THE UNIVERSITY OF CALGARY

Groupware Places for Lightweight Meetings

by

Yibo Sun

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

CALGARY, ALBERTA

SEPTEMBER 2010

THE UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled “Groupware Places for Lightweight Meetings” submitted by Yibo Sun in partial fulfillment of the requirements for the degree Master of Science.

Supervisor, Saul Greenberg
Department of Computer Science

Ehud Sharlin
Department of Computer Science

External Examiner, Warren Wilson
University of Calgary

Date

Abstract

The thesis described the design, implementation, and preliminary evaluation of Come Together (CT), a groupware system that supports light-weight interactions between intimate collaborators in small groups. As a foundation to its design, we begin by comparing light-weight and heavy-weight group working practices via a sociological framework, where we generated a list of basic design considerations. As well, we analyze and review a wide spectrum of groupware systems to see how they support or hinder light-weight formation and working practices of such groups. From all this, Come Together was created to meet three main design goals: supporting light-weight group formation and on-going maintenance; integrating people and artefacts by treating them equally; and support one's different levels of engagement in a group with different people and artefacts. Come Together is described by its features, and then by its technical aspects. Finally, a preliminary evaluation of Come Together was conducted to elicit participants' initial reactions. The positive reception indicates that our design generally matches our goals. However, participants' criticisms and suggestions also point out directions for future research.

Publications

Materials, ideas, and figures from this thesis have appeared previously or will appear in the following publications:

Sun, Y. and Greenberg, S. (2010) **Places for Light-weight Meetings: the Design of Come Together.** Proceedings of ACM Group'2010 Conference. November, ACM Press.

Sun, Y. and Greenberg, S. (2010) **Places for Light-weight Meetings: the Design of Come Together.** Report 2010-958-07, *Department of Computer Science* ¸ University of Calgary, May, 2010.

Acknowledgments

I would like to thank Saul Greenberg, my supervisor, for his tutelage. I am grateful to Stephen Voida, who contributed ideas in the early stage of my research. In addition, I would also like to acknowledge the help provided by all lab colleagues and other students who participated in my preliminary study of my software prototype.

Table of Contents

List of Tables

Code Listings

List of Figures

Chapter 1. **Introduction**

People are inherently social beings who communicate, work, play, socialize and interact with one another in a variety of ways. In this thesis, I investigate how computer technologies can support distance-separated *intimate collaborators* – small informal groups of ~2 to 10 people who have a real need and desire to stay connected with one another for a variety of purposes. A distributed collaboration system prototype is provided as a possible solution for *light-weight* formation and *light-weight* working practice of such groups.

1.1 Context and Background

People constantly come together in small social groups. While some gatherings happen via scheduling formal meetings, the vast majority are informal, casual, opportunistic, and somewhat ad hoc. The people involved are not strangers. Most know the other to various extents, and have a real desire – driven by social, work, play or a variety of other purposes – to interact with one another.

In the everyday co-located setting, such gatherings occur easily. People collect information about who is around and what others are doing. People use this awareness to move into frequent interactions that are often unplanned, opportunistic and brief (Kraut, Egido, & Galegher, 1988; Whittaker, Frohlich, & Daly-Jones, 1994). Most are seemingly mundane: greetings, social banter, and casual chit-chat. They also occur in casual settings: hallways, coffee rooms, over the printer, etc. Yet such casual interactions prove critical for small group effectiveness (Kraut et al., 1988; Whittaker et al., 1994). Discussions may include status updates, coordination information, and knowledge exchange. They also add to the social foundations of the group necessary for interpersonal solidarity. Discussions create opportunities: introductions to others, beginnings of new joint tasks, idea development, and so on. They sometimes move into more purposeful meetings, where its members move into the details of a task or goal.

Five elements contribute to the way casual interactions work so well in the co-located setting. *Informal awareness* - knowledge about presence, activity, and availability of each other - triggers meeting opportunities (Kraut et al., 1988). *Light weight meetings*, where people can easily act on this awareness to engage with others at negligible cost, means that even the briefest interaction can be done routinely. The proximity of *intimate collaborators* in the co-located setting makes both awareness and light-weight meetings desirable and easy (Kraut et al., 1988). People notice others who are close by, and can quickly move into conversation by making eye contact, moving closer to them, and speaking. In addition, people can – if desired – quickly *share artefacts* simply by bringing the artefacts to the attention of others and exploiting the tools ready to hand (e.g., documents, whiteboards, pens, etc.) (Whittaker et al., 1994). Finally, *membership and involvement* in a group is elastic. People affiliated with a group can not only come and go, but have various degrees of involvement with it.

The challenge is how to support this kind of casual interaction between distance separated people. The computer has altered this equation, where groupware can provide distant-separated colleagues with awareness and opportunities to move into computer-mediated interaction. In particular, *distributed groupware* is software that helps geographically-separated people communicate, collaborate and socialize via their computers. For groupware to work (either individually or in concert with other groupware), it must support the five basic elements above: how people form into social / work groups of intimate collaborators, how people participate in those groups at different levels of engagement, how people stay aware of others in those groups, how people use that awareness to move into light weight communications, and how people actually perform their joint work or social actions with others by sharing their artefacts.

Well-known groupware for asynchronous (non-real time) interaction includes email, social networking, internet forums, issue tracking systems, electronic calendars, project management, online documents and spreadsheets. Popular groupware for real-time interaction includes instant messaging, voice over IP, video conferencing, shared screen systems, and others. In spite of the myriads of groupware genres, most support only some of the five elements above, as each is specialized for particular kinds of exchanges. In practice, most people use a variety of groupware genres in tandem. The simplest but perhaps most effective example in regular use is instant messaging. People collect their intimate collaborators into buddy lists. They see the online status – an estimate of availability – of their buddies on this list. If the other person is present, they can move into a textual chat at the click of a button. Once engaged, they can exploit other available groupware tools to pursue actual work (e.g., email for information exchange). Indeed, several commercial systems bundle other tools into it, such as file transfer, audio / video calls for voice, shared sketchpads, and screen sharing (e.g., Microsoft NetMeetings and Skype).

Yet, the facilities provided by instant messaging are fairly rudimentary. For example, the only awareness information provided is whether the other person is present at their computer, which is approximated by capturing the idle time of keyboard / mouse activity of others. Other (mostly research) systems try to provide richer facilities. For example, Community Bar (McEwan & Greenberg, 2005) lets people gather into multiple electronic places, where it presents awareness information of people in those places via low frame rate video. Community Bar will be discussed in more detail in Chapter 2, as my own work builds upon it.

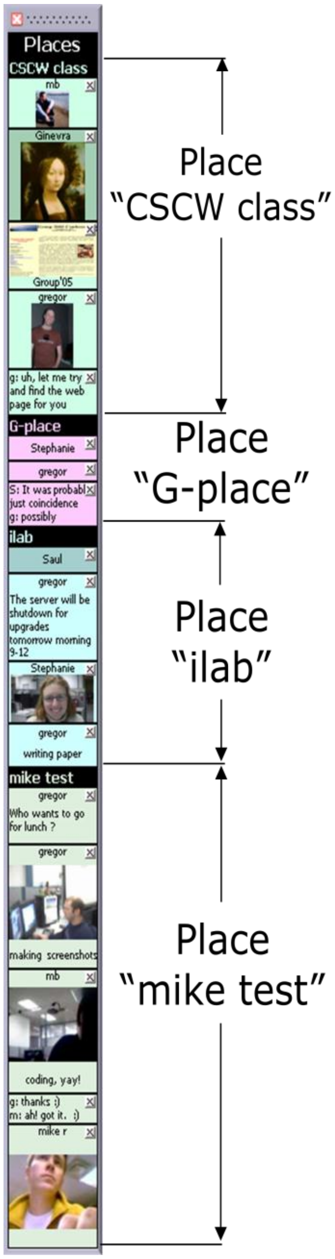
Shared artefacts are also essential components of socialization and collaboration. For example, a person may show family photos to a visiting friend, and co-workers meeting over coffee may work on a joint report. Whittaker et al. (1994) found that over half of all casual interactions in an office involved some form of document sharing. Consequently, groupware systems should maintain some form of *artefact awareness*: one person's up to the moment knowledge of the artefacts other group members are working with (Tee, Greenberg, & Gutwin, 2009). Similarly, groupware should facilitate opportunistic interactions with such artefacts. Tools do exist that do this, although most are somewhat specialized for particular types of artefact sharing. Collaborative authoring systems such as Orbit (Mansfield et al., 1997) emphasizes document sharing rather than collaboration around artefacts in general. Light-weight IM-styled/based systems such as Document Presence System (Morán, Favela, Enríquez, & Decouchant, 2001) and Project-View IM (PVIM) (Scupelli, Fussell, Kiesler, Quinones, & Kusbit, 2007) provide awareness information about the presence, availability and status of documents and files. Artifact Buddy (Greenberg, Voida, Stehr, & Tee, 2010) blends awareness of both people and artefacts by integrating "artifact buddies" into a commercial, unaltered instant messenger buddy list. The Community Bar is perhaps the most general, where it lets people post artefacts of interest to the group (e.g., photos, web pages, even their screens), where that artefact is immediately visible to others (McEwan & Greenberg, 2005). Any group member can move into interactions not only through text chatting, but by working over these shared artefacts. Developers can extend the types of artefacts via Community Bar's plug-in architecture (McEwan, Greenberg, Rounding, & Boyle, 2006).

1.2 Motivation

This project is primarily motivated by the successes and limitations of the Community Bar (McEwan, 2006; McEwan & Greenberg, 2005; McEwan et al., 2006). The Community Bar (Figure 1-1) was designed to maintain informal awareness and casual interaction of small communities, comprising ad-hoc and long-standing members. Community Bar's theoretical base is the Locales Framework (G. Fitzpatrick, 2003). To briefly summarize this framework, a *locale* consists of a *group* of people, a *site* where people are centered, and a number of *means* that people use for collaboration or socialization. Community Bar emulates a locale via a "Place". For example, four places are illustrated in Figure 1-1. Each Place provides the 'site' (a named container on a sidebar that people can subscribe to) and 'means' (via *media items* – small interactive windows – that allowed people to post tools. Figure 1-1 illustrates several media items including live video, text chatting, web page and photo sharing and screen-sharing. All are visible to group members via a sidebar display. Community Bar lets a user belong to multiple locales and have all her locales in view. i.e., each person sees their collection of their Places in the sidebar. Community Bar also provides different presentation levels for media items (lower vs. higher fidelity) to support a user's different degrees of involvements, as will be described in more detail in Chapter 3.

While seemingly well designed for lightweight group working practices, the Places as provided by Community Bar proved problematic. Romero et al. (2007) performed a field study of Community Bar use, and found that users rarely created multiple places. They did not create separate Places for long-lived meetings (where one would think it would be worth the effort), let alone Places for short-lived, ad-hoc encounters. They explain that Places as designed into the Community Bar are too heavy-weight to serve as a locale, and thus did not support how groups formed, evolved, and disappeared. Instead, group members used a single Place, where individuals mediated what was a locale by deciding what media items each would attend to.

The motivation of my work, then, is to revisit the design of a locales-based system. Ultimately, a successful system should support how groups can form in a very light-weight manner around a locale that offers the site and means for staying aware of others and their artefacts, and for moving into interaction with one another and with the group artefacts.



g: uh, let me try and find the web page for you

S: It was probabl just coincidence
g: possibly

The server will be shutdown for upgrades tomorrow morning 9-12

writing paper

Who wants to go for lunch ?

making screenshots

coding, yay!

g: thanks :)
m: ah! got it. :)

Figure 1-1 Community Bar (McEwan, 2006)

1.3 The Problem, Goals and Methods

The general problem is that our current technologies do not support the actual light-weight working practices of the groups that are not necessarily subject to strict restrictions, rigid controls, and formal procedures. (Romero et al., 2007). This leads to three specific inter-related goals described below.

1. *Design a paradigm for the light-weight formation and on-going maintenance of the distributed groups via a locale.* To achieve this goal, I will create and implement a distributed groupware system that allows people to easily create such locales and to allow

others to spontaneously join and leave the locale in a very light-weight manner. The intellectual underpinnings of this system will be motivated largely by the Locales Framework (G. Fitzpatrick, 2003).

2. *Support light-weight artefact incorporation, awareness and artefact-sharing by treating people and artefact equally as part of a locale.* Currently, some systems focus almost entirely on the artefacts that collaborators share (e.g., screen-sharing); others focus on collaborators who share artefacts (e.g., instant messaging). To achieve this goal, each locale will provide the site and means (Fitzpatrick, 2003) for artefact sharing. In particular, the lifecycle of a group will include both people and shared artefacts implemented as media items (McEwan, 2006) that can exist outside any locale and be easily brought into one or more locales. Our belief is that locales should evolve spontaneously as members come and go and as they share these artefacts.
3. *Let people adjust their involvement in a locale which in turn changes the awareness they receive for an individual view of the locale.* To achieve this, we incorporate the multiple presentation levels design of the Community Bar's media items to let people explore items of interest, and a similar multi-tiered method of letting people adjust the size and contents of places which hosts media items.

Specifically, our methodology is centered around a significant evolution and remix of key concepts of the Community Bar (McEwan, 2006) and Instant Messengers. While the Community Bar was designed around principles similar to the goals mentioned above, it failed to achieve all its goals. Its concept of 'Places', originally intended to support ad-hoc groups, proved too heavyweight. To improve upon places, we will incorporate and extend ideas from Instant Messaging into our design, as they have proven extremely successful at supporting idiosyncratic group formation (via buddy lists) and light-weight interactions (by single button presses).

While Community Bar did have shared artefacts that all could see, participants could only create them within the context of a single Place. That is, artefacts could not pre-exist a locale, nor could it exist outside its parent Place, nor could it be moved or replicated across places. However, the Community Bar developed the notion of a media item as a basic container of a shared artefact; this architectural idea is powerful and will be kept as the basis of our own design. We will recreate media items so they can be created outside of locales, where they can be moved in and out of locales as needed. Like the Community Bar, our new system will be a desktop application focusing primarily on real-time meetings for synchronous interactions rather than asynchronous interactions supported by web-based systems such as social networking services, although it will allow both to occur.

1.4 Contributions

1. **Development of design principles for lightweight group interactions**. The Locales Framework is used to analyze the difference between lightweight and heavyweight group working practices. A set of design principles is derived from understanding of people's needs for lightweight group interactions.
2. **Implementation of a groupware system—Come Together—to facilitate lightweight group interactions.** New interactive paradigms are designed and implemented to match the above principles.
3. **An open plug-in architecture to allow for customization and extension.** Stock media items' offering is limited, but the scope for customization and extension of the functionalities is broadened through customized media item plug-ins.
4. **Evaluation of Come Together.** A preliminary user study evaluates and critiques the design principles and the software user interfaces.

1.5 Thesis Overview

Chapter 2 provides background, motivation and intellectual underpinnings via a literature review and synthesis. In particular, it briefly summarizes the theoretical foundation of the Locales Framework, and develops a set of design principles that will guide my system design.

Chapter 3 reviews various genres of groupware systems, people-centric and artefact-centric, from a Locales perspective. In particular, I discuss how the design premises behind a genre support or hinder light-weight locale formation. I also discuss how these systems support light-weight customization of individual views of a locale for personal and artefact awareness with regard to the centre/periphery principle.

Chapter 4 introduces Come Together by four scenarios that progressively reveal its main features and building blocks.

Chapter 5 deconstructs CT into its components. It also explains (when needed) how these components meet the design goals as derived from the Locales Framework and the review of other groupware systems in the first three chapters.

Chapter 6 describes the system architecture and its technical implementation. It also describes the extensible nature, and how a third party developer can create plug-ins for it. Several example plug-ins are described.

Chapter 7 discusses the result of a preliminary user study. The validity of the design, to facilitate lightweight groups, is examined.

Chapter 8 concludes this thesis with a summary of the contributions and the prospect of future research work.

Chapter 2. **The Locales Framework**

This chapter frames the overall research goals described in Chapter 1. I describe the theoretical foundations behind the rationale of lightweight groupware design, and synthesise these as basic design considerations that will guide the design of my own system as described in later chapters. In the subsequent chapter, I will discuss how these design considerations have been met or hindered in particular groupware genres, where I pay special attention to the Community Bar.

The chapter begins with a summary of the Locales Framework (Fitzpatrick, 2003), a theoretical framework that was developed to help an analyst understand the nature of social and collaborative interactions. Amongst other things, we will see how the various foundations of the Locales Framework describe the way people easily form into short and long term groups, and how a *locale* is easily formed around the *site and means* people use for opportunistic group interaction. In later chapters, we will see how these two aspects drive my primary system design consideration for light-weight groups: i.e., that such a groupware system should be very light-weight in the way it supports group formation and the working practices typical of most collaboration. Other critical aspects of locales will be revealed, such as *mutuality* (i.e. the provision and perception of awareness), and *individual views* (i.e., how people maintain their own personal perspective of the collective activity). I will use these as secondary system design considerations: support for personal and artefact awareness, and easy customization of individual views of a locale.

Fitzpatrick (2003) developed the Locales Framework as a theoretic foundation that can be used by analysts to probe and describe the nature of social and collaborative interactions. That is, it is an applied descriptive theory of group interaction. The Locales Framework is not a prescriptive theory that predicts human behaviour. Nor is it a design theory that prescribes system design. Still, others have used it to inform system design

(Mansfield et al., 1997; McEwan, 2006; Rounding, 2004), where designers have used particular elements of the framework to guide the inclusion of groupware features to support people's real interactional needs according to that element. This chapter briefly reviews the major elements of the Locales Framework: a full description is in (Fitzpatrick, 2003). For each element, I emphasise factors that support a group's light-weight working practices and interactional needs, and then suggest how these can be applied as a system design rationale. The first element – locales foundations – will be discussed at length in comparison to the other elements, as it is the primary concept behind the Locales Framework.

The following sections and sub-sections will be structured as follows. First, I will summarize an aspect of Fitzpatrick's (2003) Locales framework. I will then state Greenberg, et al.'s (2000) corresponding groupware design heuristic that addresses that aspect. Subsequently, for each aspect, I will list a set of very specific design considerations that I crafted from the prior works; these design considerations are original and should be considered a contribution of this thesis.

2.1 Locales Foundations

The most fundamental element of the Locales Framework is a *locale*, which is formed by a *social world* (a group) using *sites* (a physical and/or virtual place) and *means* (resources) to cooperatively work on their collective goal.

