability of the complete logs provided the most useful support to workshop effectiveness.

These results confirm the predictions of the socio-psychological theories, which argue that the depersonalization effect, induced by the use of less-rich and less-social media, limits domination, group/social pressure, and the other dysfunctional aspects intrinsic to F2F group communication (see Section 4.6).

### 8.9.2. $RQ_2$

Research question $RQ_2$ aimed to understand whether both CMC requirements elicitations and CMC requirements negotiations represent an appropriate task/technology fit.

To answer $RQ_2$, we again distinguish our findings with respect to the levels of *comfort with communication mode* and *satisfaction with performance* perceived by stakeholders. The role factors is once more ignored in the discussion of our findings because it was not found to have a significant impact.

GSS research has shown that groups interacting on text-based channels have often outperformed collocated groups in task of idea generation because of the possibility to input ideas in parallel. Conversely, collocated groups have usually outperformed distributed groups in executing tasks that involve problem solving, decision making, and conflict resolution [Mur00]. Neither the use of rich media, like video or F2F communication, has been shown to positively affect the performance quality of the work when it involves negotiation [Fin97, Ols97]. Thus, consistently with these findings, we expect that synchronous, text-based elicitation represents a better task/technology fit than synchronous, text-based negotiation. The box plot in Figure 49 shows a large and statistically significant difference between subjects' satisfaction with performance during F2F and CMC negotiations, perceived as the best and worst fit, respectively. In contrast, the difference between F2F and CMC elicitation is not statistically significant. These results suggest, on the one hand, that in terms of satisfaction with performance CMC elicitation is a better task/technology fit than CMC

negotiation and, on the other hand, that the general preference for F2F requirements workshops is due to the strong preference for the F2F negotiation fit over the CMC counterpart (see Table 12). Nevertheless, Table 21 shows that many of the activities relevant to requirements negotiations, such as reaching an agreement prioritizing, requirements, resolving conflicts, and drawing conclusions, were perceived not to be significantly impaired by the use of text-based communication. Hence, further analyses are needed to provide a more thorough answer, since these results only address stakeholders' perception of the workshop process performance.

With respect to the level of comfort with communication mode, a very few differences resulted in the comparison between CMC elicitations and negotiations. One statistically difference worth of mentioning that we found is that, during negotiations, subjects perceived F2F to be better than CMC in controlling the professionalism of participants. This problem was perceived during the negotiations probably because the number of participants and sub activities to execute was higher than during elicitations. In general, one explanation for this issue is suggested by Media Synchronicity theory. Due to the lower synchronicity level of text-based channels, and the consequent slower interaction in text-based communication, during CMC workshops the stakeholders perceive a higher time-constraint pressure and hence, are less permissive towards any non-task-focused activity. Mora data in support of this explanation may probably be found applying content analysis to the discussion logs of the CMC workshops. Nevertheless, previous research has already acknowledged that time constraint (i.e., the difference between the amount of time available and the amount of time required for task accomplishment) and the number of tasks to be executed significantly impair performance [Sve93, Mau97].

One limitation of this study is that we evaluated the effects of group, task, and media interactions (i.e., the context) only on the group process and through self-reported data. Instead, the causal model presented in Chapter 5 showed that the context has an effect on the outcome of the group process (i.e., the requirements workshop in this case). Therefore, further evidence not related to subjects' perceived satisfaction can be obtained analyzing the differences in the requirements workshops outcomes, i.e., the specification documents RS 1.0 and RS 2.0, produced by stakeholders as a result of the

requirements elicitations and negotiations (see Figure 50). The *requirements workshop* and *communication mode* factors created two variants in the iterative process used to produce the requirements specification document (see Figure 46). In our study three of the six projects were completed following the first process variant, which includes CMC elicitations and F2F negotiation workshops. Conversely, the remaining three projects were completed following the second process variant, which includes instead, F2F elicitation and CMC negotiation workshops. Hence, to confirm that CMC elicitation represents a better task/technology fit, the results from the analyses of the specification documents are expected to show that (1) no significant difference is found with respect to the quality of the RS 1.0 (e.g., completeness, correctness, clarity), whatever the process variant utilized; (2) the RS 2.0 documents created following the first process variant (CMC elicitation and F2F negotiations) are better than those created following the second process variant (F2F elicitation and CMC negotiation).

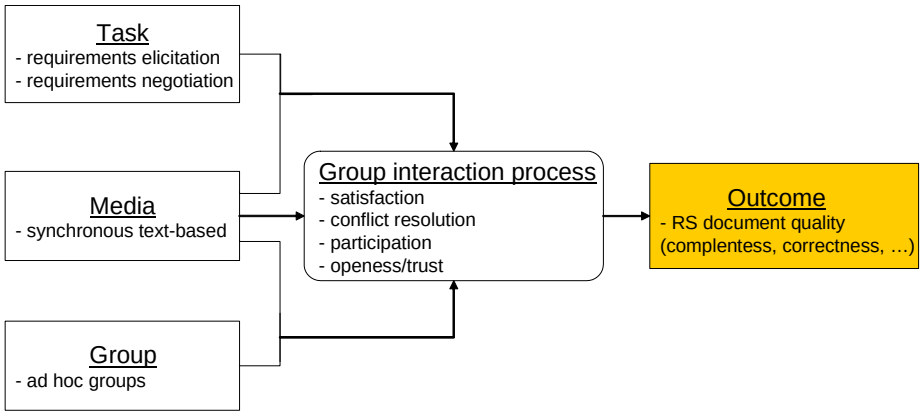


**Figure 50. The effects of task, media, and group characteristics on requirements workshops can be evaluated through the quality of the requirements specification documents produced as an outcome**

## 8.10. Threats to Validity

One of the key issues in experimentation is evaluating the validity of results. In this section the validity of the findings is discussed with respect to the threats that are relevant for our experiment.

### 8.10.1. Internal Validity

Threats to internal validity influence the conclusion about a possible causal relationship between the treatment and outcome of a study [Woh00]. The following rival explanations for the findings have been identified.

**Instrumentation.** The threat of instrumentation deals with the differences in the results that may be caused by differences in experimental material. As we evaluated the interaction between stakeholders who defined the software requirements for six different applications, it cannot be excluded that the differences in the application domain and complexity have influenced our study as confounding factors.

**Motivation.** During the execution of experiments, subjects may react differently over time. Since our experiment was performed during a considerable part of the whole course duration (about 10 weeks out of 16), boredom and tiredness effects cannot be disregarded and may partially explain the lower return-rate of the post-negotiation (second) questionnaire. However, since the subjects were graded on the overall outcome of the requirements definition process (i.e., the RS 2.0), they were motivated to keep a deep commitment to the tasks.

### 8.10.2. External Validity

External validity is not concerned with the validity of the specific study. Instead, it describes the study representativeness and the ability to generalize the results outside the scope of the study. We identified the following threats to external validity.

**Generalizability of subjects.** For any academic laboratory experiment, the ability to generalize the results to industry practice is restricted by the employment of students as study participants. While the students may not be representative of the entire population of software professionals, it has been shown that the differences between students and real developers may not be as large as assumed by previous research [Hös00]. Another issue with the representativeness of subjects is related to their familiarity with the use of synchronous, text-based communication. Computer science students are very accustomed with text-based interaction. They reported in the questionnaires to use IM and chat on a daily basis or almost daily basis. Nevertheless, these synchronous, text-

based communication tools are increasingly being adopted in the workplace, not only in the field of software development, to complement email [Mul03].

**Generalizability of settings.** Another threat to generalizeability is the simulation of the geographical dispersion. The subjects in the study were not actually dispersed. Instead, the members of each team were dispersed in the three laboratory used during the CMC requirements workshop sessions. This threat was mitigated to some extent by the strict control asserted over the students in order to prevent them from interacting verbally throughout the workshops. During the CMC elicitation session, one of the students involved could not come to the laboratory and thus, decided to join from home. Later, we informally interviewed him and he reported no technical difficulty due to the fact that he was accessing from his home PC, and no difference compared to the CMC negotiation workshop that he participated in, regularly accessing from the laboratory. Nevertheless, we could prevent rich interaction between the subjects only during the CMC sessions, while it cannot be excluded that the students had follow-up F2F discussions after.

### 8.10.3. Construct Validity

Construct validity concerns the degree of accuracy to which the variables defined in the study measure the constructs of interests [Woh00]. We identified a couple of threats to construct validity.

**Appropriateness of measures.** The constructs of *satisfaction with performance* and *comfort with communication mode*, selected as criteria for comparing CMC workshops to F2F workshops, have been adapted from a similar study on media effects by Murthy & Kerr [Mur00]. The several questions used to measure these constructs were defined by the researcher, taking into account (1) the communication issues commonly experienced and already acknowledged by previous research in requirements engineering (see Section 2.3), and (2) the issues informally reported by the students (see Section 8.6). While one could argue about the arbitrariness in the definition of the questions to measure the two constructs, in our study this issue has been overcome by executing the principal component and scale reliability analyses, which respectively assess the cohesiveness of questions in the scales, and the extent to which the responses

to questions can be treated as measuring a single latent variable. While one could argue about the arbitrariness in the definition of the scales use to operationalize each construct, this issue has been overcome in the study by executing principal component analysis and scale reliability analysis.

**Self-reported data.** Our measures of the constructs are taken from self-reported data. However, subjects' preference for the communication mode is not always aligned with actual performance gaining, as shown by GSS-related research. In our study this drawback is mitigated by having the subjects express their media preference not for hypothetical situations, but upon the accomplishment of realistic experimental tasks (i.e., the requirements workshops).

## 8.11. Conclusions

In this chapter, we have presented an experiment conducted at the University of Victoria on the effects of synchronous, text-based communication in distributed requirements workshops. In particular, we analyzed the differences between F2F and text-based communication in terms of satisfaction with performance and comfort with communication mode, as perceived by stakeholders during both elicitation and negotiation workshops.

Differently from many other experiments on media effects, this study did not use generic, puzzles-like tasks that involve either idea generation or problem solving. Instead, the experimental tasks were elicitations and negotiation of software requirements for non-toy-sized, realistic systems. In addition, the participants needed to recall specialized knowledge (e.g., the RFP during the elicitation workshops, the RS 1.0 during the negotiations) and techniques learned through the course (e.g., meeting facilitation), to effectively accomplish the tasks. This resulted in a higher cognitive load for the study participants and an increased, more realistic effort required for accomplishing the experimental tasks [Mur00].

The findings from the first analyses of the experimental data have confirmed the results of previous research, showing that subjects perceived a higher level of comfort with F2F communication mode, although CMC has not lowered the motivation to

participate and the sense of involvement. In addition, the study findings have suggested, on the one hand, that CMC elicitation is a better task/technology fit than CMC negotiation in terms of satisfaction with performance, and, on the other hand, that the general preference for F2F over CMC is due to the strong preference for the F2F negotiation fit over the CMC counterpart. These findings resulted from the analysis performed on the post-hoc satisfaction questionnaires administered to the subjects after the requirements workshops and hence, they specifically address how stakeholders perceived the workshop process performance. Nevertheless, in order to accurately assess the effectiveness of using a synchronous, text-based communication channel for conducting requirements workshops, we need to perform further analyses on data other than those self-reported on satisfaction questionnaires. Indeed, the large body of knowledge about media richness has proved that asking directly about both media preferences and media effectiveness is deeply affected from the perceived richness and social presence of the media themselves, regardless of the type of task.

## 8.12. Summary

This chapter has presented the empirical study to gain insights on the use of synchronous text-based communication for supporting ad hoc teams of stakeholders in conducting distributed requirements elicitation and negotiations. The findings from the study have confirmed the predictions of the theoretical framework discussed in Chapter 5, showing that, during the requirements meetings, the subjects perceived a higher level of comfort with F2F communication mode than with CMC, while keeping an equal level of motivation to participate. Nevertheless, the findings have also shown that: (1) compared to F2F requirements workshops, synchronous text-based workshops grant a higher opportunity to participate in a more structured, equal, and open discussion; (2) stakeholders are more satisfied with performance in synchronous, text-based elicitations than in synchronous text-based negotiations.

Overall, these results suggest to distributed teams of stakeholders that synchronous text-based elicitations represent a better task-technology fit than synchronous text-based

negotiations, for reducing the negative effects of distance, as well as the need and the number of collocated requirements workshops.

# Chapter 9:

# CONCLUSIONS

## 9.1. Contributions

No previous study had compared face-to-face (F2F) to synchronous text-based media for accomplishing a complex task such as conducting requirements workshops. Although many research works in the field of Requirements Engineering found mixed results in supporting that video communication is as effective as F2F, ineffectiveness of text-based communication has been given for granted because many prominent theories on computer-mediated communication had predicted so. However, we showed that also the complex theoretical background is rather equivocal. On the one hand, the theories of media richness posit that the more complex the task, the richer the medium to adopt. On the other hand, however, socio-psychological and cognitive theories postulate that the depersonalization effect imposed by lean media can be beneficial for reducing both the information overload and the emotional side-effects, like domination and social consensus pressure observed with rich media, thus increasing the meeting effectiveness in group communication. One contribution of this research effort was to fill this gap by conducting an empirical investigation, in a well-defined context, to evaluate the support of synchronous text-based communication for conducting distributed requirements elicitation and negotiations, involving ad hoc groups of stakeholders.

Given the somewhat exploratory nature of our research, we reviewed the complex and vast theoretical background on computer-mediated communication (CMC). A

second contribution of this research was the combination of the very many, and apparently conflicting, CMC theories in a fully-comprehensive and consistent framework on media effects, useful to identify the positive and negative forces that constraint the selection process of appropriate task/technology fits. The framework was then used for discussing the findings from the empirical investigation, with respect to the predictions posited by these theories. The findings confirmed that subjects perceive a higher level of comfort with face-to-face communication mode. Nevertheless, they also suggested that synchronous text-based elicitation represents a better task/technology fit than synchronous text-based negotiation, for reducing the need of collocated requirements workshops. Consequently, the use of lightweight text-based communication tools can increase the opportunities of interaction at a distance, instead of solely relying on a few and very hard to organize F2F meetings. On the one hand the stakeholders may increase the number of effective requirements workshops by organizing lightweight and distributed follow-up meetings. On the other hand, they may reduce the need for collocated meetings by conducting distributed requirements elicitation and reserve the use of F2F interaction for requirements negotiation only.

This research also contributed to the study of ad hoc groups, for which a new definition was proposed (i.e., small- to medium-sized teams, highly dynamic in creation, participation, and release, with no past and future of collaborations, whose temporal scope corresponds exactly to the time needed to carry out the collaboration in attendance). Previous research had studied such teams by running experiments that took into account the execution of a single task. In contrast, 'natural' collaboration involving ad hoc groups do not usually end with the execution of a single task. Real world collaborations are complex to the point of being divided into several distinct activities. Therefore, in our experiment we studied ad hoc groups while performing a cognitively-complex task (i.e., conducting requirements workshops) within a larger collaboration activity (i.e., the definition of software requirements for a given system), which defined the groups' temporal scope. Hence, we argue that our experimental results are better generalizable because our experimental setting better represents the conditions of natural, ad-hoc group collaborations. Given the temporary nature of such groups, they need a communication infrastructure with costs kept at minimum. Thus, we developed

eConference, a tool designed for supporting ad hoc groups with both structured and unstructured synchronous communication, which exploits the public infrastructure of the XMPP network to reduce the costs.

Finally, this research contributed to take another step towards the consistent blending of synchronous and asynchronous text-based communication tools in the workplace. Email is the most successful collaborative tool but primarily a means to exchange asynchronous messages. Before becoming an indispensable tool ubiquitous in every workplace, email was initially used by the niche of research community and opposed by management. Nowadays, chat and IM are following a similar evolution path. At first mostly used by teenagers for exchanging 'social' messages, these synchronous tools have been recently spreading more and more in the workplace, although looked upon with suspicion by management as a source of continuous interruptions. In contrast, these tools provide a lightweight means to ascertain availability and interruptibility of coworkers and contact them in a timely manner. In this dissertation we presented JabberPresence, a Mozilla Thunderbird plugin designed to merge the asynchronicity of email with the synchronicity of IM and chat, so as to foster communication and better coordinate collaboration at a distance.

## 9.2. Future Work

Further work will be performed in two distinct directions. First, further statistical analyses need to be applied to the remaining data collected from the empirical investigation described in Chapter 8. The findings reported in this dissertation resulted from the analysis performed on the post-hoc satisfaction questionnaires, administered to the subjects after the requirements workshops, and hence, they specifically address the way stakeholders perceived the workshop process performance. Nevertheless, in order to confirm the validity of the findings, we are performing further analyses on the workshop outcomes, i.e., the specification documents RS 1.0 and RS 2.0, produced by stakeholders as a result of the requirements elicitation and negotiation workshops, respectively. These analyses will allow us to verify the effects of text-based communication on both group process and outcome. Secondly, the tools presented in

this dissertation, namely the JabberPresence Thunderbird plugin and eConference, will be further developed. With respect to JabberPresence, as of this writing, we are merging the code of our plugin with Sameplace, an IM plugin available, instead, only for the Firefox browser, so as to realize a unique IM extension for the two Mozilla applications. With regard to eConference, we are currently developing the fourth generation of our tool, building an abstract communication protocol layer upon the Eclipse Communication Framework (ECF). This project, named ***eConference over ECF***, is funded by IBM through the 2006 Eclipse Innovation Award.

# LIST OF PUBLICATIONS

**International Journals**

[1] Lanubile, F., Mallardo, T., and **Calefato, F**. Tool Support for Geographically Dispersed Inspection Teams. *Software Process: Improvement and Practice*, Wiley InterScience, Vol.8, No. 4, October/December 2003, pp. 217-231.

[2] **Calefato, F.**, Lanubile, F., and Mallardo, T. Function Clone Detection in Web Applications: A Semiautomated Approach. *Journal of Web Engineering*, Vol.3, No.1, Rinton Press, May 2004, pp. 3-21.

**International Conferences**

[3] Lanubile, F., Mallardo, T., **Calefato, F.,** Denger, C., and Ciolkowski, M. Assessing the Impact of Active Guidance for Defect Detection: A Replicated Experiment. *Proc. 10th Int'l Software Metrics Symposium (METRICS '04)*, Chicago, Illinois, USA, IEEE Computer Society, September 2004, pp. 269-279.

[4] **Calefato, F**., and Lanubile, F. Plugging Presence Awareness into Mozilla Thunderbird. *Proc. Int'l Conf. on Internet and Multimedia Systems and Applications (EuroIMSA '06)*, Innsbruck, Austria, 13-15 February, 2006, pp. 109-114.

[5] **Calefato, F**., Colagrossi, A., Gendarmi, D., Lanubile, F., Semeraro, G. An Information Broker For Integrating Heterogeneous Hydrologic Data Sources: A Web Services Approach. Proc. *The IFIP TC8 International Conference on Research and Practical Issues of Enterprise Information System (CONFENIS '06)*, IFIP Series (Springer), Vol. 205, 2006, pp. 41-50.

[6] **Calefato, F**., Damian, D., and Lanubile, F. An Empirical Investigation on Text-Based Communication in Distributed Requirements Engineering. To appear in the *Proc. 2nd Int'l Conf. Global Software Engineering (ICGSE '07)*, Munich, Germany, IEEE Computer Society, 27-30 August, 2007.

[7] Abbattista, F., **Calefato, F.**, Lanubile, F., and Gendarmi, D. Shaping Personal Information Spaces from Collaborative Tagging Systems. To appear in the *Proc. 11th Int'l Conf. on Knowledge-Based & Intelligent Information & Engineering Systems (KES '07)*, Lecture Notes in Computer Science (Springer-Verlag), Vietri sul Mare, Italy, 12-14 Sept. 2007.

**International Workshops**

[8] **Calefato, F.**, Lanubile, F., and Mallardo, T. Peer-to-Peer Remote Conferencing. *Proc. ICSE Int'l Workshop on Global Software Development (GSD '04)*, Edinburgh, Scotland, UK, IEE Publishing, May 2004, pp. 34-38.

[9] **Calefato, F.**, and Lanubile, F. A Decentralized Conferencing Tool for Ad-hoc Distributed Workgroups. *Proc. ASE Workshop on Cooperative Support for Distributed Software Engineering Processes (CSSE '04)*, Linz, Austria, Austrian Computer Society, 21 Sept. 2004, pp. 27-38.

[10] **Calefato, F.**, and Lanubile, F. Using The eConference Tool For Synchronous Distributed Requirements Workshops. *Proc. 1st Int'l Workshop on Distributed Software Development (DiSD '05)*, Paris, France, Austrian Computer Society, 29 August, 2005, pp. 97-108.

**National Workshops**

[11] **Calefato, F.** Improving Collaboration in Ad Hoc Development Teams. *3rd Italian Workshop on Software Engineering (GIIS '06).* Geneva, Italy, October 2006.

[12] **Calefato, F.**, and Lanubile, F. eConference over ECF. *Eclipse Italian Community Meeting*, Rome, Italy, 1 December, 2006

# REFERENCES


**[Abe90]** Abel, M. J. *Experiences in an Exploratory Distributed Organization, in Intellectual Teamwork: Social Foundations of Cooperative Work.* Lawrence Erlbaum Associates, Hillsdale, NJ, 1990.