2.1.1 Social Worlds, Sites and Means

Fitzpatrick (2003) defines a social world as a group of people with collective goal(s) and/or interest(s). The goal or interest does not have to be well-developed or completely knowable. Members of a social world perform their collaborative activities at a place, i.e. a *site* (a physical and/or virtual setting) furnished with *means* (resources and artefacts). Sites range considerably. A site is where people are engaged in their activity. It could be a dedicated physical room (e.g., a team room), a non-dedicated room used at the moment for group activity (e.g., a break-out room), or multiple physical rooms where activity takes place. A site may not even have a particular physical room in mind; it could be the way the social world meets opportunistically as they pursue their activities. Sites can also be virtual. They could be on a system that provides a metaphor of a physical place (e.g., a chat room), or could be the suite of tools that define where and how people meet for interaction (e.g., email, instant messengers). The *means* are the resources and artefacts provided or brought into the site. In a physical room, tools could include tables, desks, chairs, pen and paper, whiteboard, projector, etc. In a virtual room, these could be the digital resources packaged as part of the tool (e.g., file transfer capabilities in instant messengers). Artefacts include anything produced and worked on by the group, e.g., documents, sketches, and so on.

Finally, members, sites and means can overlap and/or be reused across multiple locales. For example, a social world can be mapped to multiple sites, which in turn defines multiple locales. A site can be mapped (or used by) multiple social worlds, which also defines multiple locales. A means can be shared across multiple sites, social worlds and locales. It is these overlapping relationships that define the global context of multiple locales. While complex, nuanced, and ever-shifting, this reflects what happens in the real world collaboration.

2.1.2 Centre/Periphery Principle

Fitzpatrick (2003) describes that *Members* in a *social world* act on a *centre/periphery* continuum. The membership of people within a social world is nuanced. It is not defined by a binary relationship (i.e., member or non-member). Instead, membership follows a centre-periphery continuum. At the *centre* is the context that holds the social world together, for example, the collective goal. Core members with high interest and/ or immediate activity in the goal are typically located at this centre. Yet other members have different levels of overall engagement or whose engagement fluctuates over time. Depending on one’s level of engagement, one’s membership may shift away from the centre towards the periphery. Thus membership at any moment of time is defined by a person’s engagement on the centre/periphery continuum. In other words, actively participating people are closer to the centre of the social world, while those less engaged are further away from the centre. For example, a person who is aware of a social world but not actively engaged in it could be considered to be on the outer periphery of that social world.

The centre-periphery continuum also defines the relationship between a social world to its site and means: more relevant sites and means are closer to the centre, while less relevant ones are closer to the periphery. For example, an artefact being worked on by group members that is an outcome of the collective goal would likely be at the centre, while a secondary artefact may be nearer to the periphery.

This principle of centre-periphery is a core concept in the Locales Framework. It not only concerns the relationship between social world members, sites and means, but also applies to the relationships of other entities that will be described shortly.

2.1.3 Light vs. Heavy-Weight Group Practices

The goal of my work is to support a group’s light-weight working practices. That is, I want to avoid the heavy-weight, sophisticated working protocols now required by a variety of groupware systems. Considering this goal from the perspective of the locales foundation, it is clear that locales must be very light-weight, i.e., it must be easy to become a member anywhere on the center-periphery continuum, and that the site and means must be readily available and easy to use. If done well, a flexible social world will emerge.

The locales framework does not use the terms light *vs.* heavy weight practices. Rather, I believe it is a consequence of how a social world is structured and supported. Thus the term ‘light-weight’ needs further elaboration within the context of locales. I define *light-weight group working practices* as:

> meeting practices that fit the serendipitous and/or immediate needs of an informal group, where the casual nature of their meetings require rapid meeting set-up, rapid involvement of its members, and a means to match the level of involvement to match the particular needs of its individual members.

This is admittedly a vague definition, as 'lightweight' vs. 'heavy-weight' practices are just two end points of a spectrum, and where many attributes may affect if a group practice is perceived as light vs. heavyweight. Table 2.1 attempts to distil some of the characteristics that may determine the spectrum between light vs. heavyweight meetings.

Properties	Heavyweight	Lightweight
Structure	Formal, hierarchical	Flexibly articulated and maintained or flat, idiosyncratic
Culture	formal	Casual and/or formative
Goal , focus, task, centre	contrived, knowable, stable	impromptu, unknowable, unstable
formation and termination	planned, created, terminated deliberately	spontaneous, improvised, formed and dissolved spontaneously
Goal , focus, task, centre	contrived, knowable, stable	impromptu, unknowable, unstable
Membership	stable, clearly defined, hard boundary, more centered, more cohesive	unstable, open, soft boundary, more peripheral, less cohesive
Member Roles	managed, assigned	Self-selected, socially negotiated upon, flexible, adaptive
Culture	formal	Casual and/or formative
Access and Security	Rigid, closed	Flexible, open

Table 2-1 Light vs. heavy-weight social world practices

Table 2-1 is somewhat of a caricature of heavy *vs.* light-weight practices. Still, it helps us understand the differences. First, a heavyweight social world typically follows a formal structure and formal culture. The organization is well planned, assembled, maintained, and dissolved in a deliberate manner. In contrast, lightweight groups are less planned and less prepared, usually because of its spontaneous nature. Thus a lightweight group's assembly is improvised; where it evolves and dissolves spontaneously over varying periods of time.

Second, a heavyweight group usually has well-formed goal(s) which serves as the centre of a locale and focus of tasks; the goal is relatively stable for a period of time. On the other hand, the spontaneous nature of a lightweight group corresponds to an impromptu goal or interest, which are usually not well developed or agreed upon at the outset. Thus the focus of activities, tasks and indeed even the 'centre' of the social world may not be clear. Even when defined, some or all of these may shift over time.

Third, heavyweight groups typically have clearly-defined membership, structure, and fixed (perhaps formally assigned) roles for its members. While members can have multiple levels of involvements, they are generally close to the centre and the organization is thus tighter. Hence, the organization tends to be stable and more cohesive. For similar reasons, a heavy-weight locale suggests that the group has a hard boundary – people are either ‘in’ or ‘out’, with a corresponding desire for security by controlling access to its site and means. In contrast, a lightweight group’s membership is highly variable, where some people may be at the core but others are far more peripheral. People may drift in and out, with quite different levels of engagement over time. People’s roles are not formally managed and usually more equal and adaptive. Accordingly, the organization is less cohesive and unstable, where security is traded off against easy access.

Fourth, the site and means of a heavy-weight group is usually planned and/or configured ahead of time. For a light-weight locale, the sites and means may be formed spontaneously and opportunistically along with the social world’s emerging impromptu goals and/or interest. I expect that the emergent goal or interest at the centre of the locale is initially formed by the social world’s core members, their artefacts, and their immediate focus of activities where – unlike a formal event – the site is improvised by convenience.

2.1.4 Design Considerations for Light-weight Group Practices

Greenberg, Fitzpatric, Gutwin, and Kaplan (2000) derived a set of groupware heuristics, each of which is from one aspect of the Locales Framework. Their general heuristic for locale foundations is:

- *Provide centers (locales) that collect people, artefacts and resources in relation to the central purpose of the social world*.

While a useful heuristic, it is still very general. In this sub-section, I elaborate on this heuristic with my own novel design considerations. In particular, I use the differences between light- and heavy-weight group practices as suggested by the locales foundations

(Section 2.1.3) to articulate several further design considerations for a groupware system supporting lightweight group practices.

D1. A person and/or group should be able to easily and spontaneously create, maintain and dissolve a locale.

D2. Because goals may be formed and altered over time, the system should not require the group to configure the site and means to satisfy a particular goal *a priori.*

D3. Membership should be flexible. People should be able to see what a locale is about without ‘joining’ it. Similarly, they should not require an invitation, or have to go through a chairperson or moderator. If they do become part of the social world of a locale, a person should be able to adjust his or her level of involvement from the center to the periphery.

D4. The group should follow its own social protocols and roles rather than a social protocol or role imposed by the system. That is, the system should support what people do naturally rather than demand they follow a prescribed set of rules and roles.

D5. Similarly, the trade-off between access and security should be maintained primarily through social means, where system control for access and/or security is added only if desired by the group.

In essence, the above design considerations envisage a groupware system that supports how people often form into a social world serendipitously and without much *a priori* planning, structure, membership, or organization. Such a system would let one or more people easily and serendipitously create an electronic locale. This locale, in turn, would create a site for the emerging social world to use, and provide various means so that people can easily pursue their interactions and goals as they involve. The site and means should be easily configurable to match the needs of the group over time. Membership within such a locale should correspond and fluctuate with one’s level of engagement rather than some forced groupware setting. Rather than imposing rules of group behaviour, the system should let people bring in and develop their own social norms; while the system could provide some scaffolding (especially if the group does develop into a more structured one), this should be optionally brought in by the social world as needed rather than imposed by the system.

2.2 Mutuality

The second element defined by Fitzpatrick's Locales Framework (2003) is *mutuality*: the mutual provision and perception of the awareness information of people, artefacts, and interactions. In a locale, people need to be aware of the state of the social world, the site, and the means, as this helps them maintain a sense of a shared place. It also helps people move into interactions and to work with others as the interactions proceed. Indeed, such awareness has been well-defined by others as a critical element of particular group interactions. For example, Dourish and Bellotti (1992) first defined *awareness* as "an understanding of the activities of others, which provides a context for your own activity" and argued that awareness is required to coordinate group activities and thus facilitate group progress. Moreover, Gutwin and Greenberg (Gutwin & Greenberg, 2002) described the crucial role of *workspace awareness* in how people understand other's actions while working together in a distributed setting. Kraut et. al. (1988) and Whittaker et. al. (1994) both described the importance of *informal awareness* in stimulating opportunistic and one-person initiated casual interactions. Alternatively, Endsley (1995) used the term "*situation awareness*" as the perception and understanding of what happens in the environment, which Bolstat et. al. (2005) also argue is an important determinant for teamwork performance. In addition, Tee (2007) brought in the important role of *artefact awareness*—one person's up-to-the-moment knowledge of the artefacts and tools that other people are using as they do their work.

Mutuality teases awareness into two parts: provision of the information (about the members, site and means in a locale) to others, and perception that is received by others. In turn, the centre-periphery principle affects the relationship between the provision and perception of awareness projection. A person or artefact at the centre will likely provide more information about itself to the rest of the world, while those at the periphery will provide less. Similarly, people drawn to this centre will perceive this information at greater salience and fidelity vs. those at the periphery. Thus we can expect the provision and perception of awareness information to degrade selectively according to how people and artefacts move from the center to the periphery.

2.2.1 Design Considerations for Light-weight Group Practices

Greenberg et al. (2000) provided a groupware heuristic for mutuality support.

- *Provide awareness (mutuality) within locales that helps people maintain a sense of shared place and that keeps them informed about shared activity. Mutuality includes one person's awareness of others, the artefacts comprising the locale, where things are located, and how things are changing*

However, this provision and perception of awareness must also reflect the center-periphery relationship. In particular, I propose the following considerations:

D6. The system should capture awareness information in a manner that reflects that person's engagement with the group (i.e., center/periphery), where the person could also choose how to view that information.

D7. The capture and presentation of awareness information should be adjusted to reflect a person's dynamic movement across the center/periphery continuum.

Heavy-weight groupware systems tend to assume that people are at the center, i.e., that all are intensely involved and thus require the system to capture a large amount of the provided awareness information, and to display it in a way that is highly perceptive. In comparison, a light-weight locale assumes that while some members may be at the center, others will be at the periphery. Thus the system should adjust its demands for awareness provision and how it displays it to others accordingly. Those less engaged in the locale only need to maintain a peripheral level of awareness, and similarly need to provide only peripheral information to others. This means that provision and perception must be individually maintained, rather than uniformly applied across all group members. Furthermore, the system should adjust this balance to reflect how entities move between the centre and the periphery over time.

2.3 Individual Views

Individual views, described by Fitzpatrick (2003) as the third element of the Locales Framework, are the different perspectives people hold of a locale. A social world is not homogeneous since people are individual beings, bringing their own perspectives and needs into a locale. The activities of a locale are not seen from a uniform perspective of the group. Rather, they are seen from the individual perspectives of contributing group members. The variety of different interests, focuses, and levels of involvement among group members result in these different perspectives—individual views—group members hold of a particular locale. Further, an individual view is highly variable due to moment-to-moment shifts of interest, and the changes of the involvement level of an individual in a locale. A person can also be in multiple locales, and shift their focus from one locale to another while maintaining varying degrees of involvement in all of them. Mutuality, discussed in the previous section, is closely coupled with individual views, because individual views are achieved through the perception of information about entities in a locale by individual group members.

In terms of the centre-periphery principle, the individual view describes a change of perspective. Instead of considering the locale as having people, activities, and means across the centre/periphery continuum, we consider instead the person's view point or "view set" of all locales he or she is involved in. That view has the individual at the centre, where locales and their contents are viewable along the centre/periphery continuum depending on that individual's interest and engagements. Looking at a locale of interest and the world of locales from the perspective of an individual produces an individual view of the locale and a view set—an aggregation of individual views of all the interesting locales respectively.

2.3.1 Design Considerations for Light-weight Group Practices

The groupware heuristic for individual views from (Greenberg, Saul et al., 2000) is:

- *Allow individual views so one can view a locale or aggregate multiple locales as they relate to one's responsibilities, activities, and interests. A particular person* *should be able to view locales from his or her particular perspective and in a way that reflects their degree of focus and participation.*

In conjunction with the mutuality design considerations, a person's individual view of a locale should be through the customized perception of awareness information. For this aspect, I suggest the following three design considerations:

D8. The system should allow an individual to form their view of a locale through the aggregation of received awareness information of each entity—people, shared artefacts and interactions—all at different levels of attention and engagement.

D9. An individual view should be updated to match one's interest and engagement in a locale over time, e.g., where some entities in the locale move closer to the person and others fade out to the periphery.

D10. Similarly, the view set of all locales one is involved in should change to reflect the shifts of one's interest and engagement, e.g., from some specific locales to others.

A heavy-weight locale assigns a person with a fixed individual view. One's role and activities are usually imposed, based on the static needs. As discussed earlier, this centered involvement implies a stable focus by an individual (the degree the locale view is attended to), and stable participation (the level of engagement in the locale). In contrast, people have self-selected, adaptive roles and needs in a lightweight locale. They should be able to adjust the strength of awareness provided to others and received from others based on the need of their activities in the locale at any point of time. They may choose to concentrate on certain part of the locale at one moment and shift to another part at another moment. They may also work closely in one particular locale at one moment and move out to periphery at another moment. An individual's shift of interest within a locale or across locales can be volatile and the system should support it fluidly.

2.4 Interaction Trajectory

Interaction trajectory, the fourth element stated by Fitzpatrick in the Locales Framework (2003), addresses the temporal 'movement' of an object. This trajectory traces all interactions in a locale along time: past, present and future. A locale as a whole may have a variety of interaction trajectories arising from the individual perspective of its actors: e.g., trajectory of people, trajectory of events, trajectory of actions, and trajectory of artefacts.. Interaction trajectory is important because of the situated nature of work: it is the situational temporal context that provides situated conditions for action during a locale's evolution.

The centre-periphery principle can relate entities in a trajectory to each other and one trajectory to another. Within a trajectory, one or more entities act as the centre pulling other related entities around. In terms of inter-trajectory relationship, one particular trajectory of interest can be the main thread of a locale's activities, with other trajectories on the centre-periphery continuum.

2.4.1 Design Considerations in Light-weight Group Practices

This thesis does not explore interaction trajectories in depth. However, system design should follow the general heuristic given by (Greenberg, Saul et al., 2000):

- *Allow people to manage and stay aware of their evolving interactions over time. This includes a group's control over past, present and future aspects of routine and non-routine work; how people coordinate and negotiate plans and activities over time; how people leverage past experiences; how breakdowns are noticed and repaired; and how processes are supported*

I concentrate mostly on providing people with an appropriate state of awareness, where they can understand the current state of the social world. I do not address past experiences or planning, but these could be added as part of future work.

2.5 Civic Structure

Civic structure, discussed in (Fitzpatrick, 2003), addresses inter-locale relationships on the macro level. This global context consists of multiple locales, inter-dependent on and inter-acting with one another. Civic structure concerns the mutuality of how a locale is presented to the public sphere and perceived by other locales. People need to navigate through the public sphere and be aware of the emergence and dissolution of locales.

The centre-periphery principle can also be applied in civic structure. At the centre of the global context, one or more locales may be of primary interest. Other locales may be situated around this centre at different distances. Some may have closer relationship to the one or ones at the centre, while others are less relevant.

2.5.1 Design Considerations

Like trajectory, civic structure is also not part of the focus of my thesis. Still, it is worth acknowledging, for it should be considered in future work. The design heuristic suggested by (Greenberg, Saul et al., 2000) is:

- *Provide a way to organize and relate locales to one another (civic structures). Locales are rarely independent of one another: people need a way to structure the locales in a meaningful way, to find their way between locales, to create new locales, and to remove old ones.*

In my opinion, in a heavy-weight civic structure, locales are in a rigid structure that has to be articulated and maintained, perhaps related to each other hierarchically where small locales are contained in a big locale. Emergence of new social worlds and new locales are often a result of discovery of other locales, people and resources. As a goal or interest arises out of the activities in one locale, a new locale around this goal or interest may branch off as a sibling or a contained sub-locale in the hierarchy. Accordingly, locales can be traversed through the hierarchy of containments. Thus trajectories of contained locales are tightly-coupled with, and influenced by those of containing locales. On the other hand, a light-weight civic structure is rather flexible. It can be flat or idiosyncratic. Even if it is hierarchical, the relationships between locales can be easily formed and changed—not imposed and circumscribed by specific organizational rules. However, the system should not dictate how the bonds between locales are formed and represented. Instead, it should allow people to manage the inter-locale relationships in their own manner best for their own practises.

In later chapters, I describe Come Together’s approach to bond locales into civic structure. However, it is simplistic, and really just a placeholder for future works.

2.6 Summary

In this chapter, I provided an overview of the Locales Framework and each of its elements. The centre-periphery principle emerged as a core concept to interpret the dynamics of locales. I was particularly interested in viewing the Locales Framework from the perspective of light-weight working practices and interactional needs of a group, and how these should be considered in groupware system design.

First, the locale foundations suggests that people should easily form into a social world, where they create a locale providing the site and means for interactions, and use their natural social norms and protocols to mediate their membership and interactions. Second, mutuality and individual views together suggest how information is gathered and displayed to individuals via custom and personal views of a locale, its members, and its artefacts. Finally, interaction trajectories and civic structure – while not fully addressed in this work – are aspects that complete one’s view of a locale over time and across locales.

My groupware system design is based on the design considerations presented above.

Chapter 3. **Existing Groupware for Lightweight Groups**

The previous chapter summarized the Locales Framework, where I used that framework to suggest basic design considerations for groupware supporting light-weight groups. In this chapter, I briefly review various genres of groupware systems from a Locales perspective. Most of these systems (except instant messengers, internet forums, blogging sites, and social networking sites) are academic system with limited deployment. However, they are suggestive of future groupware technologies. In particular, I discuss how the design premises behind a genre support or hinder light-weight group formation via the two primary design criteria, i.e., easy formation of short and long term groups, and encouraging opportunistic group interaction via provision of an adequate site and means. We will see that while some support is there, it is typically provided in a heavy-weight manner that limits their actual use. I also discuss how these systems support light-weight customization of individual views of a locale for personal and artefact awareness with regard to the centre/periphery principle. I later focus on the Community Bar (introduced in Chapter 1) in detail. Community Bar deserves this special treatment, because McEwan (2006) designed it around the Locales Framework. However, a study of its actual use (Romero et al., 2007) disclosed that its design did not support some of the practices suggested by the Locales Framework. I close by describing how these groupware genres influenced my design of Come Together, a new groupware system to be introduced in later chapters. In particular, Come Together's design revisits aspects suggested by the Locales Framework that, in retrospect, were not well met with the Community Bar.

3.1 Groupware for Interpersonal interactions

Perhaps the most fundamental aspect of human social behaviour is the conversation that occurs as part of interpersonal group interaction. This can be viewed as a four-part process.

1. Discovering opportunities for conversation.
2. Moving into the conversation.
3. The conversation itself.
4. Disengaging from the conversation, perhaps to rejoin it later (steps 1 & 2)

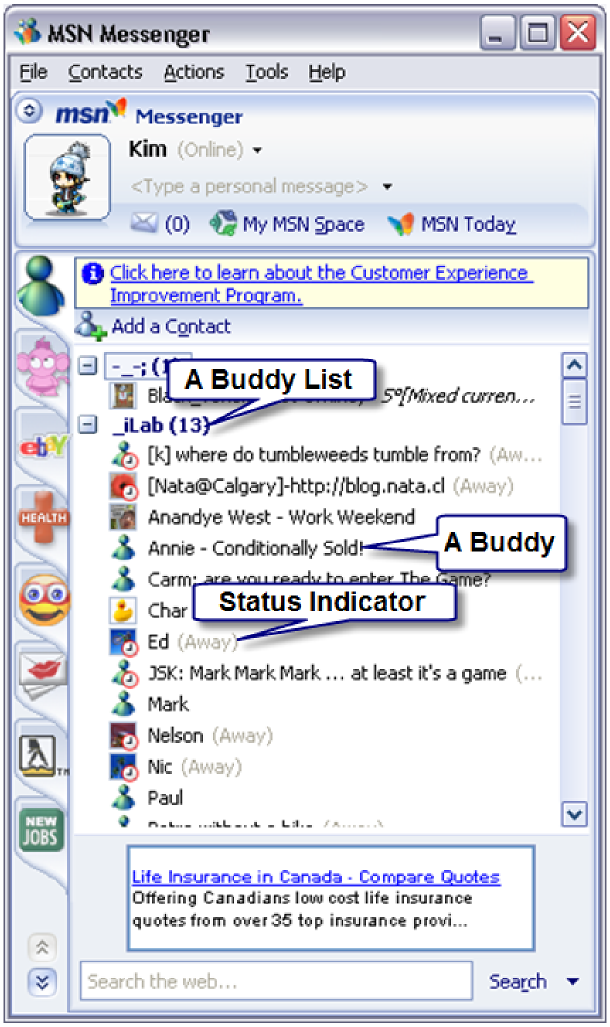
Discovering opportunities for conversation is easy when people are present in a co-located environment, as their presence, activities, and availability are always available at each other’s periphery. As discussed in chapter 1, this information produces many opportunities for people to move into casual, light-weight meetings and interactions. Moving into conversation happens via a greeting process: how people look at each other, verbal signals, moving closer to one another. The conversation itself is similarly easy: people naturally engage in talk augmented by body language, gaze awareness, gestures, and so on. Disengaging and potentially rejoining that conversation is just a matter of moving away, and then perhaps re-entering it later. Indeed, the act of discovering opportunities, moving, holding and disengaging from the conversation is so easy, routine and uneventful, that the process is part of people’s tacit knowledge, and is almost unnoticeable to them (Kraut et al., 1988; Whittaker et al., 1994).

The situation is quite different when people are separated by even relatively small distances. The disappearance of peripheral awareness of others causes a sharp decline or even disappearance of the opportunities for social and collaborative interactions. Moving into conversation – unless mediated by technology – in most situations only occur after planning (e.g., to arrange a meeting time), and synchronizing what system to use, when to use it, and so on. Disengaging from conversation is abrupt; one is either in it, or out. There is no gradual dis-engagement. While this strict process is sometimes reasonable for formal meetings, this is far too heavy-weight for casual interactions.

Groupware system designers were not blind to this, and thus created a variety of facilities to mitigate how physical separation hinders moving into conversation. The typical strategy is to increase people's virtual proximity to one another, and use that to create opportunities for conversation.

3.1.1 Instant Messengers

Instant messengers are perhaps the most successful and commonly used groupware genre that supports both moving into conversation and the conversation itself. Many major software vendors support their particular instant messenger system and infrastructure, e.g., MSN messenger, ICQ, Yahoo messenger, and Google Talk. People commonly use such instant messengers for coordination, collaborative work, and social needs (Cameron & Webster, 2005; Fussell, Kiesler, Setlock, & Scupelli, 2004; Grinter & Palen, 2002; Herbsleb, Atkins, Boyer, Handel, & Finholt, 2002; Isaacs, Walendowski, Whittaker, Schiano, & Kamm, 2002; Nardi, Whittaker, & Bradner, 2000; Patterson et al., 2008; Quan-Haase, 2008; Vos, ter Hofte, & de Poot, 2004).



Kim (Online)

<Type a personal message>

(0) My MSN Space MSN Today

Click here to learn about the Customer Experience Improvement Program.

Add a Contact

-_-; (1

Bl... 5°[Mixed curren...

_iLab (13)

[k] where do tumbleweeds tumble from? (Aw...

[Nata@Calgary]-http://blog.nata.cl (Away)

Anandye West - Work Weekend

Annie - Conditionally Sold!

Carm: are you ready to enter The Game?

Char

Ed (Away)

JSK: Mark Mark Mark ... at least it's a game (...

Mark

Nelson (Away)

Nic (Away)

Paul

Life Insurance in Canada - Compare Quotes

Offering Canadians low cost life insurance quotes from over 35 top insurance provi...

Search the web...

Figure 3-1 a MSN messenger (Tee, 2007)

Instant messengers (See an example in Figure 3-1) support all four stages mentioned earlier. Two features support Stage 1: discovering opportunities via buddy list and activity status. First, it displays a *buddy list*: a personalized list of the names (or nicknames) of other instant messenger users – co-workers, friends, family, etc.

– that collects the people that person is interested in. Second, each person's presentation is accompanied with that person's *activity status* – online, offline, busy, away, and so on – to indicate their presence and likelihood of availability. Activity status indicators are sometimes augmented by alerts that accentuate when people come online, i.e., the equivalent of them coming into view. Next, instant messengers support the 2nd 'moving into conversation' stage via easily-initiated text chat. By double-clicking on an available contact, a chat session is started immediately for two users to interact with each other. The conversation itself is the typing that follows (many instant messengers now augment text chat with a voice channel, typically triggered after a short textual conversation). Stage 3, disengaging from a conversation, is often just a matter of destroying the chat window. Yet, stage 4, re-engaging, is easy; one just clicks on one's buddy again. Furthermore, many IMs save a transcript of the previous chat conversation, where that transcript is displayed upon re-engagement.

Let us reconsider instant messengers from the Locales Framework perspective. On one hand, a chat session *could* signify the formation of a short-term locale. The viewing of opportunities (via people's activity state), the desire to communicate (via the double-click), and its acceptance forms a social world of two. The site becomes the chat window; and the means are the textual conversation, plus other tools supplied by the instant messenger (e.g., audio, video, screen-sharing, file-transferring). On the other hand, this is too simplistic a view, for this kind of locale is extremely transient. As well, locales are not a 1:1 map onto a textual conversation. Many locales could actually contain multiple IM conversations (which are seen by the IM system as separate), and conversely a single IM conversational thread can contain talk from multiple locales. IM also excludes communications from other sources that comprise the locale. Finally, instant messenger systems keep no memory of this locale except (in some systems) by logging and concatenation text conversations for a period of time. Neither is the buddy list a locale, for each person's buddy list may differ considerably from others. While the overlap between two people's list may describe people in a particular locale, this subset is a mental construct rather than one explicitly maintained by the system. That is, the buddy list collects one's intimate collaborators and friends, but has no notion of groups.

Activity indicators are excellent at fulfilling the basic socialization needs of loosely coupled social connections (e.g., John's come on line, which reminds me that I want to talk with him about something). While powerful, these indicators are not tracked as part of a locale. They are limited as they only approximate 'availability' from 'activity' (i.e., online or offline; idle time intervals). There is also no awareness of the activity a person is actually involved in (e.g., an activity pertinent to a particular locale). In summary, the instant messenger genre is best considered a 'means' that can be used within a locale.

We are not arguing that instant messengers are fundamentally flawed, for that would belie their very success. Indeed, instant messengers embody one of the light-weight qualities we promote: enabling users to move easily from contacts' awareness to light-weight interactions. Rather, we argue that they could be improved even more if they treated a locale as a first class entity. Currently, they only support a transient locale. Nor does the buddy list know about locale membership. Nor is there any support for the principle of centre/periphery within a locale, e.g., by providing selective awareness information of particular buddies based on a person's interest in a particular locale.

3.1.2 Media Spaces

The genre of media spaces create a virtual environment by connecting physically distributed spaces through high-speed, real-time, audiovisual links. Media spaces differ considerably from video-conferencing rooms (Okada, Maeda, Ichikawaa, & Matsushita, 1994). First, media spaces typically connect personal spaces, such as people's offices (Sara A. Bly, Harrison, & Irwin, 1993; Dourish & Sara Bly, 1992), or communal spaces like coffee rooms and other public spaces (Jancke, Venolia, Grudin, Cadiz, & Gupta, 2001). That is, these spaces are usually owned and occupied by its participants, rather than a specialized space configured for a video-based meeting. Second, the video connection is either always-on, or permit some kind of 'glance' facility where one can walk a virtual hallway to see who is around (Gaver et al., 1992; Root, 1988; Tang & Rua, 1994). Finally, conversations on these spaces are normally serendipitous or initiated by one person, rather than by an arranged meeting time.

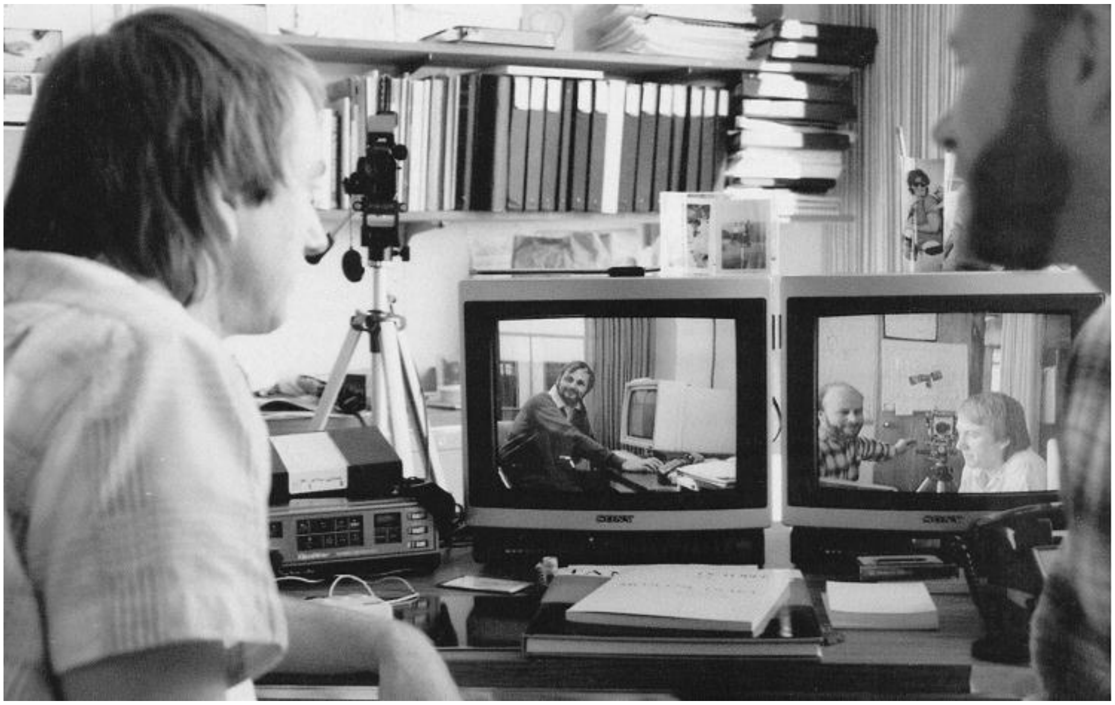


Figure 3-2 Media space connecting offices (Bly et al., 1993)

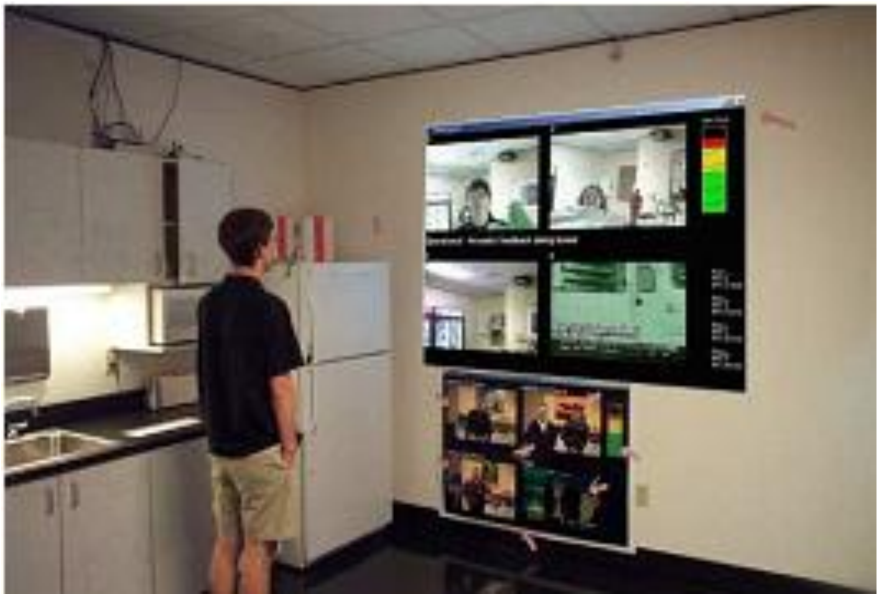


Figure 3-3 Virtual Kitchen (Jancke et al., 2001)

The major argument for media spaces is that they provide high fidelity awareness of what is going on in the other space by creating adjacent physical spaces via video, i.e., they ‘re-establish’ a form of physical proximity and, in turn, these provide awareness that creates opportunities for interaction and lightweight conversations. As in real life, people feel physically connected. In an always-on video link, conversations begin by (perhaps) trying to establish eye contact, making sounds, and then by talking. No interface mechanisms are required (although some systems do provide end-user control over video fidelity and audio capture, for privacy reasons). The kind of awareness (and how people consider it) varies with the media space configuration. For example, a media space connecting personal offices (e.g., through webcams and normal-sized monitors) are somewhat akin to people cohabitating an open office, where each person is quite aware of what the other is doing, and may regularly have brief casual conversations with them about ongoing activities of interest. A media space connecting two communal spaces through a large screen such as wall displays joining two coffee rooms gives people a sense that they are in one large communal space.

Media spaces do have limitations. Interaction beyond the visual and auditory are usually very limited, unless augmented by other software tools (Tee, 2007). A person's view of the other space is usually fixed and imposed, unless remote camera control is provided (Gaver, Smets, & Overbeeke, 1995; Kuzuoka, Kosuge, & Tanaka, 1994; Nakanishi, Murakami, & Kato, 2009; Ranjan, Birnholtz, & Balakrishnan, 2006, 2007). Yet such control introduces privacy issues. Perceptions of what the other can see or hear may be inaccurate, which again affects the delicate balance of privacy and awareness.

From a locales perspective, a media space equates a locale with a site. By bringing two or more sites together in an always-on video/audio connection, the premise is that locale activity is supported. Center and periphery is bounded by what the camera and microphone transmits and displays, and how people enter that visual/auditory field of view. The means of a site are usually limited to what can be displayed and heard across the channel. Media spaces seem to have fared better when used to connect the personal offices of close collaborators (Borning & Travers, 1991; Dourish & Sara Bly, 1992; Gaver et al., 1992; Root, 1988; Tang & Rua, 1994; Watabe, Sakata, Maeno, Fukuoka, & Ohmori, 1990) *vs.* two communal sites (Fish, Kraut, & Chalfonte, 1990; Jancke et al., 2001). The likely reason is that the shared office space is a reasonable estimate of a true locale, whereas the shared communal space is not. As with IM chats, a single media space may not directly map onto a single locale. The media space is just a connection, and as such it is blind to the social world's purpose. That is, the place may not actually represent a locale, or that multiple locales may occur using that site.

3.1.3 Chat rooms, MUDs, MOOs, and CVEs

Another groupware genre covers chat rooms, MUDs, MOOs, and Collaborative Virtual Environments (CVE) (Fahlén, Brown, Ståhl, & Carlsson, 1993; Frécon & Nöu, 1998; Greenhalgh & Benford, 1995; Nakanishi, Yoshida, Nishimura, & Ishida, 1996). While they differ radically in how they present interactions, all support multiple and persistent real-time interactions by communities of people via a 'room' metaphor. That is, they approximate multiple, longer-standing locales.

Chat rooms contain multiple topic-specific 'rooms'. People can enter a room, and then converse with anyone who happens to be there (ostensibly about the topic, but they are not restricted from talking about anything). MUDs (or multi-user dungeon) are somewhat similar, except that people can control what is happening in the room, e.g., by moving around it, by picking up objects in the room, and so on. MOOs are object-oriented MUDs; of relevance is that most modern MOOs comprise multi-media objects (perhaps including video). Finally, a CVE is a 3D environment, where people present themselves as physical avatars within a simulated world, and can interact with the world and the people (avatars) inhabiting it. CVEs are now extremely common in multi-player online games. In all these systems, people can either choose to join one of the many pre-constructed rooms, or they can (usually) create a new room for others to join. In most of them, people present themselves anonymously. That is, they use masking pseudonyms and avatars rather than their real-life identity.

These room-like systems are somewhat akin to multiple, longer-lasting locales. We can equate a locale with a particular dedicated room, although unlike media spaces these are completely virtual sites that mimic (with varying levels of fidelity) various physical characteristics and affordances of the space. In terms of light-weight groups, these room-based systems have limitations. A user has to explicitly create a room and configure it, something that may not even be permitted without approval from an administrator. Rooms are hard containers, i.e., they do not support center-periphery involvement. That is, one has to enter a room before knowing what is inside or gaining awareness of what is going on. A room – especially if portrayed as a very large space such as within a CVE – may actually contain many locales. While most room-based systems do allow people to configure private messages and message filtering functions (and thus identify a type of locale within the room), this is heavy-weight. Finally, artefact manipulation is usually limited to the virtual objects in a room rather than digital artefacts that a person may want to bring in a room (Curtis, Dixon, Frederick, & Nichols, 1995). All in all, while these systems do resemble some locale aspects, they are too heavy-weight at locale formation and are not flexible enough for light-weight, casual interactions.