**[Aga90]** Agarwal, R., Tanniru, M.R. Knowledge Acquisition Using Structured Interviewing: An Empirical Investigation, Journal of Management Information Systems, Vol. 7, No. 1, 1990, pp. 123-140.

**[Age06]** Ågerfalk, P.J., and Fitzgerald, B. Flexible and Distributed Software Processes: Old Petunias in New Bowls? *Communications of the ACM, Special Issue on Flexible and Distributed Software Processes*, Vol. 49, No. 10, 2006.

**[Alg03]** Alge, B.J., Wiethoff, C., and Klein, H.J. When does the medium matter? Knowledge-building experiences and opportunities in decision-making teams. *Journ. on Organizational Behavior and Human Decision Processes*, Vol. 91, 2003, pp. 26–37.

**[All77]** Allen, T.J. *Managing the Flow of Technology*. MIT Press, 1977.

**[AlR96]** Al-Rawas, A., and Easterbrook, S. Communication Problems In Requirements Engineering: A Field Study. *Proc. 1st Westminster Conference on Professional Awareness in Software Engineering*, Royal Society, London, 1-2 February, 1996.

**[Ant05]** Antoniu, G., Hatcher, P. Jan, M., Noblet, D.A., Performance Evaluation of JXTA Communication Layers. *Proc. 5th IEEE/ACM Int'l Workshop on Global and Peer-to-Peer Computing (GP2PC '05)*, IEEE Press, Cardiff, UK, 2005.

**[Aur05]** Aurum, A., and Wohlin, C. *Requirements Engineering: Setting the Context*, in Engineering and Managing Software Requirements, Aurum, A., and Wohlin, C. (eds.). Springer, 2005.

**[Bab06]** Babar, M.A., Kitchenham, B., Zhu, L., Gorton, I., and Jeffrey, R. An Empirical Study of Groupware for Distributed Software Architecture Evaluation Process. *The Journal of System and Software*. Vol. 79, No. 7, 2005, pp.912-925.

**[Bal51]** Bales, R.F. *Interaction Process Analysis: A Method for the Study of Small Groups*. Addison-Wesley, 1951.

**[Bal79]** Bales, R.F., and Cohen, S.P. *SYMLOG: A system for multiple observation of groups*, New York, Free Press, 1979.

**[Bec89]** Beck, K, and Cunningham, W. *A Laboratory for Teaching Object-Oriented Thinking*. Proc. Conference on Object Oriented Programming, Systems, Languages, and Application (OOPSLA '89), New Orleans, USA, 1-6 October, 1989.

**[Bel03]** Bellotti, V., Duchenaut, N., Howard, M., and Smith, I. Taking Email to Task: The Design and Evaluation of a Task Management Centered Email Tool. *Proc. Int'l ACM Conference on Human Factors in Computer Systems (CHI '03)*, 2003, pp. 345-352.

**[Ben93]** Benbasat, I., and Lim, L. The Effects of Group, Task, Context and Technology Variables on the Usefulness of Group Support Systems: A Meta-analysis of Experimental Studies. *Small Group Research*, Vol. 24, 1993, pp. 430-462.

**[Ben94]** Benford, S., Bowers, J., Fahlen, L. E., Mariani, J., and Rodden, T. Supporting Co-Operative Work in Virtual Environments. *The Computer Journal*, Vol. 37, No. 8, 1994, pp. 653-668.

**[Bir05a]** Birsan, D. On Plug-ins and Extensible Architectures. *ACM Queue*, Vol. 3, No. 2, March 2005, pp. 40-46.

**[Bir05b]** Birnholtz, J.P., Finholt, T.A., Horn, D.B., and Bae, S.J. Grounding Needs: Achieving Common Ground Via Lightweight Chat In Large, Distributed, Ad-Hoc Groups. *Proc. Int'l Conf. on Human Factors in Computing Systems (CHI '05)*, Portland, USA, April 2-7, 2005, pp. 21-30.

**[Boe88]** Boehm, B.W. A Spiral Model of Software Development and Enhancement. *IEEE Computer*, Vol. 21, No, 5, 1988, pp.61-72.

**[Boe98]** Boehm, B.W., Egyed, A., Kwan, J., Port, D., Shah, A., and Madachy, R. Using the WinWin Spiral Model: A Case Study. *IEEE Computer*, Vol. 31, No. 17, July 1998, pp. 33-44.

**[Boe00a]** Boehm, B.W., *Spiral Development: Experiences, Principles, and Refinements*. Hansen W.J. (ed.), CMU/SEI-00-SR-08, 2000.

**[Boe00b]** Boehm, B.W., Port, D., Al-Said, M. Avoiding the Software Model-Clash Spiderweb, *IEEE Computer*, 2000, pp. 120-123.

**[Boe01a]** Boehm, B.W., Grünbacher, P., and Briggs, R.O. Developing Groupware for Requirements Negotiation: Lessons Learned. *IEEE Software*, Vol. 18, No. 3, 2001, pp. 46-55.

**[Boe01b]** Boehm, B.W., and Basili, V.R.: Software Defect Reduction Top-10 List. Computer, *IEEE Computer*, Vol. 34, No. 1, 2001, pp. 135-137.

**[Boe01c]** Boehm, B.W., Basili, V.R.: Software Defect Reduction Top-10 List. *IEEE Computer*, Vol. 34, No. 1, 2001, pp. 135-137.

**[Boo03]** Booch G., and Brown A.W. Collaborative Development Environments. *Advances in Computers*, Vol. 59, 2003.

**[Bor70]** Bormann, E.G. The Paradox and Promise of Small Group Research. *Speech Monographs*, Vol. 37, No. 3, 1970, pp. 211-217.

**[Bor00]** Borghoff, U.M., and Schlichte, J.H. *Computer-Supported Cooperative Work: Introduction to Distributed Applications*. Springer, 2000.

**[Boy93]** Boyd, J. Floor Control Policies in Multi-User Applications. *Proc. Int'l Conference on Human Factors in Computing Systems*, Amsterdam, The Netherlands, 1993, pp. 107-108.

**[Boy98]** Boyer, D.G, Handel, M.J., and Herbsleb, J. Virtual community presence awareness. *ACM SIGGROUP Bulletin*, Vol. 19 , No. 3, 1998, pp. 11-14.

**[Cal04a]** Calefato, F., Lanubile, F., and Mallardo, T. Peer-to-Peer Remote Conferencing. *Proc. ICSE Workshop on Global Software Development (GSD '04)*, Edinburgh, Scotland, UK, IEE Publishing, 2004, pp. 34 -38.

**[Cal04b]** Calefato, F., and Lanubile, F. A Decentralized Conferencing Tool for Ad-hoc Distributed Workgroups. *Proc. ASE Workshop on Cooperative Support*

*for Distributed Software Engineering Processes (CSSE '04)*, Linz, Austria, Austrian Computer Society, 2004, pp. 27-38.

**[Cal06a]** Calefato, F., Lanubile, F. Plugging Presence Awareness into Mozilla Thunderbird. *Proc. Int'l Conf. Internet and Multimedia Systems and Applications (EuroIMSA '06)*, Innsbruck, Austria, 13-15 February, 2006, pp. 109-114.

**[Car94]** Carlson, J., and Zmud, R. Channel Expansion Theory: A Dynamic View of Media and Information Richness Perceptions. In *Academy of Management Best Papers Proceedings*, Moore, D.P. (ed.), 1994, pp. 280-284.

**[Car95]** Carlson, J. *Channel Expansion Theory: A Dynamic View of Media and Information Richness Perceptions*. PhD dissertation, Florida State University, Tallahassee, 1995.

**[Car99a]** Carlson, J.R., and Zmud, R.W. Channel Expansion Theory and The Experiential Nature of Media Richness Perceptions. *Academic Management Journal*, Vol. 42, No. 2, 1999, pp. 153–170.

**[Car99b]** Carmel, E. *Global Software Teams*. Prentice Hall, Upper Saddle River, NJ, 1999.

**[Car01]** Carmel E., and Agarwal, R. Tactical Approaches for Alleviating Distance in Global Software Development. *IEEE Software*, Vol. 18, No. 2, March/April 2001, pp. 22-29.

**[Chu99]** Churchill, E.F., and Bly, S. It's all in the words: Supporting work activities with lightweight tools, *Int'l Conf. Supporting Group Work (Group '99)*, Phoenix, AZ, USA, 1999.

**[Chr97]** Christensen, E, Fjermestad, J. Challenging Group Support Systems Research: The Case for Strategic Decision Making. *Group Decision and Negotiation*, Vol. 6, 1997, 351-372.

**[Cic85]** Cicchetti, D.V., Showalter, D., and Tyrer, P.J. The Effect of Number of Rating Scale Categories on Levels of Inter-Rater Reliability: A Monte Carlo Investigation. *Applied Psychological Measurement*, Vol. 9, 1985, pp. 31-36.

**[Cla91]** Clark H.H., and Brennan, S.E. *Grounding in Communication*, in Perspectives on Socially Shared Cognition, American Psychological Association, Washington DC, 1991, pp. 127-149.

**[Coa04]** Coar, K. The Sun Never Sets on Distributed Development. *ACM Queue*, Vol. 1, No. 9, 2004, pp. 32-39.

**[Coh60]** Cohen, J. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, Vol. 20, 1960, pp.37-46.

**[Coh04]** Cohn, M. *User Stories Applied: For Agile Software Development*, Addison Wesley, 2004.

**[Con80]** Conover, W.J. *Practical Nonparametric Statistics (2nd ed.).* J. Wiley & Sons, New York, NJ, 1980.

**[Cou04]** Coulin C., and Zowghi D. Requirements Elicitation for Complex Systems: Theory and Practice. In *Requirements Engineering for Sociotechnical Systems*, Maté J.L., and Silva, A. (eds.), Idea Group, 2005.

**[Cro51]** Cronbach, L.J. Coefficient Alpha and The Internal Structure of Tests. *Psychometrika,* Vol. 16, 1951, pp. 297-334.

**[Cum00]** Cummins, R.A., and Gullone, E. Why We Should Not Use 5-Point Likert Scales: The Case for Subjective Quality of Life Measurement. *Proc. 2nd Int'l Conf. Quality of Life in Cities*, Singapore, 8-10 March, 2000, pp. 74-93.

**[Cun01]** Cunningham, W., Leuf, B. *The Wiki Way. Quick Collaboration on the Web*. Addison-Wesley, 2001.

**[Cur88]** Curtis, B., Krasner, H., and Iscoe, N. A Field Study of the Software Design Process for Large Systems. *Comm. ACM*, Vol. 31, 1988, pp. 1286-1287.

**[Daf79]** Daft, R.L., and Wiginton J. Language and Organization. *Acad. Management Rev.*, Vol. 4, No. 2, 1979, pp. 179-191.