3.1.4 Asynchronous web-based groupware systems

Many other commonly used web-based groupware systems support asynchronous interactions between people rather than real-time meetings. They include Internet forums, blogging and social networking services. Some of them provide manual or automatic means to update personal awareness information. Locales also exist in such applications: a thread in a discussion board, a blogging page, a list of followers, a network of interconnected people. Although, some (such as a Google Buzz) are formed in a very light-weight manner, most are long standing and heavy-weight locales. However, Most are designed for publication or exchange of opinions or information which are totally different purposes from people’s communications in meetings.

Facebook is the most popular social networking service on the web. It allows users to set up a personal web page with profiles, photos, and personal posts. It differs from a conventional blogging service in that it lets people easily link to others’ pages and form a social network on the web site. One can update personal awareness information via profiles, photos, or posts. Each contact of the user will see the update made on their own personal pages (which aggregate all updates from all contacts). From a locale perspective, a user and all the people in the contact list implicitly form a locale. The user’s personal page provides an individual view of the social networking locale of hers. Besides this implicit, permanent locale consisting of all personal contacts, Facebook also allow users to join other types of explicitly pre-configured locales such as "groups" or "networks". A "Group" allows a number of people to come together for a certain purpose. It is an explicit list of people rather than the implicit list of all contacts. A "Network" is a pre-constructed, long-standing locale such as a school or a workplace. These locales are also heavy-weight to create and join. All in all, social networking sites are primarily designed for people to stay connected with each other using asynchronous updates (*vs.* our emphasis on real time meetings), although they have the capability to allow online users to communicate synchronously via means such as textual chat.

Location-based social networking services are also gaining popularity. These websites allow users to update their geo-location information which can be seen by their contacts or even the public to trigger interaction opportunities. One can see who are nearby and what they are doing. Such systems do support very light-weight engagement in a locale. For example, in Google Buzz, a user can click on a "buzz" message posted by another nearby user and start a discussion in a forum thread style. The discussion thread following a "buzz" is essentially a locale. However, they are still like other social networking services in that they do not provide a real-time meeting site and sufficient means for personal awareness (e.g. live-video) and artefact sharing (e.g. synchronous interactions on a shared artefact).

3.2 Groupware for Artefact sharing

In previous chapters, we have discussed the equally important role of shared artefacts. For example, the whiteboards and sketches used for discussion in a meeting are often the focus of people's attention and as important as people in the meeting. Most genres of systems reviewed in the last section have people-centered designs; while some do allow artefacts to be brought into them (e.g., file transfer in IM), they do not support artefact collaboration particularly well.

Yet collaboration is not only about people being together. People bring artefacts into collaboration to share with others. Using language from the Locales Framework, groupware systems should provide sufficient means to help people bring artefacts easily into a locale without much overhead for configuration, and then facilitate interactions around shared artefacts. Groupware systems should also, on a centre-periphery continuum, maintain *artefact awareness*, defined by (Tee, 2007) as “one person’s up to the moment knowledge of the artefacts that other group members are working with”. Tee summarized why artefact awareness is valuable. First, people need to monitor each other’s activities on joint work for coordination. Next, artefact awareness can trigger interest in others’ activities. Third, knowledge of artefacts undertaken by others can be used to determine their availability. Last but not least, artefact awareness creates opportunities to start conversations around artefacts and then transition to collaboration around them. In this section, we sample several groupware systems designed primarily around artefact sharing.

A *shared workspace* is a visual space that all participants can see. To coordinate real time collaborative work, people in a distributed group need such a shared workspace if they are to see each other’s activities. This is why many groupware systems are realized as a shared workspace vs. an interpersonal conversational space, e.g., collaborative document editors, shared screen and window systems, and drawing whiteboards.

Implementing a shared workspace can come with considerable non-trivial technical challenges, particularly in management of concurrency control. The problem is that inconsistencies can arise if two or more people try to alter the same artefact. There are a variety of ways to solve this, although these tend to be specific to the type of shared workspace being considered. Many collaborative authoring systems solve this via coarse-grained access control, e.g., managing a document as multiple parts and allowing a person exclusive access to one of these parts (thus two people cannot change the same part). Others do more fine-grained acces, e.g., MACE (Newman-Wolfe & Pelimuhandiram, 1991), SEPIA (Haake & Wilson, 1992), SASSE (Baecker, Nastos, Posner, & Mawby, 1993) and DCWA (Chang et al., 1995). These support synchronous, real-time collaboration on shared documents by multiple authors, where they allow one user to concurrently edit a part at a time. Changes made by one author are made visible immediately to others and thus authors have a “WYSIWIS”—what you see is what I see”—experience. Some systems finesse this issue by only supporting asynchronous collaboration, e.g., CES (Greif & Sarin, 1986), Quilt (Fish, Kraut, & Leland, 1988;

Leland, Fish, & Kraut, 1988), and PREP (Neuwirth, Kaufer, Chandhok, & Morris, 1990). These systems are designed for asynchronous, long-term collaborations, where people do not usually work simultaneously. Thus changes made by collaborators are not visible – and thus do not affect – others in a real-time fashion. Of course, concurrency control in shared workspaces is not limited to document editors. For example, MOODS (Bellini, Nesi, & Spinu, 2002) is a synchronous real-time cooperative editor for music scores. Finally, some systems simply ignore problems with concurrency control, as inconsistencies are considered minor. Examples include shared drawing systems, where people collaboratively sketch on a common drawing area in real-time (Bly & Minneman, 1990; Ishii, 1990; Ishii & Kobayashi, 1992; Ishii, Kobayashi, & Grudin, 1993; Ishii & Miyake, 1991; Tang & Minneman, 1991; Tang & Minneman, 1991). Screen sharing systems, usually VNC or RDP based, allow a remote client computer to share the real-time screen images on a host computer.

In terms of the Locales Framework, shared workspace systems are designed for collaborative situations where artefacts are at the centre of a locale, as they are expected to be the focus of people and their activities. For example, in a collaborative writing session, the shared document being edited is at the locale`s centre. This contrasts to people-centered teleconferencing systems such as media spaces, where the center of the locale is expected to be the people in that shared space. This extreme view of artefacts vs. people can be tempered by augmenting the shared workspace with a video connection. Example shared whiteboard applications that do this are VideoDraw, VideoWhiteboard, and TeamWorkstation, and ClearBoard, as does the SEPIA document sharing system. Figure 3-4 shows ClearBoard where we see the remote person's video

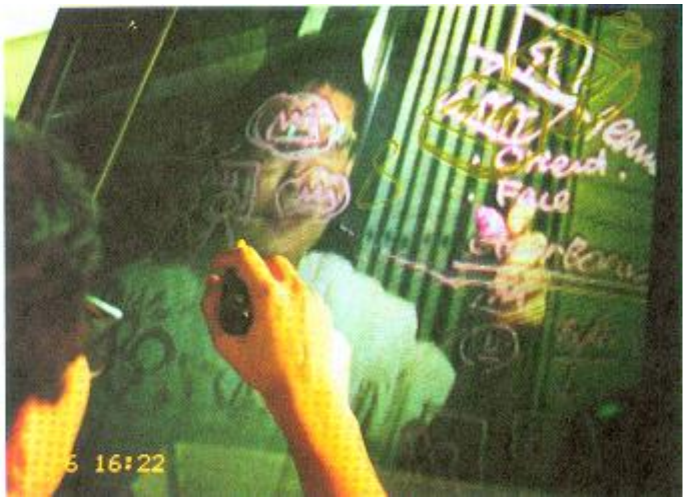


Figure 3-4 ClearBoard (Ishii & Kobayashi, 1992; Ishii, Kobayashi, & Grudin, 1993).

where another one can draw atop the image to create a shared artefact. Artefact sharing system can be augmented by other types of inter-personal information. For example, the Quilt, PREP, MACE, DCWA collaborative authoring systems support messaging between collaborators. SASSE, in the collaborative document view, provides information about who the collaborators are, where in the document they are working, and what they are doing (Baecker et al., 1993). It also has a shared tele-pointer to facilitate communication. Many screen sharing systems also support a voice connection between a host computer and a remote client computer.

From a locale perspective, a shared workspace approximates the site and means of a locale: it is a gathering point that offers the means to share artefacts and the tools to manipulate those artefacts. However, these systems pre-suppose that the artefact is always at the center of interest, and that this artefact is the sole reason why people gather at that site. This is a very restricted form of locale.

3.3 Integration of interpersonal interaction and artefact sharing

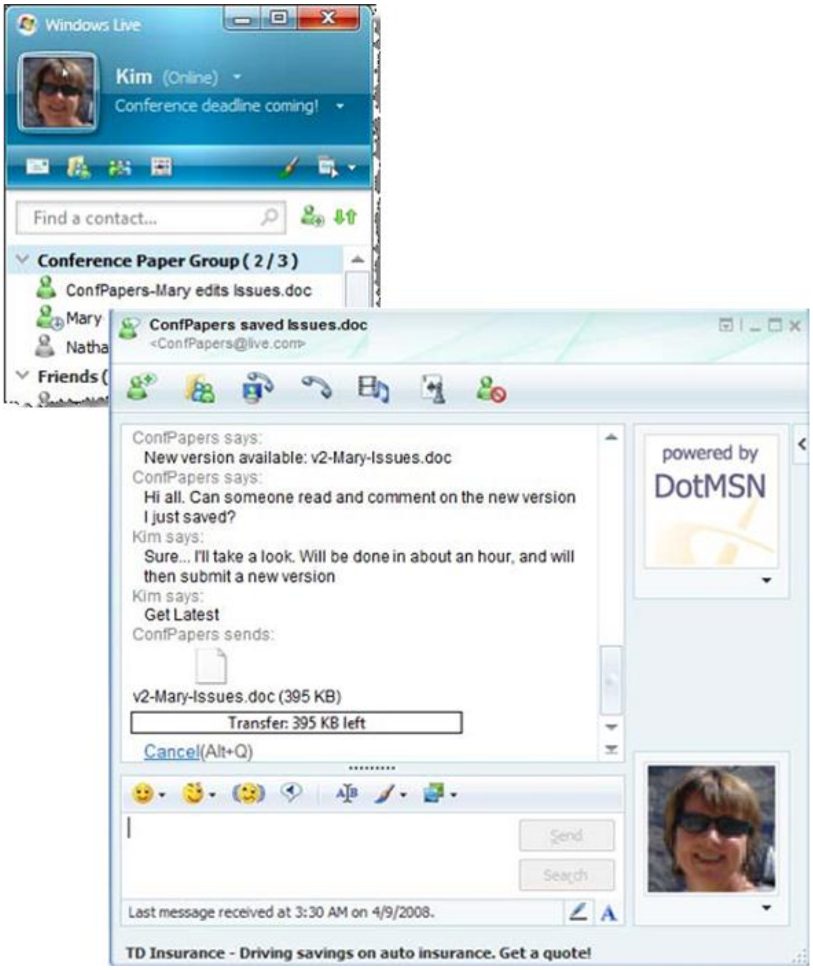
Some groupware systems do try to balance attention to people and artefacts.

3.3.1 Artefact integration with instant messaging

As Instant Messengers increased in both popularity and sophistication, they have begun to incorporate artefact sharing and coordination. Typical enhancements include artefact transfer between participants (e.g., files and photos), and the ability to start a shared whiteboard or shared screen after the conversation has been initiated. Several research systems take this one step further, as described below.

Doc2U (Morán et al., 2001) is an IM-based document sharing system. Like normal instant messengers, it has a buddy list for awareness of people. In addition, it provides a separate “document list” for awareness of shared documents that people collaborate on. A subscribed document appears in the document list in the same manner a contact appears on the buddy list. Users are notified of the status changes of subscribed documents through “instant message” style pop-ups. Similar to a contact in the buddy list, a document has a status indicator (online/offline/idle/locked). Status related to collaborators' interactions with a document is indicated as "being read", "being annotated", and "being written".

A locale in Doc2U is somewhat akin to a Doc2U "project", which defines a number of documents and authors. However, a person must use a separate web interface to upload documents and add authors to create a project, which is quite heavy-weight for casual artefact-sharing. As well, the Doc2U client user interface only has two lists: one for people and one for documents. There is no clear boundary between projects or groups, i.e., the user interface does not display any notion of a locale. Additionally, role management is heavy-weight, where people have to define strict access privileges to documents. In terms of awareness, a shared document appears like an IM "buddy" and only shows a few pre-defined status and availability indicators described above. When


Windows Live
Kim (Online)
Conference deadline coming!
Find a contact...
Conference Paper Group (2 / 3)
ConfPapers-Mary edits Issues.doc
Mary
Natha
Friends (
ConfPapers saved Issues.doc
<ConfPapers@live.com>
powered by
DotMSN
Send
Search
TD Insurance - Driving savings on auto insurance. Get a quote!

ConfPapers says:
New version available: v2-Mary-Issues.doc
ConfPapers says:
Hi all. Can someone read and comment on the new version I just saved?
Kim says:
Sure... I'll take a look. Will be done in about an hour, and will then submit a new version
Kim says:
Get Latest
ConfPapers sends:
v2-Mary-Issues.doc (395 KB)
Transfer: 395 KB left
Cancel(Alt+Q)

Last message received at 3:30 AM on 4/9/2008.

Figure 3-5 Artifact Buddy (Greenberg, Stehr, & Tee, 2008)

considered from the centre/periphery principle, no option is available to customize one`s perceived awareness of these indicators.

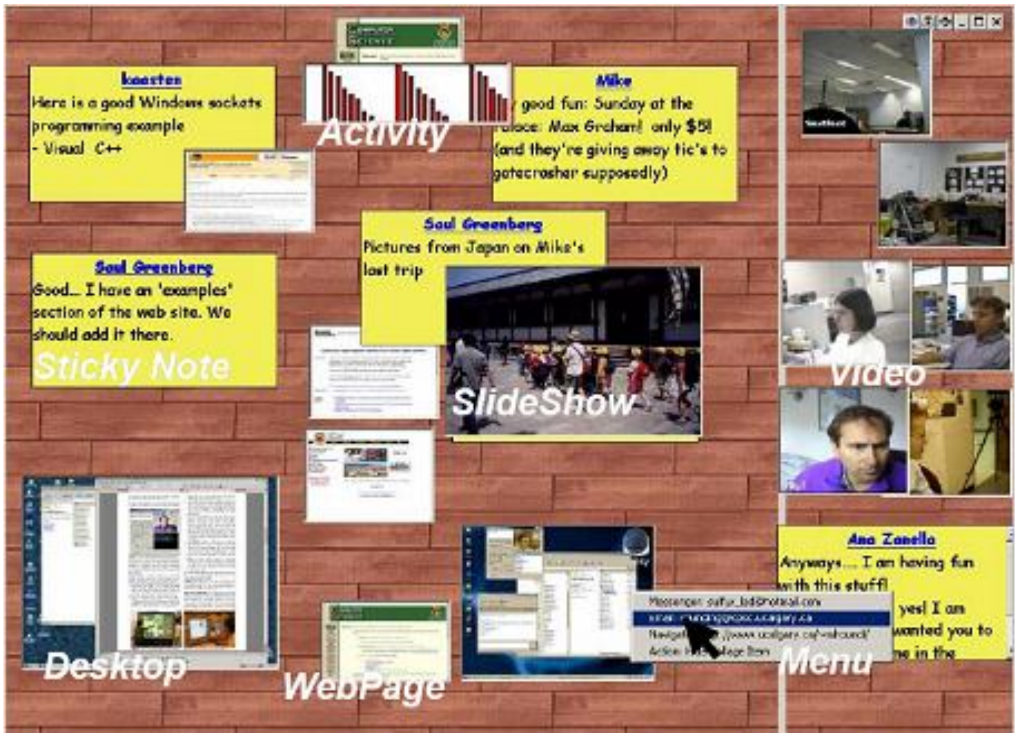
A somewhat similar system is Artifact Buddy (Greenberg et al., 2010) which incorporates a file as a buddy into a commercial instant messaging service. Artefact Buddy (Figure 3-5) equates files and people. A shared file is registered with the IM system as a user, and people add this shared file into their buddy list. We see this in Figure 3-5 as the 'ConfPapers' entry on the buddy list. A collaborator sees the status of the file, and can even start a chat session with the file (a person can send predefined commands to the file, such as requesting it to transmit an updated version of the document; examples are shown in Figure 3-5 bottom). Under the covers, a helper application monitors the file status and responds to chats, where it acts like a pseudo-user. Finally, the trajectory of the versioning and interactions with all collaborators are kept as a history.

Thus Artifact Buddy enables reasonably light-weight collaborations around shared artefacts, where it builds upon the affordances of the commercial instant messenger. As an extension of IMs, it suffers the same problems when viewed as a locale (albeit artefacts are now brought in as first-class entities). This additional facility does come at a cost (mostly arising from this being a hack to appropriate existing IM abilities). That is, forming an artefact-sharing locale is heavy-weight. A new instant messenger account for that artefact has to be manually created, and one user has to sign into it via the helper application. Other participants have to add this new "artefact buddy" to their buddy list.

3.3.2 Orbit

Orbit Gold (Mansfield et al., 1997) is a document collaboration system that attempted to directly implement the Locales Framework. It is currently limited to document collaboration, although its authors envisioned it more broadly as a collaborative desktop.

An Orbit user can involve and share documents in multiple locales (group zones). The Orbit user interface has two main components: the Navigator and the Workspace. The Navigator shows the list of locales the user is in, while the Workspace shows all documents shared in all locales with different color chips corresponding to different locales. Orbit Gold focuses on the aspects of locale foundations, individual views and mutuality. For mutuality, users can select how to make their presence available to others in a locale via a portrait icon or via video and audio. Awareness of shared documents is achieved by a very simple notification service: a red star appears on the corner of a document icon when an event occurs about that document. A user can select, from each locale, documents to be visible in the Workspace, and can aggregate them to form an individual view. They can also show all selected documents across all locales. The user can further customize the positions of these visible documents in order to produce an individual view to reflect centre/periphery difference between documents. However, the system is generally heavy-weight. Locale creation and configuration, member and artefact management all have to be done explicitly.



koesten
Here is a good Windows sockets programming example
- Visual C++

Mike
good fun: Sunday at the alace: Max Graham! only $5!
(and they're giving away tic's to gatecrasher supposedly)

Saul Greenberg
Pictures from Japan on Mike's last trip

Saul Greenberg
Good... I have an 'examples' section of the web site. We should add it there.