**[Daf84]** Daft, R.L., and Lengel, R.H. Information Richness: A New Approach to Managerial Behaviour and Organizational Design. In Staw, BM., and Cummings, L.L. (eds.) Research in Organizational Behaviour, CT JAI Press, Greenwich Vol. 6, 1984, pp. 191-233.

**[Daf86]** Daft, R.L., and Lengel, R.H. Organizational Information Requirements, Media Richness and Structural Design. Management Science, Vol. 32, No. 5, 1986, pp. 554-571.

**[Dam01]** Damian, D. Empirical Studies of Computer Support for Distributed Requirements Negotiation. *PhD Dissertation*, Dept. of Cs, University of Calgary, 2001.

**[Dam03a]** Damian, D., and Zowghi, D. Requirements Engineering Challenges in Multi-Site Software Development Organizations. *Requirements Engineering Journal*, Vol. 8, 2003, pp. 149-160.

**[Dam03b]** Damian, D., Eberlein, A., Shaw, M., and Gaines, B. An Exploratory Study of Facilitation in Distributed Requirements Engineering. *Requirements Engineering*, Vol. 8, No. 1, 2003, pp. 23-41.

**[Dav93]** Davis, A.M. *Software Requirements: Objects, Functions, and States.* Englewood Cliffs, NJ, Prentice- Hall, 1993.

**[Dav94]** Davis, A.M. Software Requirements: Analysis and Specification. Prentice-Hall, NJ, 1990.

**[Dav95]** Davison, R. Socio-Psychological Aspects of Group Processes, 1995, available at: http://www.is.cityu.edu.hk/Research/WorkingPapers/paper/9503.pdf

**[Den90]** Dennis, A.R., Easton, A.C., Easton, G.K., George, J.F. and Nunamaker, J.F. Ad hoc versus established groups in an electronic meeting system environment. *Proc. 23rd Hawaii Int'l Conf. on System Sciences (HICSS-23)*, 1990, Vol. 3, pp. 23-29.

**[Den91]** Dennis, A.R., Nunamaker, J.F., and Vogel, D.R. A Comparison of Laboratory and Field Research in the Study of Electronic Meeting Systems. *Journal of Management Information Systems*, Vol. 7, No.2, 1991, pp. 107-135.

**[Den93]** Dennis, A. R., and Gallupe, R.B. A History of GSS Empirical Research: Lessons Learned and Future Directions, In *Group Support Systems: New Perspectives*,Jessup, L., and Valacich, J. (eds.), MacMillan, NY, 1993.

**[Den98a]** Dennis, A.R., and Valacich. J.S. Testing Media Richness Theory in the New Media: The Effects of Cues, Feedback, and Task Equivocality. *Information Systems Research*, Vol. 9, No. 3, 1998, pp. 156-274.

**[Den98b]** Dennis, A.R., Valacich. J.S., Speier, C., and Morris, M.G. Beyond media richness: an empirical test of media synchronicity theory. *Proc. 31st Hawaii Int'l Conf. on System Sciences (HICSS-31)*, 1998, Vol. 1, pp. 48-57.

**[Den99]** Dennis, A.R., and Valacich. J.S. Rethinking Media Richness: Towards a Theory of Media Synchronicity. *Proc. 32nd Hawaii Int'l Conf. on System Sciences (HICSS-32)*, Vol. 1, 1999.

**[deS04]** de Souza, C.R.B., Redmiles, D., Millen, D., Patterson, J. Sometimes you need to see through walls: a field study of application programming interfaces. *Proc. Int'l Conf. on Computer Supported Cooperative Work (CSCW '04)*, Chicago, Illinois, USA, November 6-10, 2004, pp. 63-71.

**[Dil95**] Dillenbourg, P., Baker, M. Blaye, A., O'Malley, C.. The Evolution of Research on Collaborative Learning. In *Learning in humans and machines. Towards an interdisciplinary learning science*. Reimann P., and Spada, H. (eds.), London, Pergamon, 1995, pp. 189-211.

**[Dix03]** Dix, A., Finley, J., Abowd, G., and Beale, R. *Human-computer interaction (3rd ed.).* Prentice-Hall, 2003.

**[Dou92]** Dourish, P., and Bly, S. Portholes: Supporting Awareness in a Distributed Work Group, *Int'l Conf. Computer-Human Interaction (CHI '92)*, Monterey, CA, USA, 1992, pp. 541-547.

**[Duc01]** Ducheneaut, N., Bellotti, V., E-mail as Habitat: An Exploration of Embedded Personal Information Management. *ACM interactions*, Vol. 9, No. 5, 2001, pp. 30-38.

**[Eas91]** Easterbrook, S. Handling Conflict Between Domain Descriptions With Computer-Supported Negotiation. *Knowledge Acquisition: An International Journal*, Vol. 3, 1991, pp. 255-289.

**[Eas94]** Easterbrook, S. *Resolving Requirements Conflicts with Computer-Supported Negotiation*. In *Requirements Engineering: Social and Technical Issues*, M. Jirotka and J. Goguen (eds.), London: Academic Press, 1994.

**[Ebe01]** Ebert, C,. and De Neve, P. Surviving Global Software Development. *IEEE Software*, Vol. 18, No. 2, 2001, pp. 62-69.

**[EIA06]** IBM Eclipse Innovation Award - 2006 Competition – http://www-304.ibm.com/jct09002c/us/en/university/scholars/products/eclipse/eig_2006.html

**[Ell91]** Ellis, C.A., Gibbs, S.J., and Rein, G. Groupware: Some Issues and Experiences. *Commun. ACM*, Vol. 34, No. 1, Jan. 1991, pp. 39-58.

**[Ens90]** Ensor, R. How can we make groupware practical? (panel moderator) *Proc. ACM Int'l Conf. on Human Factors in Computing Systems (CHI '90)*, Seattle, WA, USA, 1990, pp. 87-89.

**[Eri00]** Erickson, T., and Kellogg, W.A. Social Translucence: An Approach to Designing Systems that Support Social Processes. *ACM Transactions on Computer-Human Interaction (TOCHI)*, March 2000, pp. 59-83.

**[Fis92]** Fish, R.S., Kraut, R.E., Root, R.W., and Rice, R.E. Evaluating Video as a Technology for Informal Communication, *Int'l Conf. Computer-Human Interaction (CHI '92)*, Monterey, CA, USA, 1992.

**[Fin97]** Finn, K.E., Sellen, A.J., and Wilbur, S.B. *Video-Mediated Communication*. Lawrence Erlbaum Assoc. Inc., Hillsdale, NJ, 1997.

**[Fje00]** Fjermestad, J., and Hiltz, R. Case and Field Studies of Group Support Systems: An Empirical Assessment. *Proc. Int'l Hawaii Conference on System Science (HICSS-33)*, Hawaii, Jan. Vol. 1, pp. 4-7.

**[Fod94]** Foddy, W. *Constructing Questions for Interviews and Questionnaires*. Cambridge University Press, 1994.

**[Fow02]** Fowler, M. Public versus published interfaces. *IEEE Software*, Vol. 19, No. 2, March/April 2002, pp. 18-19.

**[Gar91]** Garland, R. The Mid-Point on a Rating Scale: Is it Desirable? *Marketing Bulletin*, Vol. 2, 1991, pp. 66-70.

**[Gar95]** Garlan, D., Allen, R., and Ockerbloom, J. Architectural mismatch: why reuse is so hard, *IEEE Software*, Vol. 12, No. 6, November 1995, pp. 17-26.

**[Gey03]** Geyer, W., Vogel, J., Cheng, L.T. and Muller, M. Supporting Activity-centric Collaboration through Peer-to-Peer Shared Objects. *Proc. Int'l*

*Conference on Supporting Group Work (Group '03)*, Sanibel Island, FL, USA, 2003.

**[God00]** Godefroid, P.I., Herbsleb, J.D., Jagadeesan, L.J., and Li, D. Presence Awareness in Collaborative Systems, US Patent No. 6697840, Feb 29, 2000.

**[Gog93]** Goguen J.A., and Linde, C. *Techniques for Requirements Elimination***,** Proc. IEEE Int'l symposium on Requirements Engineering, San Diego, USA, 1993, pp. 152-164.

**[Gon01]** Gong, L. JXTA: A network programming environment. *IEEE Internet Computing*, Vol. 5, No. 3, 2001, pp. 88-95.

**[Goo95]** Goodhue, D.L., and Thompson, R.L. Task-Technology Fit and Individual Performance. *MIS Quarterly*, Vol. 19, No. 2, Jun. 1995, pp. 213-236.

**[Got02]** Gottesdiener, E. *Requirements by Collaboration*. Addison-Wesley, MA, 2002.

**[Gre88]** Greif, I. Computer-Supported Cooperative Work: A Book of Readings, Morgan-Kaufmann, San Mateo, 1988.

**[Gre91]** Greenberg, S. Personalizable groupware: Accommodating individual roles and group differences. *Proc. 2nd European Conference on Computer Supported Cooperative Work (ECSCW '91)*, Amsterdam, The Netherlands, September, 24-27, 1991, pp.17-31.

**[Gri99]** Grinter, R.E., Herbsleb, J.D., and Perry, D.E. The Geography of Coordination: Dealing with Distance in R&D Work. *Proc. Int'l ACM SIGGROUP Conf. on Supporting Group Work (GROUP '99)*, Phoenix, AZ, USA, 1999, pp. 306-315.

**[Gru94a]** Grudin, J. Groupware and Social Dynamics: Eight Challenges for Developers. *Communications ACM*, Vol. 37, No.1, Jan. 1994, pp. 92-105.

**[Gru94b]** Grudin, J. Computer-Supported Cooperative Work: History and focus. *IEEE Computer*, Vol. 27, No. 5, May 1994, pp. 19-26.

**[Gru05a]** Gruber, O., Hargrave, B.J., Mcaffer, J., Rapicault, P., and Watson, T. The Eclipse 3.0 platform: Adopting OSGi technology. *IBM Systems Journal*, Vol. 44, No. 2, 2005, pp. 289-299.

**[Gru05b]** Grünbacher, P., and Seyff, N. *Requirements Negotiation*. In *Engineering and Managing Software Requirements*, Aurum, A., and Wohlin, C. (eds.), Springer, 2005.

**[Gun97]** Gunawardena, C.N., and Zittle, F.J. Social Presence as a Predictor of Satisfaction within a Computer-Mediated Conferencing Environment, *American Journal of Distance Education*, Vol. 11, No. 3, 1997, pp. 8-26.

**[Hac69]** Hackman, J.R. Toward Understanding the Role of Tasks in Behavioral Research. *Acta Psychologica*, Vol. 31, 1969, pp. 97-128.