Ana Zanella
Anyways.... I am having fun with this stuff!
yes! I am
wanted you to
ne in the

Figure 3-6 Notification Collage (Greenberg & Rounding, 2001)

3.3.3 The Notification Collage

Various researchers have attempted to integrate the features found in IM, media spaces, and room-based systems. The Notification Collage (Rounding, 2004) is one such example (Figure 3-6). The Notification Collage presents itself as a single communal ‘room’, which each person usually keeps on permanent display on a second monitor. People enter that room by connecting to a server address, after which they see all activities within it. As with MOO chat rooms, this room is somewhat akin to a locale. Rooms contain multiple *media items* (small interactive windows). As soon as a person enters the room, a video media item is created showing a video stream of that person fed from a live camera, i.e., the room automatically behaves as a media space. This also serves as a type of buddy list, for one can now quickly see who is online and what they are doing. In addition, people can post other media items, such as sticky notes, web pages, shared screens, slide shows and so on. The sticky note (which can be typed into) serves as a text chat, thus making conversation easy to initiate. People can also enter into audio conversation by clicking on another’s video. The other media items let people post and interact with activities and content of mutual interest (i.e., the ‘means’). This includes activities outside the room, e.g., the shared screen posts thumbnails of a person’s actual screen, and lets others move into screen-sharing sessions. Finally, individual views are also supported to some extent, as people can easily customize the size and position of media items on their own display for a personal view of the connected virtual world.

The biggest limitation with Notification Collage is that it offers only a single permanent ‘locale’. Everything created within it is visible to others. Creating another ‘locale’ means creating a server, something that is heavy-weight, Although actually easy to invoke a new server, advertising that server location is difficult and getting people to join must be done out of band, i.e., these sites are not discoverable. As mentioned, one site - even a physical room - rarely maps onto the idea of one social world and one locale for only one purpose. In contrast to nuanced locales, the Notification Collage supports community awareness and interaction. The single site and ever-present communication and awareness of all activities considers this community as a single social world, in one pre-constructed, permanent locale. Locales may form within the Notification Collage, but this comes from how people perceive their activity rather than from how a locale is explicitly supported by the software.

3.4 Community Bar

The Community Bar (McEwan, 2006) was expressly designed around the Locales Framework. Its primary intention was to support light-weight and casual interactions for small groups of intimate collaborators. It also heavily influenced the Come Together system I introduce in the next chapter.

The Community Bar is a sidebar (Figure 3-7) located at the periphery of a person`s screen. The sidebar contents provide peripheral awareness of people and artefacts, and a means for people to move into conversation, and into interaction over artefacts. The small tiles on the bar show are media items, each designed for a specific purpose. One type shows presence information of people via a live video or static image. Another type is a persistent textual chat. Others involve artefact sharing, e.g., a shared photos media item, a shared web page media item, and even a shared screen item.

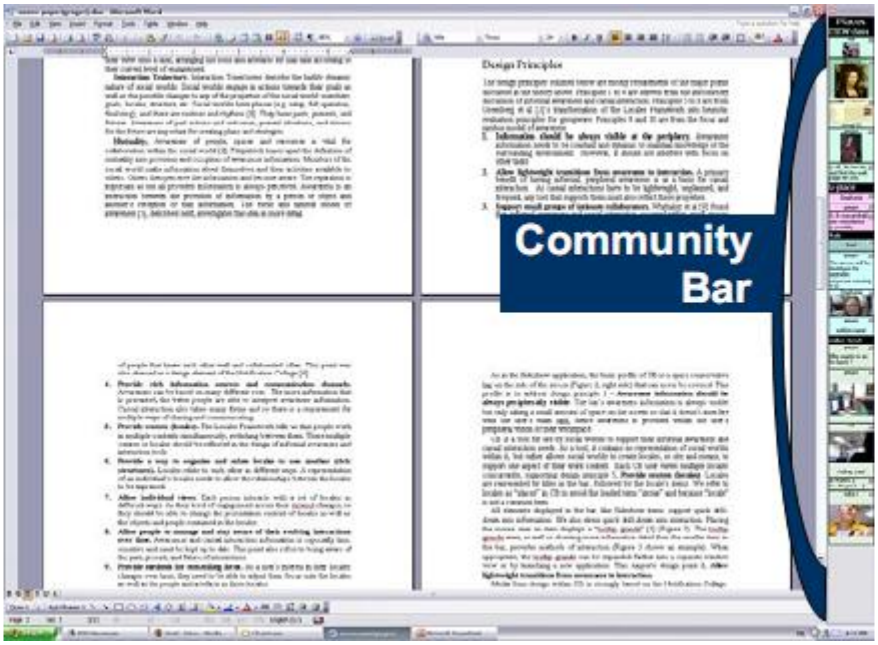


Figure 3-7 The Community Bar (McEwan, 2006)

The entire sidebar (Figure 3-8) is divided into segments, each called a Place, where each place emulates a locale. A Place (site) is a container for a number of media items (means), which as explained are small tiles representing people and artefacts. Hovering over a tile brings up a tooltip grande (a larger tile), offering a higher-fidelity image or representation of the person or artefact. Clicking on the tile of the tooltip grande raises a full, interactive view with maximal level of fidelity. Under the covers, the Community Bar implements a plug-in architecture where developers can create third-party media items (beyond the stock video, chat, photo, web items) (McEwan et al., 2006).

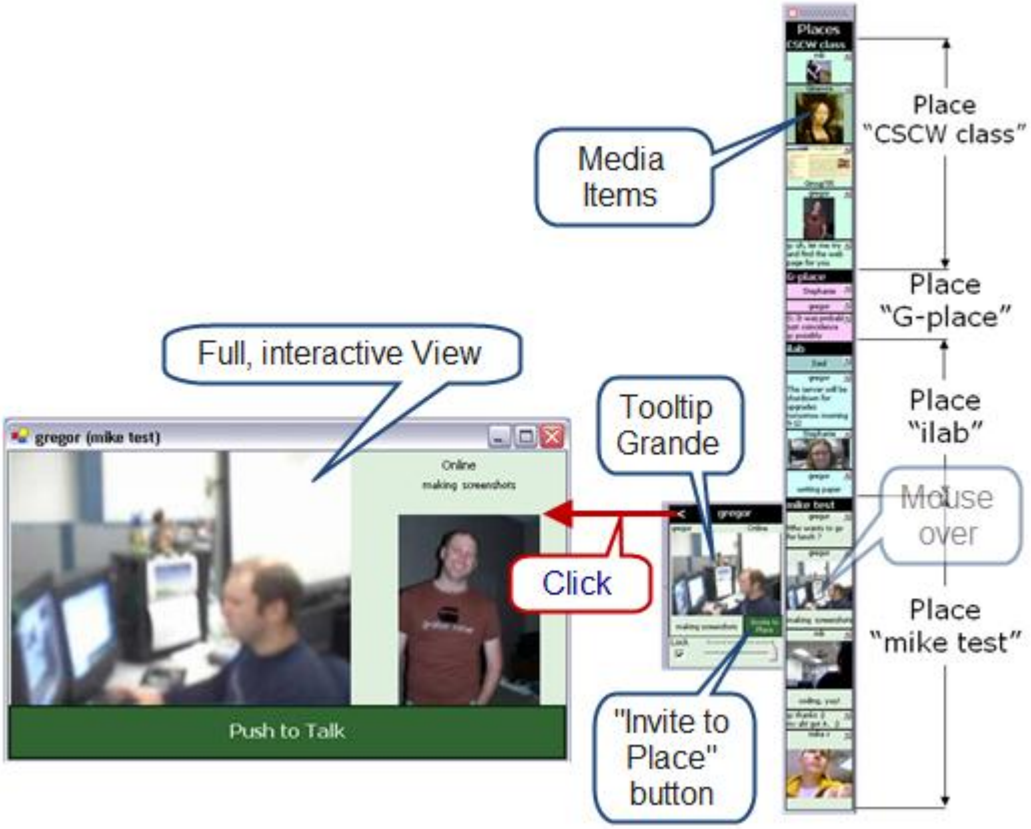
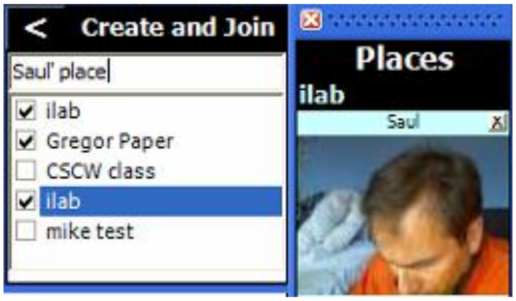


Figure 3-8 the Community Bar (McEwan, 2006)



Saul' place

☑ ilab
☑ Gregor Paper
☐ CSCW class
☑ ilab
☐ mike test

Figure 3-9 (McEwan, 2006)

The process of creating and using Places deserve special mention. A person can create and name a Place for a new locale via a pop-up window accessed from the top of the sidebar (Figure 3-9). Once a Place is created, people can be invited into it via an “Invite To Place” button, found in the tooltip grande of a person (Figure 3-8). Afterward, the person has to accept the invitation to join the Place. This is clearly heavy weight. Indeed, a field study of Community Bar use in practice (Romero et al., 2007) revealed significant flaws with this heavyweight Place design: people almost never created Places for new locales. Instead, they all stayed in one single Place, where they would appropriate that Place into multiple mini-locales either conceptually (by how they thought about relationship between media items) or (more rarely) by positioning related media items together into groups. The cost of using mini-locales rather than Places is that people suffer from distractions of uninteresting information.

The above treatment of locales is consistent with the Locales Framework, as it shows that locales are indeed formed. Yet the explicit mechanism provided by CB to create locales (Places) is ignored, as it proved too heavy-weight, demanding a priori place creation along with a multi-step invitation protocol. This was enough to stop people creating explicit Places. The actual use of mini-locales indicates a way how a locale should be formed and maintained: simple user interface protocols by spatial positioning of related items. We will return to this notion in the next chapter.

In spite of this failure in CB, its design concept of Place is reasonable, i.e., as a way to provide a site and means for light-weight, casual interactions of small groups. The media item’s drill-down design—tile view, transient view (tooltip grande), and full view—did facilitate natural transition from awareness to interactions (Figure 3-8). On the sidebar, users can adjust the size of the displaying media items in their tile views for different levels of awareness. For example, depending on the size, a tile for a person may show a name, a photo or a live video. However, its user interface was clumsy, so this was rarely done. Beyond that, tooltip grande and full view provide for higher levels of awareness. Thus, in terms of the centre-periphery principle, a user can customize the level of interest or engagement to another person or artefact through different sizes, from low-fidelity to high-fidelity, and different views, from peripheral awareness to close interactions, of a media item. An individual view of a locale is achieved through the aggregate of custom-sized media items in a Place. In the next chapter, we will see how the successful aspects of Community Bar influenced the design of my next generation system.

3.5 Design inspirations

Most of the reviewed systems hinder light-weight locale formation by forcing users to configure the site and/or means ahead of time. Most notably, even those systems designed to support light-weight group practices (including the Community Bar's Place) impose a high threshold to start and maintain a locale.

Yet there are many positive aspects of these systems. For example, instant messengers are exceptionally efficient for impromptu, transient communication sessions. Forming a simple locale of a chat session is as simple as a double-clicking. This suggests that a groupware design must not impose too much overhead—unnecessary user interface protocols—if it is to support fluid interactions. Yet another successful feature of instant messengers is the separate buddy list, which provides awareness of people to facilitate the formation of a chat locale. My system design—described in the next chapter—uses a similar approach to support light-weight locale formation through the easy transition from awareness of people and artefacts. The site and means for a locale should be created and configured implicitly as an impromptu goal or interest, which involves the core members of a social world and their artefacts, emerges.

Next, most system designs do not consider the centre/periphery principle. They usually support awareness at a fixed level of fidelity and provide a fixed view for constructed locales. The Community Bar is a notable exception, where it successfully uses drill-down media items of variable-size to reflect a user's different levels of interest and engagement with a person or artefact. As we will see in the next chapter, I adopt this media item model within my own design, i.e., I provide multiple levels of presentation detail for a person or an artefact both in and out of locales.

Another design point considers the integration of people and artefacts. As mentioned in the first chapter and illustrated in this chapter, some systems primarily focus on interpersonal interactions, while others mainly concentrate on artefact sharing.

Artefact Buddy, Notification Collage, and Community Bar are exceptions, as they show people and artefacts by the same representation. As argued in the first chapter, people and artefact are actually equally important components of locales in terms of interactions. Inspired by these systems, my own implementation treats people and artefact equally in and out of locales.

Finally, within Notification Collage and Community Bar, people and artefacts can only exist after a locale has been created. As well, an artefact cannot be shared across locales. These conditions can hinder how locales are created and used. Again, my system will consider people and artefacts as persistant entities that can exist outside of a locale and that can be easily brought into and shared across multiple locales.

Chapter 4. **Come Together**

This chapter introduces a light-weight groupware system—Come Together (CT). Based on the theory of the Locales Framework, its design emphasizes light-weight formation of locales, the centre-periphery continuum, and people-artefact integration. CT also attempts to incorporate the merits of existing groupware systems, especially instant messengers and the Community Bar (McEwan, 2006). For example, the primary goal, light-weight locale formation, is a feature of instant messengers, which the Community Bar fails to deliver. On the other hand, Come Together adopts the Community Bar's multi-level, drill-down media item model, which successfully supported the centre-periphery principle and awareness-interaction transition. Furthermore, the system represents people and artefacts equally as media items and allows them to be brought into and taken out of locales, i.e., they exist both as independent entities and as part of a locale. As a reminder, table 4-1 is a list of the design considerations which I developed and applied to the CT design, following the process in Appendix C.

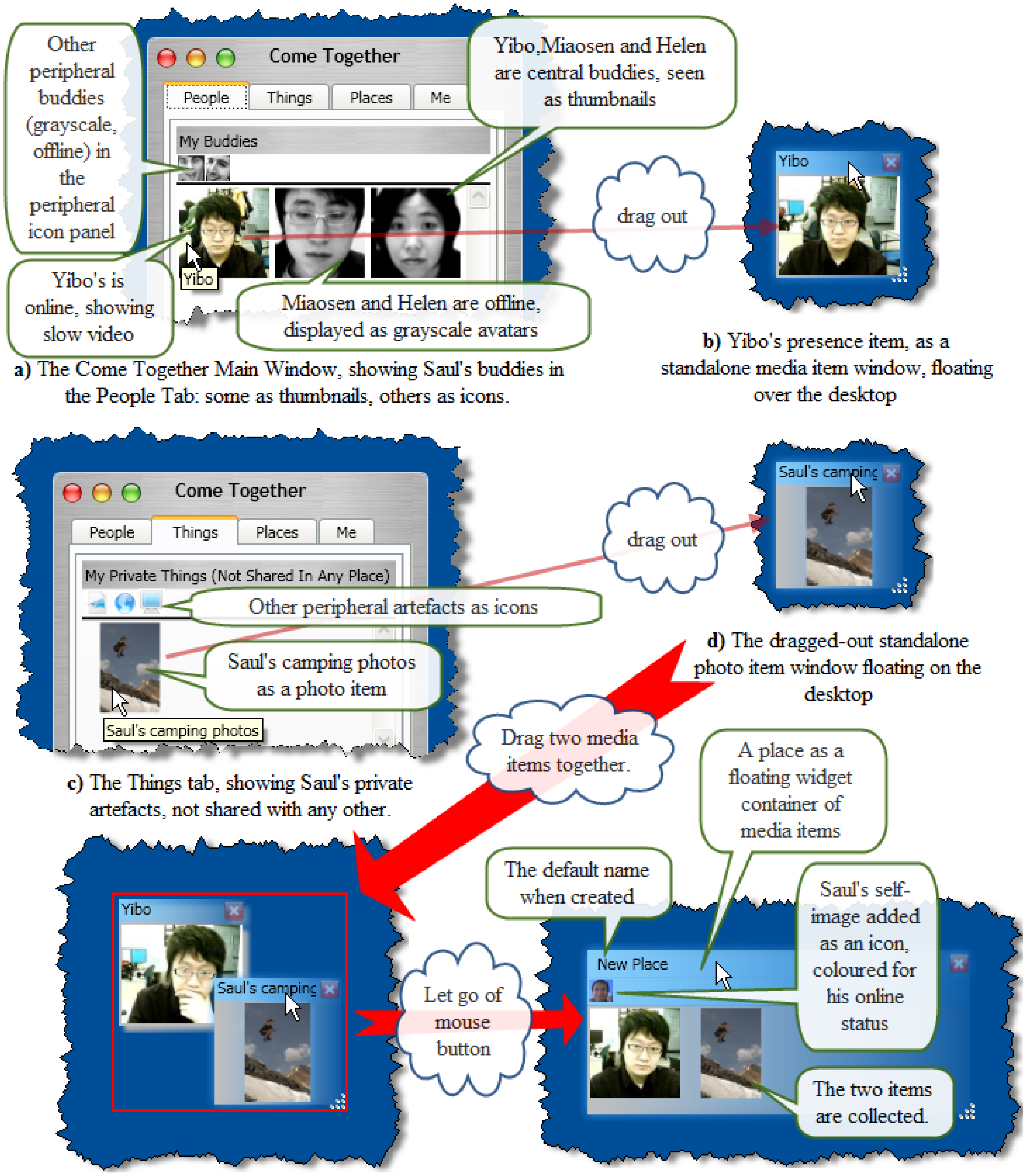
The chapter introduces Come Together by four scenarios that progressively reveal its main features and building blocks. A later chapter will deconstruct the user interface—*media items* representing people and artefacts, *places* collecting media items, the *CT Console* managing all people, artefacts, and locales, and the use of the entire desktop space—to match the design philosophy and goals mentioned in the previous chapters. That later chapter will also introduce a number of stock media item types included in Come Together.

The four scenarios in this chapter describe a group's use of CT. The scenarios are contrived to keep the group's tasks deliberately trivial to illustrate the process enabled by Come Together. While some of the images were edited after captured to ease image creation, they all show the real system's user interfaces and interaction flows.

In this group, Saul is a university professor; Yibo, Helen, Misaosen are graduate students supervised by Saul. They regularly use Come Together for their casual interactional needs. Each had previously created an account on the Come Together server. The first scenario shows how Saul and Yibo use Come Together to construct and destruct a short-lived, light-weight locale for photo sharing with Yibo. The other scenarios show how the group uses additional features to support the people-artefact integration and the centre-periphery differentiation.

1	A person and/or group should be able to easily and spontaneously create, maintain and dissolve a locale.
2	Because goals may be formed and altered over time, the system should not require the group to configure the site and means to satisfy a particular goal *a priori*.
3	Membership should be flexible. People should be able to see what a locale is about without ‘joining’ it. Similarly, they should not require an invitation, or have to go through a chairperson or moderator. If they do become part of the social world of a locale, a person should be able to adjust his or her level of involvement from the center to the periphery.
4	The group should follow its own social protocols and roles rather than a social protocol or role imposed by the system. That is, the system should support what people do naturally rather than demand they follow a prescribed set of rules and roles.
5	The trade-off between access and security should be maintained primarily through social means, where system control for access and/or security is added only if desired by the group.
6	The system should capture awareness information in a manner that reflects that person’s engagement with the group (i.e., center/periphery), where the person could also choose how to view that information.
7	The capture and presentation of awareness information should be adjusted to reflect a person’s dynamic movement across the center/periphery continuum.
8	The system should allow an individual to form their view of a locale through the aggregation of received awareness information of each entity—people, shared artefacts and interactions—all at different levels of attention and engagement.
9	An individual view should be updated to match one’s interest and engagement in a locale over time, e.g., where some entities in the locale move closer to the person and others fade out to the periphery.
10	The view set of all locales one is involved in should change to reflect the shifts of one’s interest and engagement, e.g., from some specific locales to others.

Table 4-1 Group Design Considerations for lightweight group working practices


Other peripheral buddies (grayscale, offline) in the peripheral icon panel
Come Together
People
Things
Places
Me
My Buddies
Yibo
Yibo,Miaosen and Helen are central buddies, seen as thumbnails
Yibo's is online, showing slow video
Miaosen and Helen are offline, displayed as grayscale avatars
drag out
Yibo
Come Together
People
Things
Places
Me
My Private Things (Not Shared In Any Place)
Other peripheral artefacts as icons
Saul's camping photos as a photo item
Saul's camping photos
drag out
Saul's camping
Drag two media items together.
Yibo
Saul's camping
Let go of mouse button
The default name when created
A place as a floating widget container of media items
New Place
Saul's self-image added as an icon, coloured for his online status
The two items are collected.

a) The Come Together Main Window, showing Saul's buddies in the People Tab: some as thumbnails, others as icons.

b) Yibo's presence item, as a standalone media item window, floating over the desktop

c) The Things tab, showing Saul's private artefacts, not shared with any other.

d) The dragged-out standalone photo item window floating on the desktop

e) A red frame indicating the system is about to create a place (a locale).

f) The system aligns the two items and creates a "New Place" (locale).

Figure 4-1 Saul's view: creating a place.

4.1 Scenario 1. Constructing and Destructing a Short-Lived Place

Saul loves to share photos of his adventures. He has just returned from a camping trip, and takes this moment to share selected photos with whichever of his students are online.

After Saul logs onto Come Together via a typical name/password dialog, the main screen (CT console) appears showing the People tab (Figure 4-1a). He sees three of his buddies - Yibo, Miaosen, and Helen - portrayed within thumbnail-sized “presence media items”, and two other more peripheral buddies as icon-sized media items (these media items are somewhat akin to those in the Community Bar; the size difference is explained shortly). Yibo is online, so Saul sees his presence as a live video feed. The others are offline; so Saul sees them as a static photo (two thumbnails and two icons), muted into grayscale by the system to indicate this offline status.

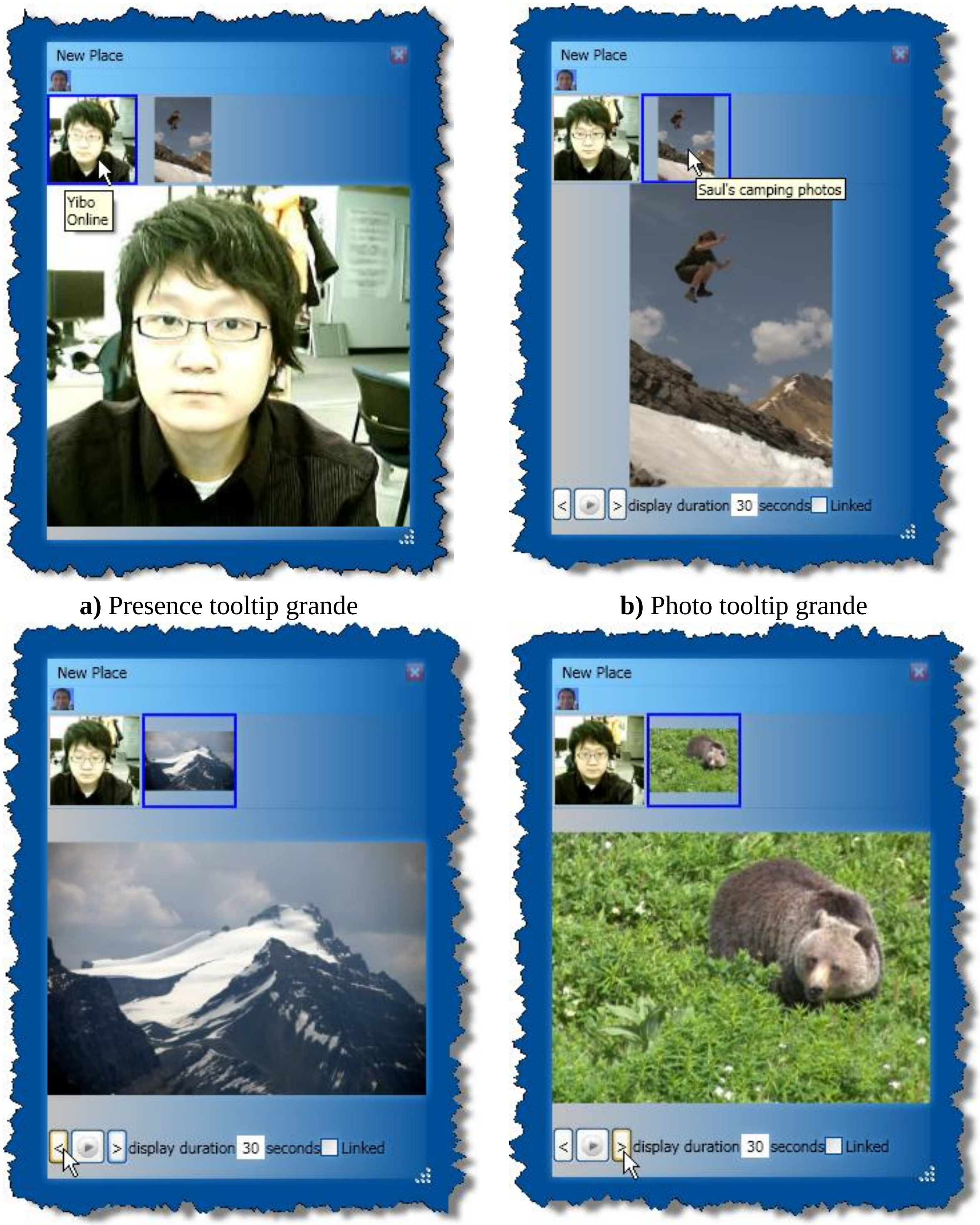
Because Yibo is online, Saul decides to share the photos with him, where he will create a place that includes Yibo, a photo item and a chat item. First, he drags Yibo’s presence item out of the CT console, which creates a duplicate media item that now appears as a small standalone media item window, title “Yibo”, floating over the desktop space (Figure 4-1b). Second, Saul switches to the “Things tab” (Figure 4-1c), which lists his artefacts previously collected as media items. In this case, his camping photos are in the sole thumbnail-sized “Photo media item”. A few other peripheral icon-sized media items for other artefacts (similar to the size difference of presence item, explained shortly) are located in the “My Private Things (Not Shared In Any Place)” section of this tab (other sections of this tab are also explained shortly). He drags the photo media item out of the CT console and, as before, a copy of this media item appears as a standalone media item window (Figure 4-1d).

Third, Saul creates a Place (a locale) collecting Yibo, himself, and the photos simply by continuing to drag the photo window toward Yibo’s presence item. As they near one another, a red frame appears around them, indicating that the system is about to create a locale containing Yibo and the photos (Figure 4-1e). As soon as Saul releases the mouse, the two media items—Yibo’s presence item and the photo item—are aligned and collected in a new larger standalone place window titled “New Place” by default (Figure 4-1f). Saul is also in this place as he created it, but the system, by default, only shows his presence as an icon-sized media item containing his static avatar image, located above the two thumbnail-sized media items.

Saul now explores this Place using a mechanism that is somewhat akin to the Tooltip Grande pop-ups found in Community Bar (McEwan, 2006). When he moves his cursor over Yibo's presence item, the size of the Place automatically increases to reveal a pane holding a larger video of Yibo running at a higher frame rate, appears within it (Figure 4-2a).

Similarly, when he continues to move his cursor over the photo item, a larger photo from that collection appears (Figure 4-2b). When he moves the cursor away from the two media items, the large views disappear and the place shrinks down to its original size (Figure 4-1f). Saul can also lock any of these larger full views of a media item into place by clicking on it. For example, when he clicks on the photo item, the larger full photo view will persist, which will let him move his cursor elsewhere, e.g. to use the controls at the bottom to flip through the photo set (Figure 4-2c).

Because Saul added Yibo's presence item to this place, various things happen on Yibo's display. First, Yibo sees a notification appear as a popup 'toast' from the Windows tray area, which invites him into a new place (Figure 4-3a). One way to act on this is for him to go to his Places Tab, which lists all his places. He sees a new titled "New Place" containing Saul's video and the same photo thumbnail as seen on Saul's screen (Figure 4-3b). He drags this "New Place" to his desktop (Figure 4-3c), and he can now explore Saul's video and the photos as Saul did above (Figure 4-3d). The only major difference is that Yibo's place shows Saul's thumbnail-sized presence item, while Yibo's presence item is included as an icon by default. Currently, Yibo and Saul see the same set of photos, but the photos are not strictly linked, i.e. Yibo and Saul can independently navigate and view slides. However, if either selects the "Linked" checkbox (Figure 4-3d, bottom), both will see the same slide and can concurrently navigate through them.



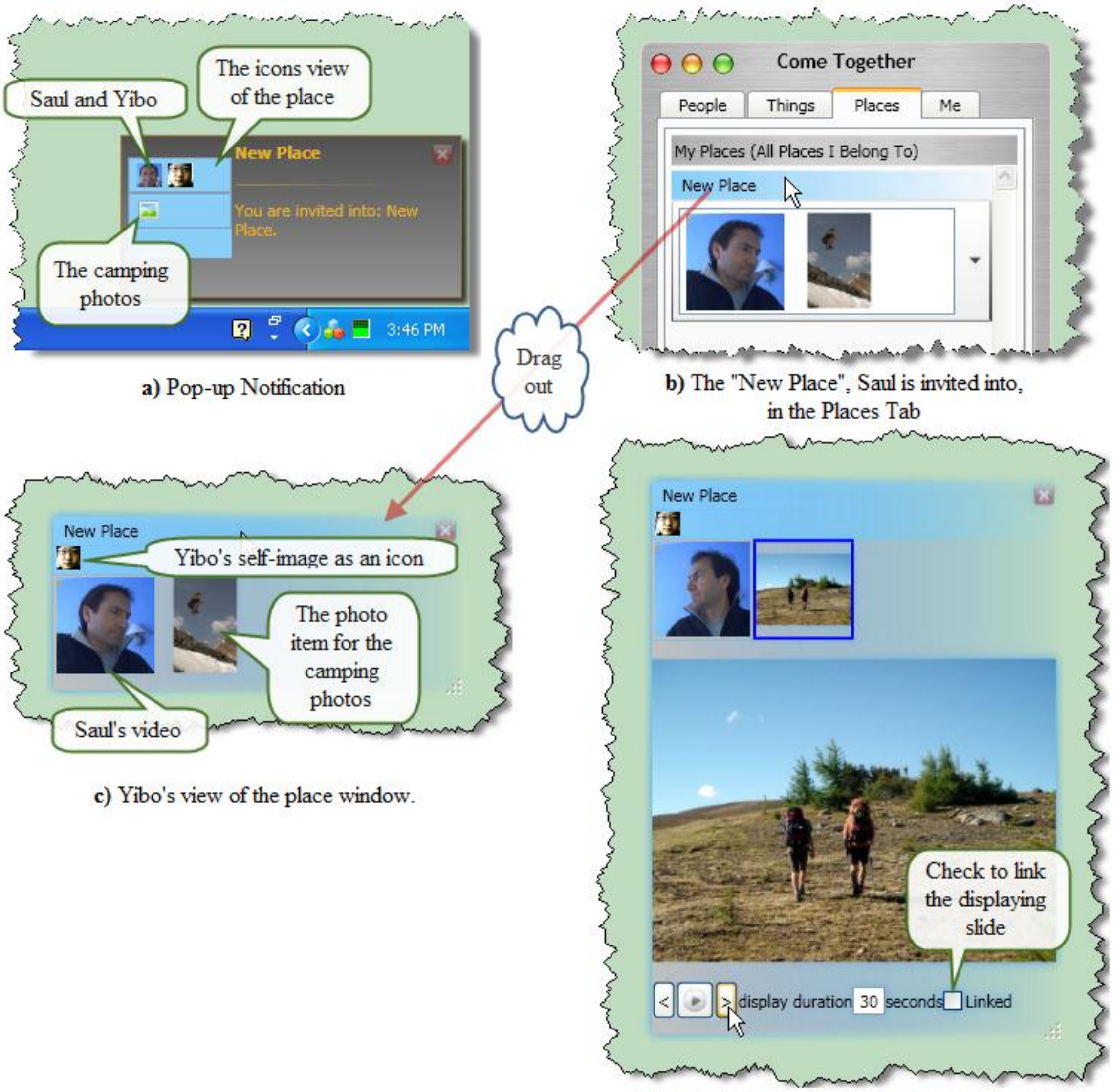
display duration 30 seconds ☐ Linked

a) Presence tooltip grande

b) Photo tooltip grande

c) Using the photo tooltip grande to view photos

Figure 4-2 Exploring a place’s media items


Saul and Yibo
The icons view of the place
New Place
You are invited into: New Place.
The camping photos
3:46 PM
Drag out
Come Together
People
Things
Places
Me
My Places (All Places I Belong To)
New Place
New Place
Yibo's self-image as an icon
The photo item for the camping photos
Saul's video
New Place
Check to link the displaying slide
display duration 30 seconds
Linked

a) Pop-up Notification

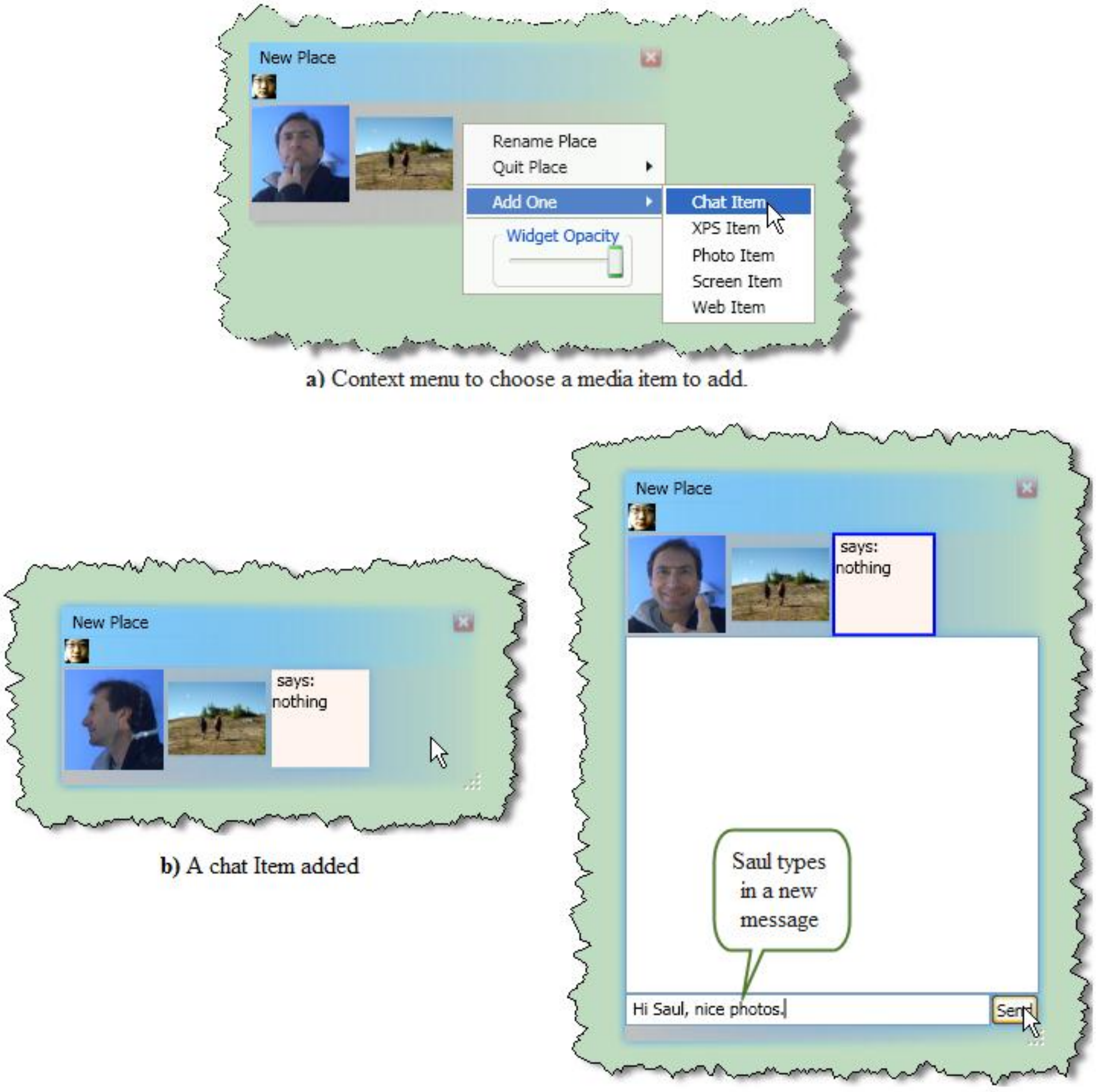
b) The "New Place", Saul is invited into, in the Places Tab

c) Yibo's view of the place window.

d) Lock the large, interactive photo view and flip through the photos.

Figure 4-3 Yibo’s view: joining and using the place.

Yibo now starts a text chat with Saul. He raises a context menu by right-clicking an empty area of this place, and selects the chat item from list of available media items (Figure 4-4a). This creates and adds a “chat item” to the place (Figure 4-4b). Yibo clicks the chat item, and an expanded, interactive chat panel appears below (Figure 4-4c). He types in the first message to the conversation.



a) Context menu to choose a media item to add.