**[Hal66]** J Hall, J., and Williams, M.S. A Comparison of Decision Making Performance in Established and Ad Hoc Groups. *Journal of Personality and Social Psychology*, Vol. 3, 1966, pp. 214-222.

**[Hal03]** Halepovic, E., and Deters, R. The Cost of Using JXTA. *Proc. 3rd IEEE Int'l Conf. on Peer-to-Peer Computing (P2P '03)*, Linköping, Sweden, IEEE Press, 2003, pp. 160-167.

**[Han02]** Handel, M. and Herbsleb, J.D. What is Chat Doing in the Workplace? *Proc. Int'l Conference on Computer-Supported Cooperative Work (CSCW '02)*, New Orleans, LA, USA, 2002.

**[Har97]** Harter, J.K. *The Psychometric Utility of The Midpoint on a Likert Scale*. PhD Dissertation, The University Of Nebraska, Lincoln, 1997.

**[Hat94]** Hatcher, L., and Stepanski, E.J. *Step-By-Step Approach to Using the Sas System for Univariate and Multivariate Statistics*, Paperback, 1994.

**[Hau05]** Hauswirth, M., Podnar, I., and Decker, S. On P2P Collaboration Infrastructures. *14th IEEE Int'l WETICE Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise (DMC '05)*, Linkoping, Sweden, 13-15 June 2005, pp. 66-71.

**[Hef04]** Hefke, M., and Stojanovic, L. An Ontology-Based Approach for Competence Bundling and Composition of Ad-Hoc Teams in an Organization. *Proc. 4th Int'l Conf. on Knowledge Management (I-KNOW '04), Journal of Universal Computer Science (J.UCS)*, Graz, Austria, June, 2004, pp. 126-134.

**[Her99a]** Herbsleb, J.D., and Grinter, R.E. Splitting the Organization and Integrating the Code: Conway's Law Revisited. *Proc. Int'l Conference on Software Engineering (ICSE '99)*, Los Angeles, CA, USA, 1999.

**[Her99b]** Herbsleb, J.D., and Grinter, R.E. Architectures, Coordination, and Distance: Conway's Law and Beyond. *IEEE Software*, Sep/Oct 1999 Vol. 16, No. 5, pp. 63-70.

**[Her00]** Herbsleb, J.D., Mockus, A., Finholt, T.A., and Grinter, R.E. Distance, Dependencies, and Delay in a Global Collaboration. *Proc. Int'l Conference on Computer-Supported Cooperative Work (CSCW '00)*, Philadelphia, PA, USA, 2000.

**[Her01a]** Herbsleb, J.D., and Moitra, D. Global Software Development. *IEEE Software*, Vol. 18, No. 2, March/April 2001, pp. 16-20.

**[Her01b]** Herbsleb, J.D., and Grinter, R.E. Architecture, Coordination, and Distance: Conway's Law and Beyond. *IEEE Software*, Vol. 16, No. 5, September/October 2001, pp. 63-70.

**[Her01c]** Herbsleb, J.D., Mockus, A., Finholt, T.A., and Grinter, R.E. An Empirical Study of Global Software Development: Distance and Speed. *Proc. Int'l Conference on Software Engineering (ICSE '01)*, Toronto, Canada, 2001, pp. 81-90.

**[Her01d]** Herlea, D., and Greenberg, S. Using a Groupware Space for Distributed Requirements Engineering. *Proc. Int'l Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, Stanford, CA, USA, 17-19 June 1998, pp.57-62.

**[Her02]** Herbsleb, J.D., Atkins, D.L., Boyer, D.G., Handel, M., and Finholt, T.A. Introducing Instant Messaging and Chat into the Workplace. *Proc. Int'l Conference on Computer-Human Interaction (CHI '02)*, Minneapolis, MN, USA, 2002.

**[Her03]** Herbsleb, J.D., and Mockus T.A. An Empirical Study of Speed and Communication in Globally Distributed Software Development. *IEEE Transactions on Software Engineering*, Vol. 29, No. 6, 2003, pp. 481-494.

**[Hol93]** Hollingshead, A.B., Mcgrath, J.E., and O'Connor, K.M. Group Task Performance and Communication Technology. *Small Group Research*, Vol. 24, No. 3, 1993, pp. 307-333.

**[Hös00]** Höst, M., Regnell, B., and Wohlin, C. Using Students as Subjects - A Comparative Study of Students and Professionals in Lead-Time Impact Assessment. *Empirical Software Engineering*, Vol. 5, No. 3, 2000, pp. 201-214.

**[Hud02]** Hudson, J., Christensen, J., Kellogg, W., Erickson, T. *I'd Be Overwhelmed, But it's Just One More Thing to Do: Availability and Interruption in Research Management*. Int'l Conf. Computer-Human Interaction (CHI '02), Minneapolis, MN, USA, 2002.

**[Hum97]** Humphrey, W.S. *Managing Technical People: Innovation, Teamwork, and The Software Process*. Reading, Addison-Wesley, 1997.

**[Isa94]** Isaacs, E.A., and Tang, J.C. What Video Can and Cannot Do for Collaboration: A Case Study. *Multimedia Systems*, Vol. 2, 1994, pp. 63-73.

**[Jen77]** Jenkins, G.D., and Taber, T.D. A Monte Carlo Study of Factors Affecting Three Indices of Composite Scale Reliability. *Journal of Applied Psychology*, Vol. 62, 1977, pp. 392-398.

**[Joh82]** Johnson-Lentz, P., and Johnson-Lentz, T. Groupware: The Process and Impacts of Design Choices. In *Computer-Mediated Communication Systems: Status and Evaluation*. Kerr, E.B., and Hilts, S.R. (eds.), New York, Academic Press, 1982.

**[Joh88]** Johansen, R. *Groupware: Computer Support for Business Teams*. New York, Free Press, 1988.

**[Joh05]** Johns, R. One Size Doesn't Fit All: Selecting Response Scales For Attitude Items. *Journal of Elections, Public Opinion, and Parties*, Vol. 15, No. 2, 2005, pp. 237-264.

**[Jon96]** Jones, C. *Applied Software Measurement: Assuring Productivity and Quality*. McGraw-Hill, NY, 1996.

**[JXTA]** JXTA platform change history – http://platform.jxta.org/java/currentwork.html#history

**[Ker04]** Kerr, B., and Wilcox, E. Designing Remail: Reinventing the Email Client Through Innovation and Integration. *Proc. ACM Conf. on Human Factors in computing systems*, Vienna, Austria, 2004, pp. 837-852.

**[Kne47]** Kendall, M.G. *The advanced theory of statistics (vol. 1).* Charles Griffin & Co., London, 1947.

**[Kno95]** Knoll, K., and Jarvenpaa, S. Learning to Work in Distributed Global Teams. *Proc. 28th Hawaii Int'l Conf. on System Sciences (HICSS-28)*, 3-6 Jan. 1995, Vol. 4, pp. 92-101.

**[Kon96]** Kontoya, G., and Sommerville, I. Requirements Engineering with Viewpoints. *Software Engineering Journal*, Vol. 11, 1996, pp. 5-18.

**[Kon98]** Kontoya, G., and Sommerville, I. *Requirements Engineering – Process and Techniques*. John Wiley & Sons, UK, 1998.

**[Kos02]** Koskelainen, P., Schulzrinne, H., and Wu, X. A SIP-based Conference Control Framework. *Proc. Int'l International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV '02)*, Miami Beach, Florida, USA, May 12-14, 2002.

**[Kos06]** Koskelainen, P., Ott, J., Schulzrinne, H., and Wu, X. *Requirements for Floor Control Protocols*. IETF RFC 4376, Informational, Network Working Group – www.ietf.org/rfc/rfc4376.txt

**[Kra90a]** Kraut, R.E., Fish, R.S., Root, R.W., and Chalfonte, B.L. Informal Communication in Organizations: Form, Function, and Technology. In *People's Reactions to Technology*, Newberry Park, Sage, 1990.

**[Kra90b]** Kraut, R.E., Egido, C., Galegher, J. *Patterns of Contact and Communication in Scientific Research Collaboration. Intellectual Teamwork: Social Foundations of Cooperative Work*, J. Galegher, R.E. Kraut, and C. Egido, (eds.), pp. 149-171, 1990

**[Kra00]** Kraut, R., Mukhopadhyay, T., Szczypula, J., Kiesler, S., and Scherlis, B. Information and communication: Alternative uses of the Internet in households. *Information Systems Research*, Vol. 10, 2000, pp. 287-303.

**[Lan03a]** Lanubile, F. A P2P toolset for Requirements Elicitation. *Proc. ICSE Int'l Workshop on Global Software (GSD '03)*, Portland, Oregon, USA, 9 May, 2003.

**[Lan03b]** Lanubile, F., Mallardo, T., and Calefato, F. Tool Support for Geographically Dispersed Inspection Teams. *Software Process: Improvement and Practice*, Vol.8, No.4, Wiley InterScience, October/December 2003, pp. 217-231.

**[Lau90]** Lawers J.C., and Lantz, K.A. Collaboration awareness in support of collaboration transparency: requirements for the next generation of shared window systems. *Proc. SIGCHI Conf. on Human Factors in Computing Systems*, Seattle, WA, USA, 1990.

**[Lis75]** Lissitz, R.W., and Green, S.B. Effect of the number of scale points on reliability: A Monte Carlo approach. *Journal of Applied Psychology*, Vol. 60, 1975, pp. 10-13.

**[Llo02]** Lloyd., W.J., Rosson, M.B., and Arthur, J.D. Effectiveness of Elicitation Techniques in Distributed Requirements Engineering. *Proc. IEEE Int'l Conf. on Requirements Engineering (RE '02)*, Essen, Germany, 9-13 September 2002, pp. 311- 318.

**[Lor58]** Lorge, I., Fox, D.D., and Brenner, M. A Survey of Studies Contrasting the Quality of Group Performance and Individual Performance. *Psychological Bulletin*, Vol. 5, No. 6, 1958, pp. 337-357.

**[Lou95]** Loucopoulos, P., and Karakostas, V. *System Requirements Engineering*. McGraw-Hill, 1995.

**[Lud06]** Ludwig, S., Beda, J., Saint-Andre, P. Hildebrand, J. Egan, S, and McQueen, R. *Jingle*. XMPP protocol extension XEP-0166 – www.xmpp.org/extensions/xep-0166.html

**[Mar99]** Mark, G., Grudin, J., and Poltrock, S.E. Meeting at the Desktop: An Empirical Study of Virtually Collocated Teams. *Proc. 6th European Conf. on Computer-Supported Cooperative Work (ECSCW '99)*, Copenhagen, Denmark, 12-16 September, 1999, pp. 159-178.

**[Mac96]** Macaulay, L.A. *Requirements Engineering*. Springer-Verlag Telos, 1996.

**[Mac98]** Mackay, W. More Than Just a Communication System: Diversity in the Use of Electronic Mail, *Proc. ACM Int'l Conference on Computer Supported Co-operative Work (CSCW '98)*, 1998, pp. 344-353.