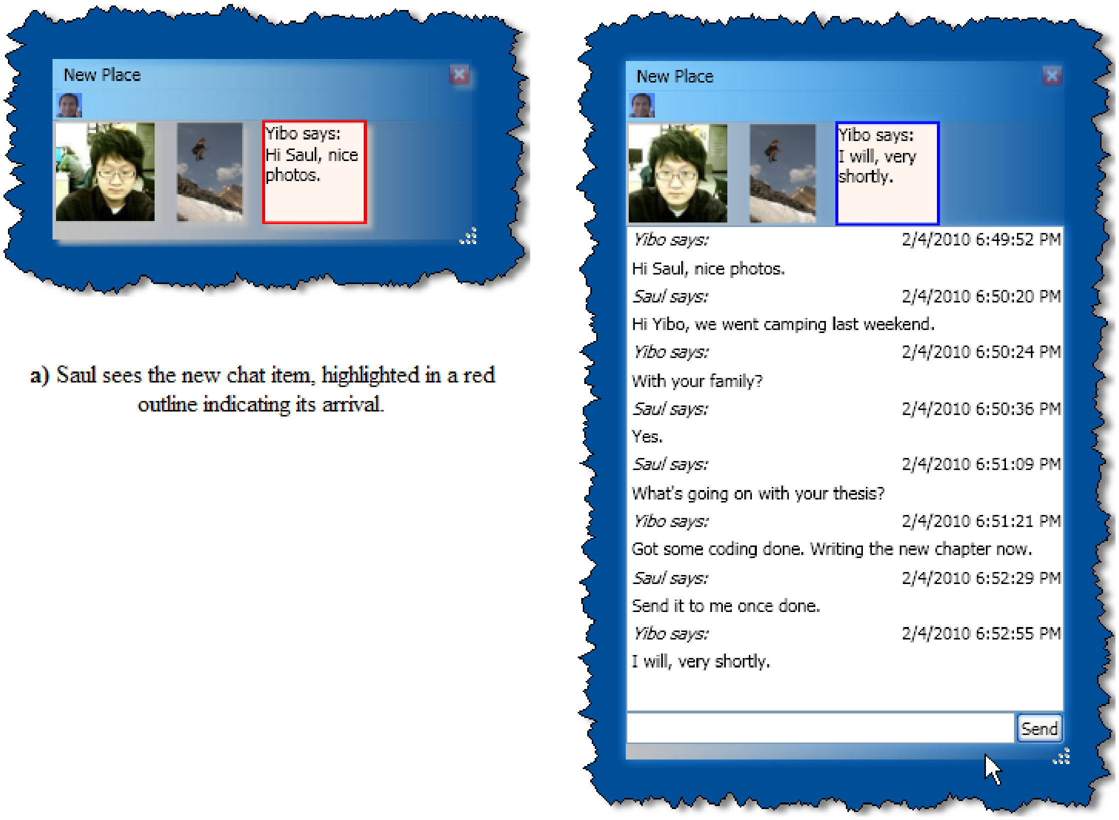
b) A chat Item added

Hi Saul, nice photos.

c) Interactive Chat Panel

Figure 4-4 Yibo adds a chat item to an existing place

As Yibo does these acts, Saul sees the new chat item added to his view of the place window, outlined in red to indicate its arrival (Figure 4-5a). Saul also expands the chat item and chats with Yibo (Figure 4-5b), where they talk briefly about the photos (and also Yibo's thesis progress).



Yibo says:
Hi Saul, nice photos.

a) Saul sees the new chat item, highlighted in a red outline indicating its arrival.

Yibo says:
I will, very shortly.

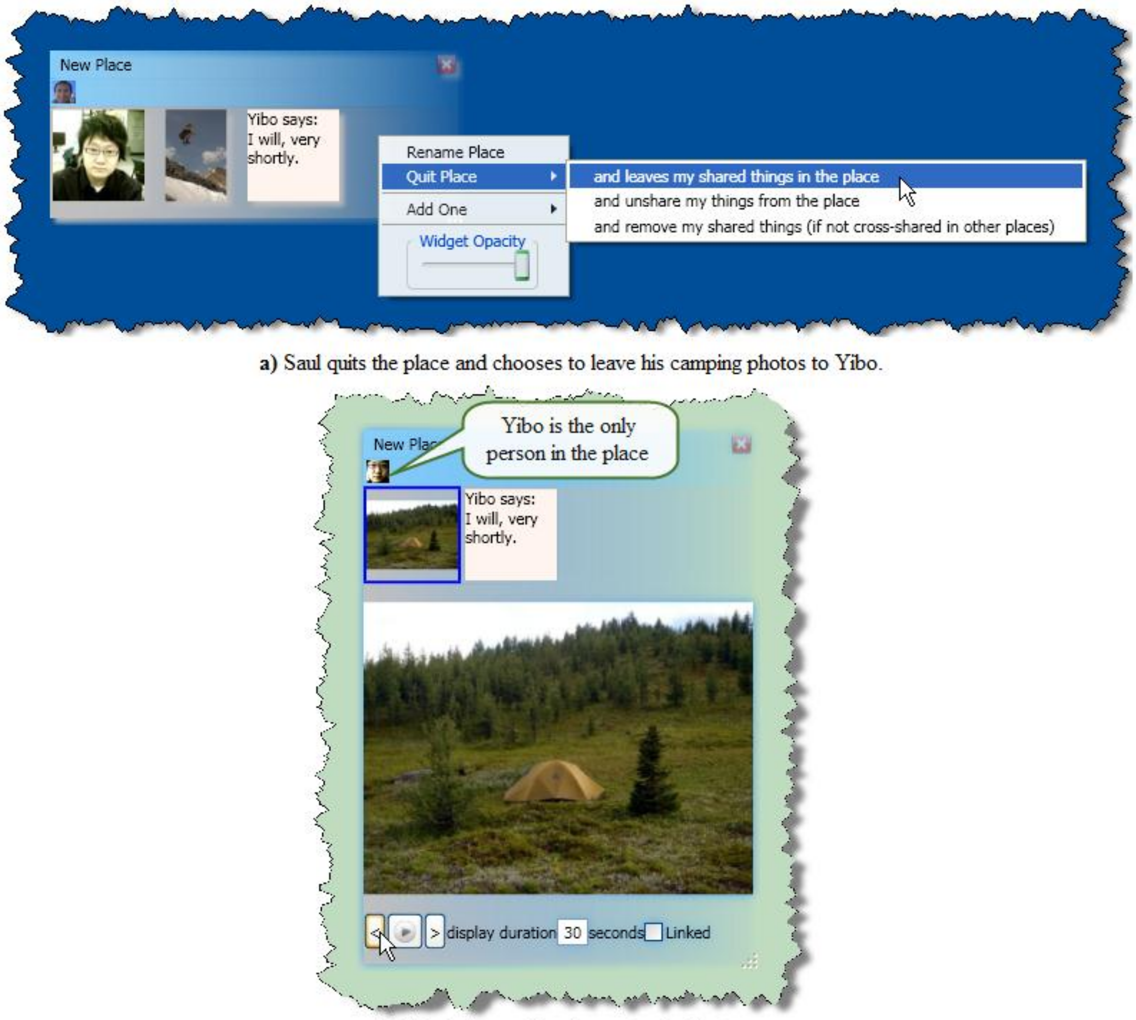
```
Yibo says:                2/4/2010 6:49:52 PM
Hi Saul, nice photos.
Saul says:                2/4/2010 6:50:20 PM
Hi Yibo, we went camping last weekend.
Yibo says:                2/4/2010 6:50:24 PM
With your family?
Saul says:                2/4/2010 6:50:36 PM
Yes.
Saul says:                2/4/2010 6:51:09 PM
What's going on with your thesis?
Yibo says:                2/4/2010 6:51:21 PM
Got some coding done. Writing the new chapter now.
Saul says:                2/4/2010 6:52:29 PM
Send it to me once done.
Yibo says:                2/4/2010 6:52:55 PM
I will, very shortly.
```

b) Saul chats with Yibo.

Figure 4-5 How Saul sees the created chat item

When their conversation is completed, Saul leaves. He raises the context menu, and selects "Quit the Place and leave my shared things in the place" (Figure 4-6a). This option means that Yibo can still look at the shared photos, even though Saul has left. At this point, the Place only contains a single person (Yibo) and the two artefacts (Figure 4-6b).

Somewhat later, Yibo also leaves the place, and a system dialog asks if Yibo wants the place destroyed since no one is in it. Yibo chooses to destroy it.



Yibo says:
I will, very shortly.

New Place
Rename Place
Quit Place
and leaves my shared things in the place
and unshare my things from the place
and remove my shared things (if not cross-shared in other places)
Add One
Widget Opacity
Yibo is the only person in the place
< > display duration 30 seconds □ Linked

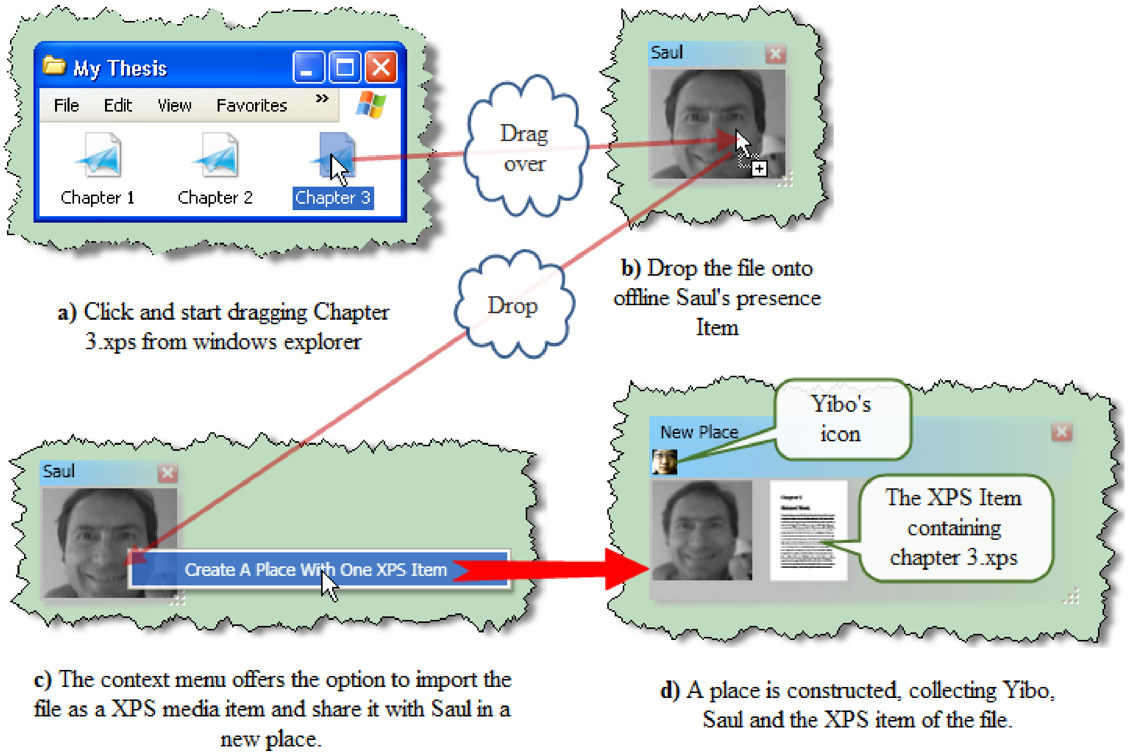
a) Saul quits the place and chooses to leave his camping photos to Yibo.

b) Yibo's view of the place after Saul left.

Figure 4-6 Saul leaves the place; Yibo remains in the place.

4.2 Scenario 2. Creating a locale for asynchronous collaboration.

Yibo has just finished his new thesis chapter. He decides to create a place containing this chapter, where Saul can look over this material at his leisure. At the same time, he also decides to share some of his primary source references used in creating his chapter with other students via a different place.



a) Click and start dragging Chapter 3.xps from windows explorer

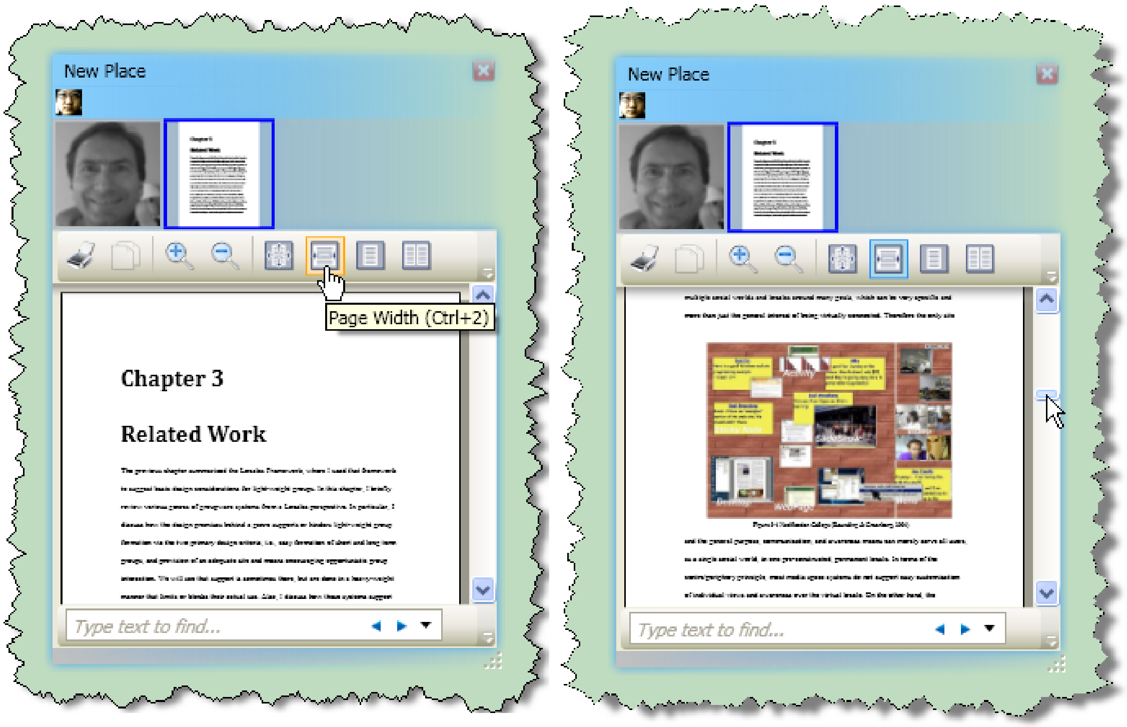
b) Drop the file onto offline Saul's presence Item

c) The context menu offers the option to import the file as a XPS media item and share it with Saul in a new place.

d) A place is constructed, collecting Yibo, Saul and the XPS item of the file.

Figure 4-7 Another way to create a place with an artefact directly from the operating system.

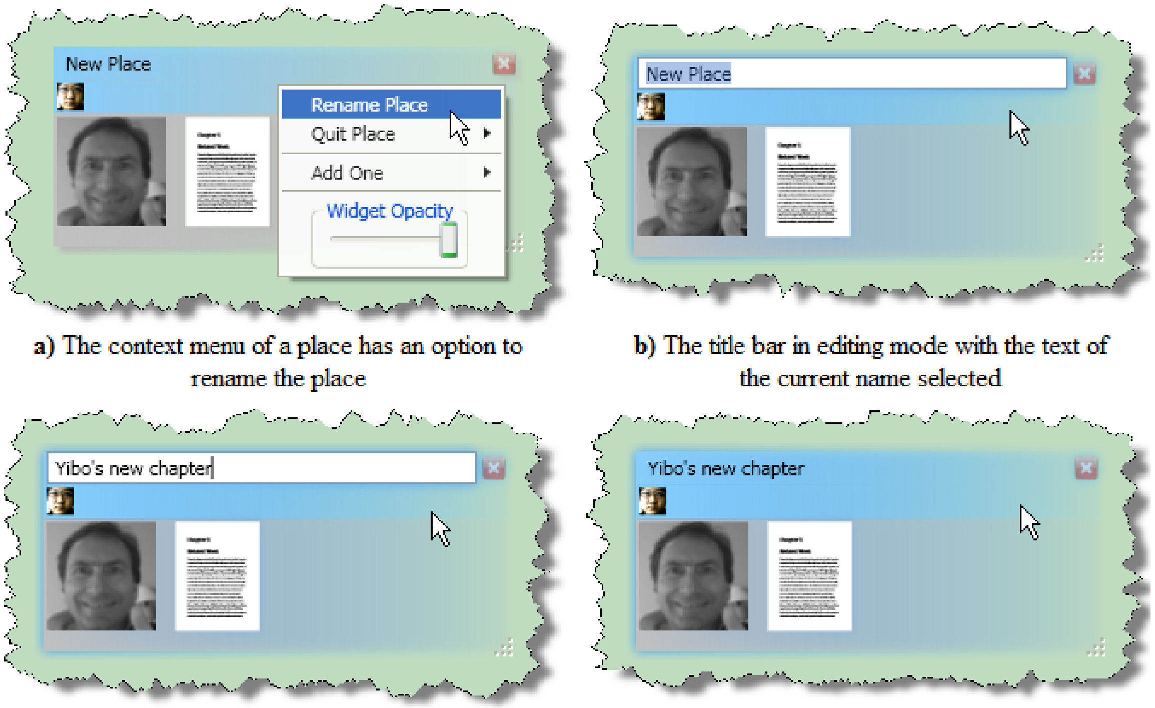
First, he shares the chapter with Saul. As before, Yibo goes to his CT console, where he drags Saul (whose avatar shows he is offline at the moment) from the People Tab to the desktop. However, Yibo does not yet have a media item representing his thesis chapter. He creates one simply by dragging and dropping his thesis chapter (an XPS file) from his standard Windows folder (Figure 4-7a) onto Saul's standalone presence item window on the desktop (Figure 4-7b). Come Together recognizes the XPS file type, and raises a popup asking Yibo if he wants to create a document viewer media item (Figure 4-7c). Yibo confirms, and a new place window is created containing Saul's offline grayscale thumbnail avatar, Yibo's icon avatar and a new document viewer media that automatically imported Yibo's "chapter 3.xps" file (Figure 4-7d). Yibo (and Saul, when he comes on line) can now use the document viewer's tooltip grande to look at the full view of the XPS document, scroll within it, change the view magnification, and so on (Figure 4-8a, Figure 4-8b). This example illustrates an alternate way to create a place (by dragging existing artefacts onto people), and how media items can be created via implicit actions.