**[Mar99]** Mark, G. , Grudin, J., and Poltrock, S.E. Meeting at the desktop: an empirical study of virtually collocated teams. *Proc. 6th European conference on Computer supported cooperative work (ECSCW '99)*, Copenhagen, Denmark, 1999.

**[Mau97]** Maule, A.J., and Edland, A.C. The effects of time pressure on human judgment and decision making. In *Decision making: Cognitive models and explanations*, Ranyard, R., Crozier, W.R., and Svenson, O. (eds.), London: Routledge, 1997, pp. 189-204.

**[Men92]** Mennecke, B.E., Hoffer, J.A., and Wynne, B.E. The Implications of Group Development & History for GSS Theory & Practice. *Small Group Research*, Vol. 23, No. 4, 1992, pp. 524-272.

**[Men93]** Mennecke, B.E., and Wheeler, B.C. Tasks Matter: Modeling Group Task Processes In Experimental CSCW Research, *Proc. 26th Hawaii Int'l Conf. on System Sciences (HICSS-26)*, 1993, Vol. 4, pp. 71-80.

**[Men95]** Mennecke, B.E., Hoffer, J.A. and Valacich, J.S. An experimental examination of group history and group support system use on information sharing performance and user perceptions. *Proc. 28th Hawaii Int'l Conf. on System Sciences (HICSS-28)*, 1995, Vol. 4, pp. 153-162.

**[McA05]** McAffer, J., and Lemieux, J-M. *Eclipse Rich Client Platform: Designing, Coding, and Packaging Java™ Applications*, Addison Wesley Professional, 2005.

**[McG84]** McGrath, J.E. *Groups: Interaction and Performance*. Prentice Hall, 1984.

**[McG91]** McGrath, J.E. Time, Interaction, and Performance (TIP): A theory of groups. *Small Group Research*, Vol. 22, No. 2, 1991, pp.147-174.

**[McG93a]** McGrath, J.E., and Hollingshead, A.B. *Putting the "Group" Back in Group Support Systems: Some Theoretical Issues about Dynamic Processes in Groups with Technological Enhancements*. In Group Support Systems: New

Perspectives, L.M. Jessup & J.S. Valacich (eds.), New York: Macmillan, 1993, pp. 78-96.

**[McG93b]** McGrath, J.E. Methods for the Study of Groups. In *Groupware and Computer-Supported Cooperative Work*. Baecker, R.M. (ed.), Morgan Kaufmann, San Mateo, 1993, pp.200-204.

**[McG94]** McGrath, J.E., and Hollingshead, A.B. Groups Interacting with Technology: Ideas, Evidence, Issues and an Agenda. Sage, 1994.

**[Moc01]** Mockus, A., and Herbsleb, J. Challenges of Global Software Development. *7th Int'l Software Metrics Symposium (METRICS '01)*, London, UK, 4-6 April 2001, pp. 182-183.

**[Mon96]** Montgomery, D.C. *Design and Analysis of Experiments ( 4th ed.)*. J. Wiley & Sons, New York, 1996.

**[Mor04]** Morris, M.R., Ryall, K., Shen, C., Forlines, C., and Vernier, F. Beyond "social protocols": multi-user coordination policies for co-located groupware. *Proc. ACM conference on Computer Supported Cooperative Work (CSCW '04)*, Chicago, Illinois, USA, 2004, pp. 262-265.

**[Mul03]** Muller, M.J., Raven, M.E., Kogan, S., Millen, D,R, and Carey, K. Introducing Chat Into Business Organizations: Toward an Instant Messaging Maturity Model. *Proc. ACM Int'l Conference on Supporting Group Work*, Sanibel Island, Florida, USA, 2003, pp. 50-57.

**[Mur00]** Murthy, U.S., and Kerr, D.S. Task/Technology Fit and The Effectiveness of Group Support Systems: Evidence in The Context of Tasks Requiring Domain Specific Knowledge. *Proc. 33rd Hawaii Int'l Conf. on System Sciences (HICSS-33)*, 2000, Vol. 2, pp. 1-10.

**[Nas02]** Nasscom-Mckinsey report 2002 – http://www.nasscom.org/artdisplay.asp?Art_id=1225

**[Nun78]** Nunnally, J. *Psychometric Theory*. McGraw-Hill, New York, NJ, 1978.

**[Nun91]** Nunamaker, J.F., Dennis, A.R., Valacich, J.S., Vogel, D.R., and George, J.F. Electronic Meeting Systems to Support Group Work. *Commun. ACM*, Vol. 34, No. 7, 1991, pp. 40-61.

**[Nus00]** Nuseibeh, B., and Easterbrook. S. Requirements engineering: a roadmap. *Proc. Int'l Conference on Software Engineering (ICSE '00)*, Limerick, Ireland, 2000, pp. 35-46.

**[Oba98]** Obata, A., and Sasaki, K. OfficeWalker: A virtual visiting system based on proxemics. *Proc. Int'l Conf. Computer-Supported Cooperative Work (CSCW '98)*. Seattle, WA, USA, 1998.

**[Ols95]** Olson, J.S., Olson, G.M., and Meader, D.K. What Mix Of Video and Audio is Useful for Remote Real-Time Work? *Proc. Int'l Conf. on Human Factors in Computing Systems (CHI '95)*. Denver, USA, 1995, pp. 362-368.

**[Ols97]** Olson, J.S., Olson, G.M., and Meader, D. *Face-To-Face Group Work Compared to Remote Group Work with and without Video*. In *Video Mediated Communication*, Finn, K., Sellen, A., and Wilbur, S. (eds.), Hillsdale, NJ: Lawrence Erlbaum Associates, 1997.

**[Ols00]** Olson G.M., and Olson J.S. Distance Matters, *Human-Computer Interaction*, Vol. 15, No. 2/3, 2000, pp. 139-178.

**[Orl92]** Orlikowski, W. Learning from Notes: Organizational issues in groupware implementation. *Proc Int'l. Conf. Computer Supported Cooperative Work (CSCW '92)*, Toronto, Canada, 1992, pp. 362-369.

**[Osa06]** Osais, Y., Abdala, S., and Matrawy, A. A Multilayer Peer-to-Peer Framework for Distributed Synchronous Collaboration. *IEEE Internet Computing*, Vol. 10, No. 6, 2006, pp. 33-41.

**[Osb79]** Osborn, A.F. *Applied Imagination*. Charles Scribner's Sons, NY, 1979.

**[Per91]** Perin, C. Electronic Social Fields in Bureaucracies. *Communication ACM*, Vol. 34 , No. 12, 1991, pp. 75-82.

**[Per94]** Perry, D.E., Staudenmayer, N.A., and Votta, L.G. People, Organizations, and Process Improvement. *IEEE Software*, Vol. 11, No. 4, pp. 36-45, 1994.

**[Phi89]** Phillips, G.M., and Santoro, G.M. Teaching group discussion via computer-mediated communication, *Communication Education*, Vol. 38, No. 2, 1989, pp. 151-161.

**[Pol05]** Poltrock S.E., and Grudin J. Videoconferencing: Recent Experiments and Reassessment. *Proc. 38th Hawaii Int'l Conf. on System Sciences (HICSS-38)*, 2005, pp. 104a-104a.

**[Pou97]** Pouloudi, A., Whitley, E.A. Stakeholder Identification in Inter-organizational Systems: Gaining insights for Drug Use Management Systems, Vol. 6, 1997, pp. 1-14.

**[Ras91]** Rasmussen, J., Brehmer, B., and Leplat, J. *Distributed Decision Making: Cognitive Models for Cooperative Work*. Chichester, England; New York, Wiley, 1991.

**[Rea99]** Reagle, J. Eskimo Snow and Scottish Rain: Legal Considerations of Schema Design. W3C Note, December 10, 1999. Available at: http://www.w3.org/TR/md-policy-design#_Social_Protocols

**[RCP-Bug]** Eclipse RCP Bug #5338: "Reduce RCP application size" – https://bugs.eclipse.org/bugs/show_bug.cgi?id=53338

**[Rob98]** Robinson, W.N, and Volkov, V. Supporting The Negotiations Life Cycle. *Comm. ACM*, Vol. 41, No. 5, 1998, pp. 95-102.

**[Rob05]** Robert, L.P., and Dennis, A.R. Paradox of Richness: A Cognitive Model of Media Choice. *IEEE Transactions on Professional Communication*, Vol. 48, No. 1, March 2005, pp.10-21.

**[Ros92]** Roseman, M., and Greenberg, S. GROUPKIT: a groupware toolkit for building real-time conferencing applications. *Proc. Int'l Conf. Computer Supported Cooperative Work (CSCW '92)*, Toronto, Ontario, Canada, 1992, pp. 43-50.

**[Sei03]** Seigneur, J.-M., Biegel, G., and Damsgaard, C. P2P with JXTA-Java Pipes. *Proc. Int'l Conf. Principles and Practice of Programming in Java*, ACM Press, 2003, pp. 207–212.

**[Sen06]** Sengupta, B., Chandra, S., and Sinha, V. A Research Agenda for Distributed Software Development. *Proc. Int'l Conference on Software Engineering (ICSE'06)*, Shanghai, China, 2006, pp. 731-740.

**[Sch98]** Schultze, U., and Vandenbosch, B. Information overload in a groupware environment: Now you see it, now you don't. *Journ. Org. Comput. Electron. Commerce*, Vol. 8, No. 2, 1998, pp. 127-148.

**[Sha49]** Shannon, C.E., and Weaver, W. *The Mathematical Theory of Communication*. Urbana, University of Illinois, 1949.

**[She92]** Sheridan, T.B. Musings on telepresence and virtual presence. *Presence*, Vol. 1, No. 1, 1992, pp. 120 - 126.

**[Sho76]** Short, J., Williams, E., and Christie, B. *The Social Psychology of Telecommunications*. London: John Wiley and Sons, 1976.

**[Smi05]** Smith, M. K. Bruce W. Tuckman - Forming, Storming, Norming and Performing in Groups. *The Encyclopedia of Informal Education*, 2005, available at: www.infed.org/thinkers/tuckman.htm.

**[Som01]** Sommerville, I. Software Engineering. Addison-Wesley, 2001.

**[Spr91]** Sproull, L., and Kiesler, S. *Connections*. Cambridge, MIT Press, 1991.

**[StA05]** St. Andre, P. Streaming XML with Jabber/XMPP. *IEEE Internet Computing*, Vol. 9, No. 5, 2005, pp. 82-89.

**[StA06]** Saint-Andre, P. *Multi-User Chat*. XMPP protocol extension *XEP-0045* – www.xmpp.org/extensions/xep-0045.html

**[Sut02]** Sutcliffe, A.G. *User-Centered Requirements Engineering*, Springer, London, 2002.

**[Sve93]** Svenson, O., and Maule, A.J. *Time Pressure and Stress in Human Judgment and Decision Making*. New York: Plenum, 1993.

**[Tan93]** Tang, J.C., and Isaacs, E.A. Why Do Users Like Video? *Computer Supported Cooperative Work*, Vol. 1, 1993, pp. 163-196.

**[Tog02]** Togneri, D.F., Falbo, R.A, Menezes, C.S. Supporting Cooperative Requirements Engineering with an Automated Tool. Proc. 5th Workshop on Requirements Engineering, Valencia, Spain, 11-22 Nov. 2002, 240-254.

**[Tre87]** Trevino, L., Lengel, R., and Daft, R. Media symbolism, media richness, and media choice in organizations. *Communication Research*, Vol.14, 1987, pp. 553-574.

**[Tre90]** Trevino, L., Lengel, R., Bodensteiner, W., Gerloff, E., and Muir, N. The richness imperative and cognitive style. *Management Communication Quarterly*, Vol. 4, 1990, 176-197.

**[Tuc65]** Tuckman, B.W. Developmental Sequence in Small Groups. *Psychological bulletin*, Vol. 63, No. 6, 1965, pp. 384-399.

**[Tuc01]** Tuckman, B.W. Developmental Sequence in Small Groups (reprint). *Group Facilitation: A Research and Applications Journal*, No. 3, 2001, available at http://dennislearningcenter.osu.edu/references/GROUP%20DEV%20ARTICLE.doc.

**[Tur04]** Turek, M. The Future Of Presence. *Business Communications Review*, May 2004, pp. 40-43.

**[Vau03]** Vaughan-Nichols, S.J. Presence Technology: More Than Just Instant Messaging. *IEEE Computer*, Vol. 36, No. 10, Oct. 2003, pp. 11- 13.

**[Wal86]** Walther, J.B. Computer-mediated Communication: Impersonal, Interpersonal, and Hyperpersonal Interaction. *Communication Research*, Vol. 23, No.1, 1996, pp. 3-43.

**[Whi94]** Whittaker, S., Frohlich, D., and Daly-Jones, O. Informal workplace communication: What is it like and how might we support it?, *Int'l Conf. Computer-Human Interaction (CHI '94)*, Boston, MA, USA, 1994.

**[Whi96]** Whittaker, S., and Sidner, C. Email Overload: Exploring Personal Information Management of Email. *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems*, Vancouver, BC, Canada, 1996, pp. 276-283.

**[Whi02]** Whittaker, S. Jones, Q., and Terveen, L. Persistence and Conversation Stream Management: Conversation and Contact Management. *Proc. 35th Hawaii International Conference on System Sciences (HICCS '02)*, Hawaii, 2002, pp. 335-344.

**[Whi05]** Whittaker, S., Bellotti, V., and Moody, P. Revisiting and Reinventing Email, *Spec. Issue Human-Computer Interaction*, Vol. 20, No. 1-2, 2005.

**[Wil91]** Wilson, P. *Computer-Supported Cooperative Work: An Introduction*. Oxford, UK, Intellect Books, 1991.

**[Woh00]** Wohlin, C., Runesson, P., Höst, M., Ohlsson, M.C., Regnell, B., and Wesslén, *A. Experimentation in Software Engineering, An Introduction*. Kluwer Academic Publishers, 2000.

**[Woo86]** Wood, R.E. Task Complexity: Definition of the Construct. *Organizational Behavior and Human Decision Processes*, Vol. 37, 1986, 60-82.

**[Woo95]** Wood, J., and Silver, D. Joint Application Development. John Wiley & Sons, NY, 1995.

**[XMPP]** eXtensible Messaging and Presence Protocol **(**XMPP) IETF RFC 3920-3923, Standards Track, Network Working Group – www.xmpp.org/rfcs/

**[Yan04]** Yankelovich, N., Walker, W., Roberts, P., Wessler, M., Kaplan, J., and Provino, J. Meeting central: making distributed meetings more effective, *Proc. ACM Conference on Computer Supported Cooperative Work (CSCW'04)*, Chicago, Illinois, USA, 2004, pp. 419-428.

**[Zig98]** Zigurs, I. and Buckland, B.K. A Theory of Task/Technology Fit and Group Support Systems Effectiveness. *MIS Quarterly*, Vol. 22, No. 3, Sep. 1998, pp. 313-334.

**[Zow05]** Zowghi D., and Coulin, C. Requirements Elicitation: A Survey of Techniques, Approaches, and Tools. in *Engineering and Managing Software Requirements*, Aurum, A., and Wohlin, C. (eds.), 2005, pp. 19-46.

# Appendix A:

# POST ELICITATION QUESTIONNAIRE, PILOT STUDY

---

***Questionnaire*** ***Name***

1. Imagine you are a developer and/or a customer who has to join a requirements elicitation workshop using eConference. Define how much you agree with the following statement:

*"Text-based synchronous communication is both sufficient and apt for the intent"*

☐ Strongly agree
☐ Somewhat agree
☐ Somewhat disagree
☐ Strongly disagree

because

4. Prioritize the features that you want to be implemented in the next eConference release?
☐ Presentation sharing
☐ Co-browsing
☐ Voting tool
☐ Hand drawing within the whiteboard

5. Given your first experience with the tool, imagine you have to use it again in another requirements workshop: what features not listed above would you like to be available?

6. Define your habit to use text chat. You chat:

☐ On a daily basis
☐ Often (once o more a week)
☐ Not very often (once or twice a month)
☐ Hardly ever

***Thank you for your time!***

# Appendix B:

# POST ELICITATION QUESTIONNAIRE, EMPIRICAL INVESTIGATION

---

## Elicitation Questionnaire

Full name: _____

**DEFINITIONS**

- **Decisions place content** = Meeting minute edited by the Scribe during the elicitation.
- **Item-based discussion** = The moderator selects what topic to discuss in the agenda, and chat screen will only show conversations related to it.
- **Other group** = If in the elicitation meeting you were in the client group, then 'other group' refers to the developers group; and vice versa.

**PLEASE ANSWER THE QUESTIONS BELOW**

Please indicate the role that you played in the requirements elicitation meeting using eConference.

☐ Client
☐ Developer

**A. Please indicate your agreement about the following statements about the usefulness of eConference in elicitation meetings. Use N/A (Not Applicable) if you have not used the indicated feature.**

| | Strongly agree | Somewhat agree | Somewhat disagree | Strongly disagree | N/A |
|---|---|---|---|---|---|
| 1. Having a complete log of both the chat and decisions place at the end of the meeting improved the effectiveness of the elicitation | ☐ | ☐ | ☐ | ☐ | ☐ |
| 2. Viewing the decisions place content being edited by the Scribe during the elicitation was useful in reaching a consensus | ☐ | ☐ | ☐ | ☐ | ☐ |
| 3. Viewing the decisions place content being edited by the Scribe during the elicitation was useful in clarifying ambiguities | ☐ | ☐ | ☐ | ☐ | ☐ |
| 4. The item-based discussion was useful in keeping the meeting on track | ☐ | ☐ | ☐ | ☐ | ☐ |
| 5. The item-based discussion facilitated the understanding of the current item | ☐ | ☐ | ☐ | ☐ | ☐ |
| 6. The private messages were useful to disagree with a member of your group, and not in front of the other participants | ☐ | ☐ | ☐ | ☐ | ☐ |
| 7. During elicitation, slow typists were left behind | ☐ | ☐ | ☐ | ☐ | ☐ |
| 8. The moderator should have had a way to control the number of people typing at once | ☐ | ☐ | ☐ | ☐ | ☐ |

**B. Please indicate your agreement about the following statements about the usefulness of eConference in requirements elicitation meetings, as compared to F2F interaction.**

| **In comparison to F2F meetings…** | **Strongly agree** | **Somewhat agree** | **Somewhat disagree** | **Strongly disagree** |
|---|---|---|---|---|
| 1. the eConference meeting offered increased opportunities to participate in the discussion | ☐ | ☐ | ☐ | ☐ |
| 2. the eConference meeting encouraged participants to more openly discuss conflicting issues with members of their own group | ☐ | ☐ | ☐ | ☐ |
| 3. the eConference meeting encouraged participants to more openly discuss conflicting issues with members of the other group | ☐ | ☐ | ☐ | ☐ |

**C. Please indicate your agreement about the following statements about the usefulness of eConference in requirements elicitation meetings, compared to F2F interaction. Provide an answer only for the kind of meeting you were involved in.**
**IF YOU EXPERIENCED BOTH KINDS OF MEETINGS, PROVIDE A SEPARATE ANSWER FOR EACH OF THEM.**

| | | **Strongly agree** | **Somewhat agree** | **Somewhat disagree** | **Strongly disagree** |
|---|---|---|---|---|---|
| 1. During the elicitation, decisions were made with a group consensus | F2F | ☐ | ☐ | ☐ | ☐ |
| | eConference | ☐ | ☐ | ☐ | ☐ |
| 2. During the elicitation, the discussion was too fast and information was missed | F2F | ☐ | ☐ | ☐ | ☐ |
| | eConference | ☐ | ☐ | ☐ | ☐ |

**D. IF YOU EXPERIENCED BOTH F2F and ECONFERENCE SESSIONS, please indicate which medium better supported**

| | F2F meeting | eConference mediated meeting | About the same |
|---|---|---|---|
| 1. consensus-based decisions | ☐ | ☐ | ☐ |
| 2. structured discussion of each agenda item at a time | ☐ | ☐ | ☐ |
| 3. documentation of decisions with respect to the issue at hand (agenda item) | ☐ | ☐ | ☐ |
| 4. moderator's activities during the meeting, such as traceability of decisions once reached; following the agenda items; documenting; and making the decision available to the group | ☐ | ☐ | ☐ |
| 5. in-depth discussions of agenda items | ☐ | ☐ | ☐ |
| 6. visibility of decisions once reached | ☐ | ☐ | ☐ |
| 7. articulation of ideas freely | ☐ | ☐ | ☐ |
| 8. spontaneous discussions | ☐ | ☐ | ☐ |
| 9. the ability to express basic ideas | ☐ | ☐ | ☐ |
| 10. the ability to express complex ideas | ☐ | ☐ | ☐ |
| 11. the control of etiquette and professionalism | ☐ | ☐ | ☐ |
| 12. the opportunity to gain familiarity with meeting participants | ☐ | ☐ | ☐ |
| 13. the ability to understand other participants' opinions | ☐ | ☐ | ☐ |
| 14. the sense of involvement in the meeting | ☐ | ☐ | ☐ |
| 15. motivation to participate in discussions | ☐ | ☐ | ☐ |

**E. Hand Raising**

a) Have you used the hand raising feature during the elicitation meeting?

☐ Yes
☐ No

If yes, say whether you found it useful and why.
If no, say why it was not used.

b) Do you think that the hand-raising feature should implement turn-based speaking the way it is realized now (i.e., as a social protocol, where turns are "advised", but everyone can still contribute to the discussion at the same time), or rather as a talking-stick (i.e. turns would be "mandatory" and only one person at a time could contribute to the discussion)?

☐ Social protocol
☐ Talking stick

Explain your answer.

**F. Other**

1. How often do you use text-based chat?

☐ Very often (On a daily basis)
☐ Often (once or more a week)
☐ Not very often (once or twice a month)
☐ Hardly ever

2. What functions would you like to see included in the next release? Please report them in order of relevance (first one most important, last one least important).

**G. Please provide any other comments if you wish**

***Thank you for your time!***

# Appendix C:

# POST NEGOTIATION QUESTIONNAIRE, EMPIRICAL INVESTIGATION

---

## Negotiation Questionnaire

Full name: _____

**This questionnaire is meant for the students who have only experienced the use of eConference for the negotiation meeting.**

Definitions:

- **Decisions place content** = Meeting minute edited by the Scribe during the eConference meeting.
- **Item-based discussion** = The moderator selects what topic to discuss in the agenda, and chat screen will only show conversations related to it.
- **Other group** = If in the negotiation meeting you were in the client group, then 'other group' refers to the developers group; and vice versa.

**PLEASE ANSWER THE QUESTIONS BELOW**

Please indicate the role that you played in the requirements negotiation meeting using eConference.

- ☐ Client
- ☐ Developer

**A. Please indicate your agreement about the following statements about the usefulness of eConference in <u>negotiation</u> meetings. Use N/A (Not Applicable) if you have not used the indicated feature. MULTIPLE ANSWERS ARE <u>NOT</u> ALLOWED.**

| | Strongly agree | Somewhat agree | Somewhat disagree | Strongly disagree | N/A |
|---|---|---|---|---|---|
| 1. Having a complete log of both the chat and decisions place at the end of the meeting improved the effectiveness of the negotiation | ☐ | ☐ | ☐ | ☐ | ☐ |
| 2. Viewing the decisions place content being edited by the Scribe during the negotiation was useful in reaching a consensus faster | ☐ | ☐ | ☐ | ☐ | ☐ |
| 3. Viewing the decisions place content being edited by the Scribe during the negotiation was useful in clarifying ambiguities | ☐ | ☐ | ☐ | ☐ | ☐ |
| 4. The item-based discussion was useful in keeping the negotiation meeting on track | ☐ | ☐ | ☐ | ☐ | ☐ |
| 5. The item-based discussion facilitated the understanding of the current item | ☐ | ☐ | ☐ | ☐ | ☐ |
| 6. The private messages were useful to disagree with a member of your group, and not in front of the other participants | ☐ | ☐ | ☐ | ☐ | ☐ |
| 7. During negotiation meeting, slow typists were left behind | ☐ | ☐ | ☐ | ☐ | ☐ |
| 8. The moderator should have had a way to control the number of people typing at once | ☐ | ☐ | ☐ | ☐ | ☐ |

**B. Please indicate your agreement about the following statements about the usefulness of eConference in requirements negotiation meetings, as compared to F2F interaction. MULTIPLE ANSWERS ARE NOT ALLOWED.**

| In comparison to F2F meetings… | Strongly agree | Somewhat agree | Somewhat disagree | Strongly disagree |
|---|---|---|---|---|
| 1. the eConference negotiation meeting offered increased opportunities to participate in the discussion | ☐ | ☐ | ☐ | ☐ |
| 2. the eConference negotiation meeting encouraged participants to more openly discuss conflicting issues with members of their own group | ☐ | ☐ | ☐ | ☐ |
| 3. the eConference negotiation meeting encouraged participants to more openly discuss conflicting issues with members of the other group | ☐ | ☐ | ☐ | ☐ |
| 4. the eConference negotiation meeting needs / requires a higher level of preparation | ☐ | ☐ | ☐ | ☐ |
| 5. the eConference negotiation meeting grants stakeholders a higher level of comfort (lower pressure felt, making comments without being afraid of intimidation) | ☐ | ☐ | ☐ | ☐ |

**C. Please indicate your agreement about the following statements about the usefulness of eConference in requirements negotiation meetings, compared to F2F interaction. PROVIDE A SEPARATE ANSWER FOR BOTH F2F AND ECONFERENCE.**

| | | Strongly agree | Somewhat agree | Somewhat disagree | Strongly disagree |
|---|---|---|---|---|---|
| 1. During the negotiation meeting, decisions were made with a group consensus | F2F | ☐ | ☐ | ☐ | ☐ |
| | eConference | ☐ | ☐ | ☐ | ☐ |
| 2. During the negotiation meeting, the discussion was too fast and information was missed | F2F | ☐ | ☐ | ☐ | ☐ |
| | eConference | ☐ | ☐ | ☐ | ☐ |

**D. Please indicate which medium better supported the <u>negotiation</u> meeting. MULTIPLE ANSWERS ARE <u>NOT</u> ALLOWED.**

| | F2F meeting | eConference mediated meeting | About the same |
|---|---|---|---|
| 1. consensus-based decisions | ☐ | ☐ | ☐ |
| 2. structured discussion of each agenda item at a time | ☐ | ☐ | ☐ |
| 3. documentation of decisions with respect to the issue at hand (agenda item) | ☐ | ☐ | ☐ |
| 4. moderator's activities during the meeting, such as traceability of decisions once reached; following the agenda items; documenting; and making the decision available to the group | ☐ | ☐ | ☐ |
| 5. in-depth discussions of agenda items | ☐ | ☐ | ☐ |
| 6. visibility of decisions once reached | ☐ | ☐ | ☐ |
| 7. keeping participant on task | ☐ | ☐ | ☐ |
| 8. reaching an agreement | ☐ | ☐ | ☐ |
| 9. prioritizing requirements | ☐ | ☐ | ☐ |
| 10. resolving conflicts | ☐ | ☐ | ☐ |
| 11. drawing conclusions | ☐ | ☐ | ☐ |
| 12. getting all the work done | ☐ | ☐ | ☐ |
| 13. articulation of ideas freely | | | |
| 14. spontaneous discussions | ☐ | ☐ | ☐ |
| 15. the ability to express basic ideas | ☐ | ☐ | ☐ |
| 16. the ability to express complex ideas | ☐ | ☐ | ☐ |
| 17. the control of etiquette and professionalism | ☐ | ☐ | ☐ |
| 18. the opportunity to gain familiarity with meeting participants | ☐ | ☐ | ☐ |
| 19. the ability to understand other participants' opinions | ☐ | ☐ | ☐ |
| 20. the sense of involvement in the meeting | ☐ | ☐ | ☐ |
| 21. motivation to participate in discussions | ☐ | ☐ | ☐ |
| 22. giving time to look up relevant information from existing documents | ☐ | ☐ | ☐ |

**E. Hand Raising**

1. Have you used the hand raising feature during the negotiation meeting?

   ☐ Yes
   ☐ No

   If yes, say whether you found it useful and why.
   If no, say why it was not used.

2. **ANSWER THIS QUESTION ONLY IF YOU HAVE NOT FILLED OUT THE ELICITATION QUESTIONNAIRE**
   Do you think that the hand-raising feature should implement turn-based speaking the way it is realized now (i.e., as a social protocol, where turns are "advised", but everyone can still contribute to the discussion at the same time), or rather as a talking-stick (i.e. turns would be "mandatory" and only one person at a time could contribute to the discussion)?

   ☐ Social protocol
   ☐ Talking stick

   Explain your answer.

F. **ANSWER THIS QUESTION ONLY IF YOU HAVE NOT FILLED OUT THE ELICITATION QUESTIONNAIRE**

   How often do you use text-based chat?
   ☐ Very often (On a daily basis)
   ☐ Often (once or more a week)
   ☐ Not very often (once or twice a month)
   ☐ Hardly ever

**G. Other**

3. What functions would you like to see included in the next release to support a requirements negotiation meeting better? Please report them in order of relevance (first one most important, last one least important).

4. Please provide any other comments if you wish

***Thank you for your time!***

# Appendix D:

# DESCRIPTION OF PROJECTS USED IN THE EMPIRICAL INVESTIGATION

---

**Project 1:** UVic Center for Scholastic Entertainment (UCSE) Educational Game.
**Project team:** Group 1 (clients), Group 2 (developers).
**Description**: An educational game designed to help students in grades 1 & 2 with Math, English and Problem-Solving skills. The game allows the users to practice their mathematical and English skills in a fun and entertaining manner. It is also a tool for teachers to evaluate the students. Teachers of the grades 1 and 2 students can use the game to track development in individual students and as a possible marking guide. Teachers can also export relevant data for creating spreadsheets and graphs of individual and group performance. The game is also available for home use so that parents of the students are able to see the progress of their child(ren).

**Project 2:** Supplies, Equipment, and Patient Tracking (SEPT) for St. Peter Hospital.
**Project team:** Group 2 (clients), Group 3 (developers).
**Description**: SEPT is a system to keep track of supplies, equipment and patients. The SEPT system integrates and improve upon the existing systems used by St. Peter's Hospital as follows: (1) By providing access to detailed medical records and automatic tracking of a patient's position in the hospital; (2) by providing an in depth medical

equipment interface with electronic sign out request and automatic location tracking capabilities; and (3) by maintaining supply stock information within the system to ensure that stock does not run out.

**Project 3:** Bus Tracking System.
**Project team:** Group 3 (clients), Group 4 (developers).
**Description**: The bus tracking system assists passengers with route planning, informs passengers of delayed busses, improves inter-bus transfers by informing bus drivers of connecting busses that are running behind schedule, helps transit management produce accurate schedules, and helps transit management allocate resources more efficiently.

**Project 4:** G4-consulting Groupwork System.
**Project team:** Group 4 (clients), Group 5 (developers).
**Description**: G4-consulting Groupwork System is collaborative development suite to expedite the software development cycle through improved inter-employee communication. The system provides secured, continuous communication, file- and code- sharing capabilities and communication archiving facilities.

**Project 5:** University of Vancouver Island Room Organization System.
**Project team:** Group 5 (clients), Group 6 (developers).
**Description**: Room Organization System is a centralized web based system used by faculty members and students to view room resources and book rooms around campus for various events.

**Project 6:** SysCal Shared Calendar.
**Project team:** Group 6 (clients), Group 1 (developers).
**Description**: SysCal shared calendar provides scheduling software for internal/external business uses and access to communications for arranging meetings/schedules of company employees. It also uploads external meeting schedule to a web server for viewing by people outside of the office.