a) A locked, full view of the XPS item

b) Scroll within the XPS document

Figure 4-8 Viewing the XPS file



a) The context menu of a place has an option to rename the place

b) The title bar in editing mode with the text of the current name selected

c) Typing in a new name

d) Hitting the Enter key finalizes the name

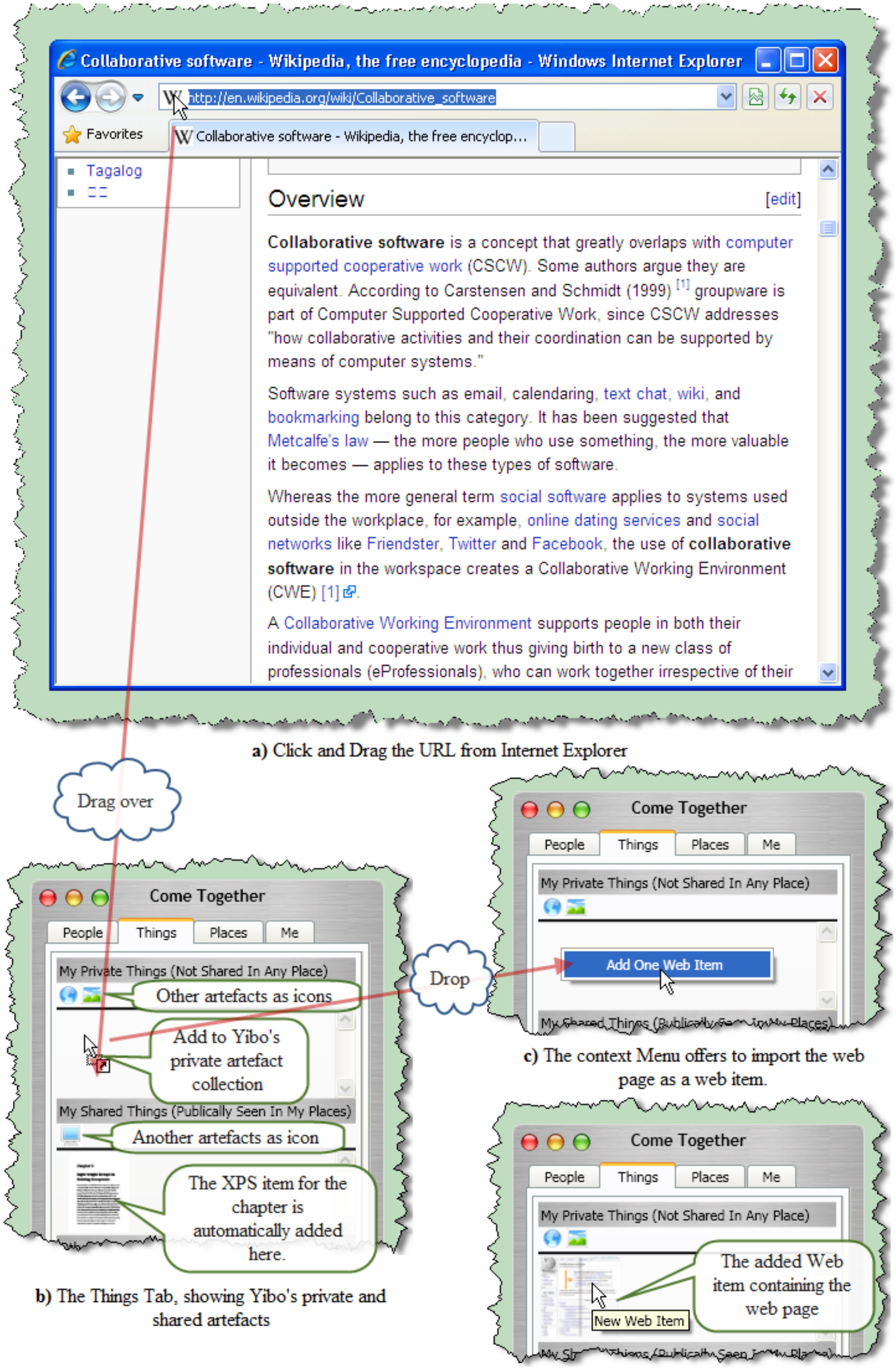
Figure 4-9 Renaming a place

To give Saul an idea of the purpose of this place, Yibo renames the place to “Yibo’s new chapter”. As typical in most Windows applications, he selects the ‘rename’ option in the place window’s context menu, and then types over the default ‘New Place’ name (Figure 4-9a-d). When Saul comes online, he will be notified of this place and can access its contents as in the earlier scenario.

4.3 Scenario 3. Creating a somewhat more complex locale

Yibo also uses Come Together for his own individual purposes, in this case, to capture, store and review some of his thesis source materials that he found on the web. At this particular moment, he wants to capture the Wikipedia ‘Collaborative software’, as it contains a nice summary of groupware systems. First, he adds this web page to his Artefact collection using the CT console. Yibo simply drags the URL of the page from the browser’s address bar (Figure 4-10a) and drops it onto “My Private Things—Not Shared In Any Place” section under the Things tab (Figure 4-10b). A context-menu pops up and he verifies that he wants to add a web media item (Figure 4-10c). The web media item with the thumbnail of the web page appears (Figure 4-10d). He then gives this item a descriptive name “Wikipedia: Groupware” by right-clicking to raise the Web Item’s context menu, selecting rename, and filling in the dialog box (Figure 4-11a-c). What this sequence illustrates is that artefacts can exist outside of places; i.e., artefacts need not be shared with others before they can be created and used.

Yibo can now use the web media item to browse that page. He drags the media item of the web page out of the CT console (Figure 4-12a). As with all media items, the web item has different levels of presentation, where the system changes the representation depending on space available (e.g., as a user resizes the standalone window). This is a form of semantic zooming. The thumbnail view shows a miniature of the web page (Figure 4-12a), while the smaller icon view shows only a titled abstract representation of the media item (a globe) (Figure 4-12b). As the size increases, the media item transforms itself into a full, interactive web browser that allows not only scrolling, but navigation from that page by selecting links (Figure 4-12c).


Collaborative software - Wikipedia, the free encyclopedia - Windows Internet Explorer
http://en.wikipedia.org/wiki/Collaborative_software
Favorites
Collaborative software - Wikipedia, the free encyclop...
Tagalog
Overview
[edit]

Collaborative software is a concept that greatly overlaps with computer supported cooperative work (CSCW). Some authors argue they are equivalent. According to Carstensen and Schmidt (1999) [1] groupware is part of Computer Supported Cooperative Work, since CSCW addresses "how collaborative activities and their coordination can be supported by means of computer systems."

Software systems such as email, calendaring, text chat, wiki, and bookmarking belong to this category. It has been suggested that Metcalfe's law — the more people who use something, the more valuable it becomes — applies to these types of software.

Whereas the more general term social software applies to systems used outside the workplace, for example, online dating services and social networks like Friendster, Twitter and Facebook, the use of **collaborative software** in the workspace creates a Collaborative Working Environment (CWE) [1].

A Collaborative Working Environment supports people in both their individual and cooperative work thus giving birth to a new class of professionals (eProfessionals), who can work together irrespective of their

a) Click and Drag the URL from Internet Explorer

Come Together
People
Things
Places
Me
My Private Things (Not Shared In Any Place)
My Shared Things (Publically Seen In My Places)
Add One Web Item
New Web Item

Drag over

Other artefacts as icons

Add to Yibo's private artefact collection

Another artefacts as icon

The XPS item for the chapter is automatically added here.

Drop

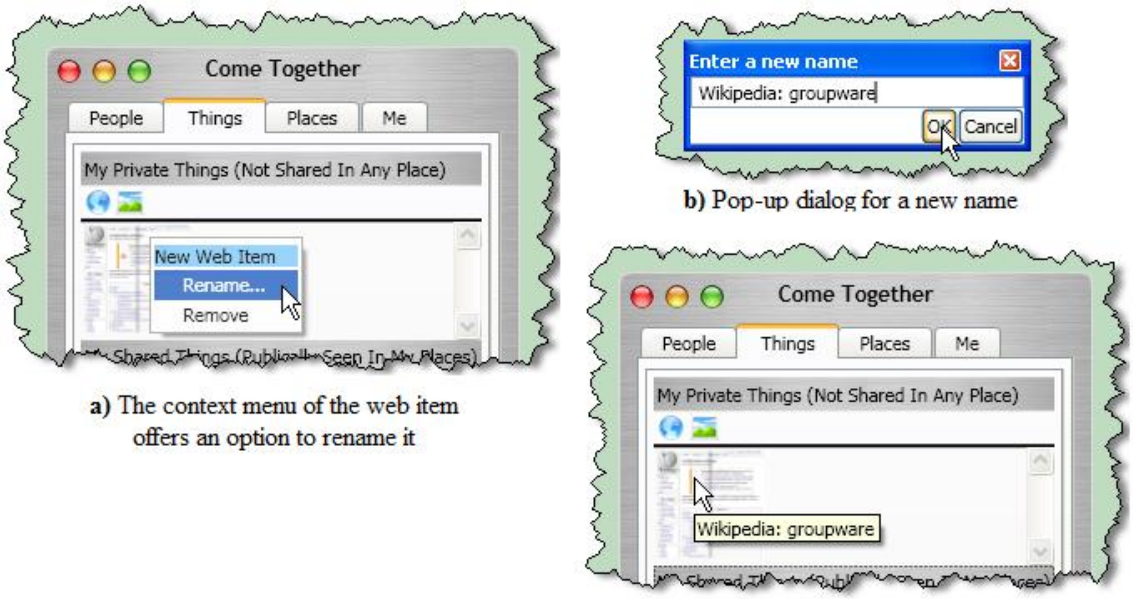
The added Web item containing the web page

b) The Things Tab, showing Yibo's private and shared artefacts

c) The context Menu offers to import the web page as a web item.

d) The web page is imported into Come Together as a private web item.

Figure 4-10 Importing a web page as a web media item into Come Together


Come Together
People
Things
Places
Me
My Private Things (Not Shared In Any Place)
New Web Item
Rename...
Remove
Shared Things (Publically Seen In My Places)
Enter a new name
OK
Cancel

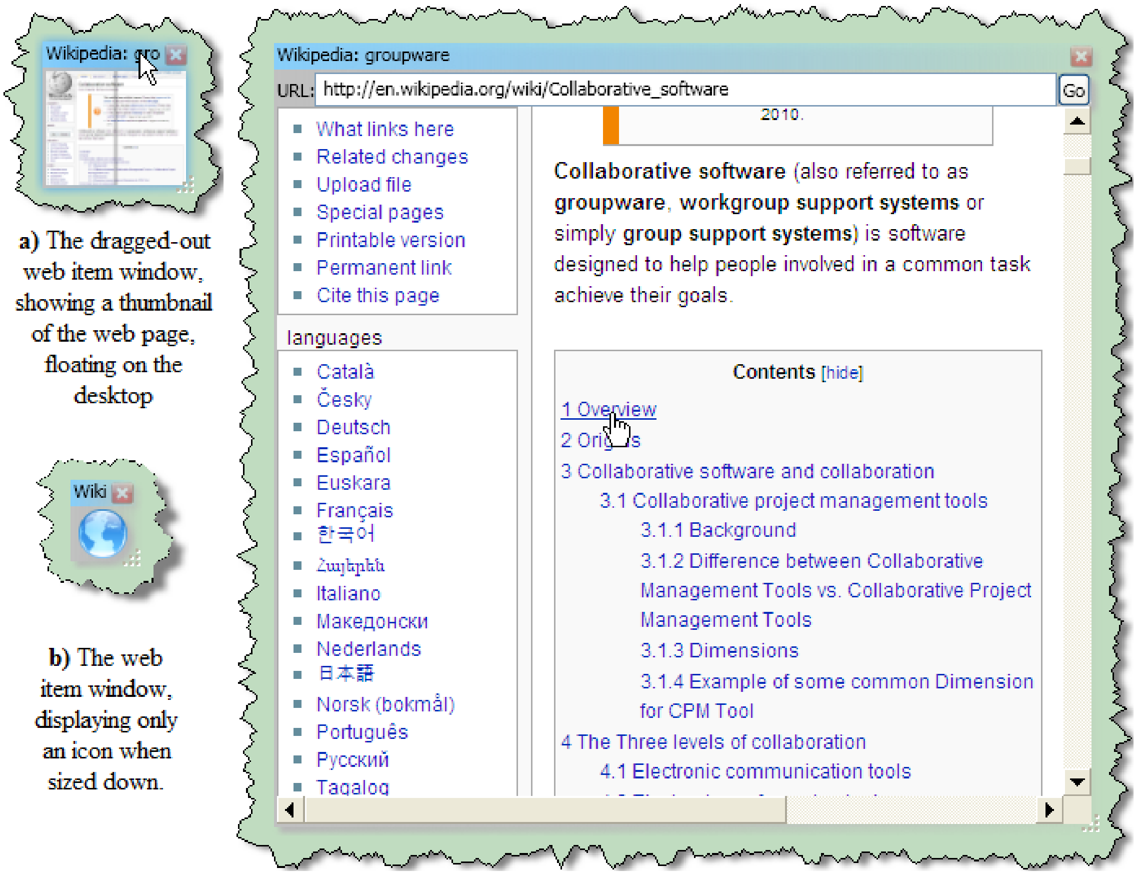
Wikipedia: groupware

a) The context menu of the web item offers an option to rename it

b) Pop-up dialog for a new name

c) The new name of the web item

Figure 4-11 Renaming a media item



URL: http://en.wikipedia.org/wiki/Collaborative_software

2010.

Collaborative software (also referred to as **groupware**, **workgroup support systems** or simply **group support systems**) is software designed to help people involved in a common task achieve their goals.

a) The dragged-out web item window, showing a thumbnail of the web page, floating on the desktop

b) The web item window, displaying only an icon when sized down.

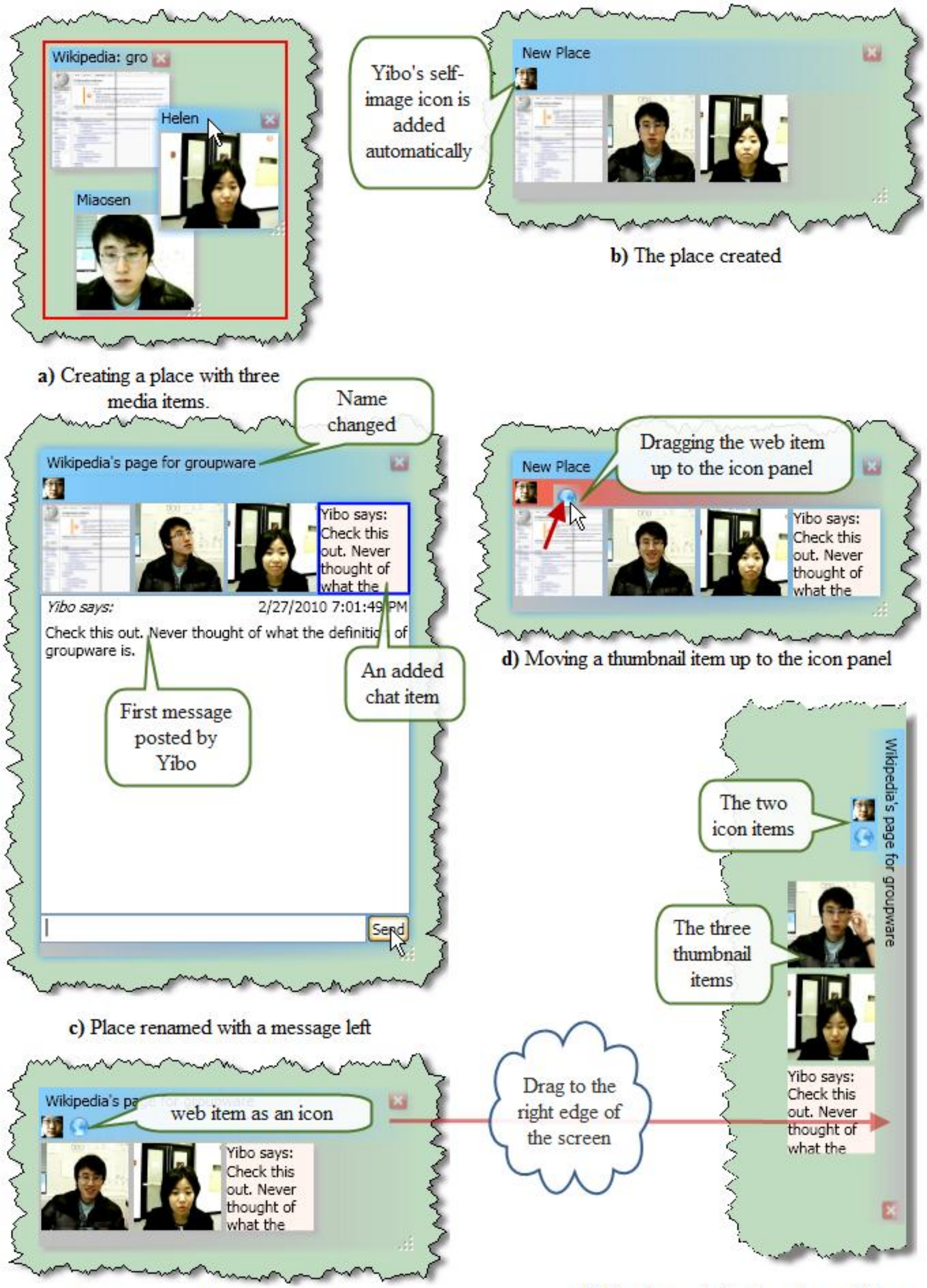
c) The web item showing a full browser of the web page when sized up

Figure 4-12 Three presentation levels of a web media item

After a while, Yibo realizes that some of his student colleagues may find this (and other groupware references) useful, so he decides to creates a place that collects these references. He notices in the CT console that Miaosen and Helen are both online, and includes them in this place. Somewhat similar to the method shown in the previous scenarios, Yibo drags out both Miaosen's and Helen's presence items (showing their video feeds), and the web item together onto the desktop (Figure 4-13a). A place window appears containing these three thumbnail items and Yibo's icon (Figure 4-13b). He then renames "New Place" into "Wikipedia's definition of groupware" (Figure 4-13c), adds a chat media item, and posts a message into it (Figure 4-13c).

Yibo then decides to adjust the way he (but not necessarily others) view this particular place, in this case by adjusting what is viewed as an icon vs. a thumbnail. All place windows have three primary sections for displaying media items: a top panel for showing icons, a middle panel for showing thumbnails, and an expandable bottom panel for tooltip grandes. Yibo is more interested in conversations about the topic and other media items people may post rather than the existing web item (which he is familiar with). Thus he transforms the thumbnail of the web item into an icon by dragging the thumbnail out of the thumbnail panel and dropping it onto the icon panel (Figure 4-13d-e).

After a few minutes of waiting, Yibo realizes that no one is immediately responding to his message; they must be busy with other things. Yibo's interest wanes, and he decided to move to a more peripheral engagement with the place he had just created. He does this by moving the place to his visual periphery, i.e., so that he can still be peripherally aware of its presence and any changes within the place. To do so, he drags the place window to the right edge of the screen: the place window is automatically transformed into a place strip with two separate panels—the icon panel and the thumbnail panel—to match the two panels in the place window for two presentation levels of media items. Using this, he can see what items are there and the changes to them in miniature. Although tooltip grande's are not available from this view, he can always grow the place back to its normal size, and/or drag out a particular media item to the desktop as a media item window to see its full view (illustrated shortly).


Wikipedia: gro
Helen
Miaosen
New Place
Wikipedia's page for groupware
Send
New Place
Wikipedia's page for groupware
Wikipedia's page for groupware

Yibo's self-image icon is added automatically

a) Creating a place with three media items.

b) The place created

Name changed

Yibo says: Check this out. Never thought of what the

Yibo says: 2/27/2010 7:01:49 PM

Check this out. Never thought of what the definitio of groupware is.

First message posted by Yibo

An added chat item

c) Place renamed with a message left

Dragging the web item up to the icon panel

Yibo says: Check this out. Never thought of what the

d) Moving a thumbnail item up to the icon panel

The two icon items

The three thumbnail items

Yibo says: Check this out. Never thought of what the

Drag to the right edge of the screen

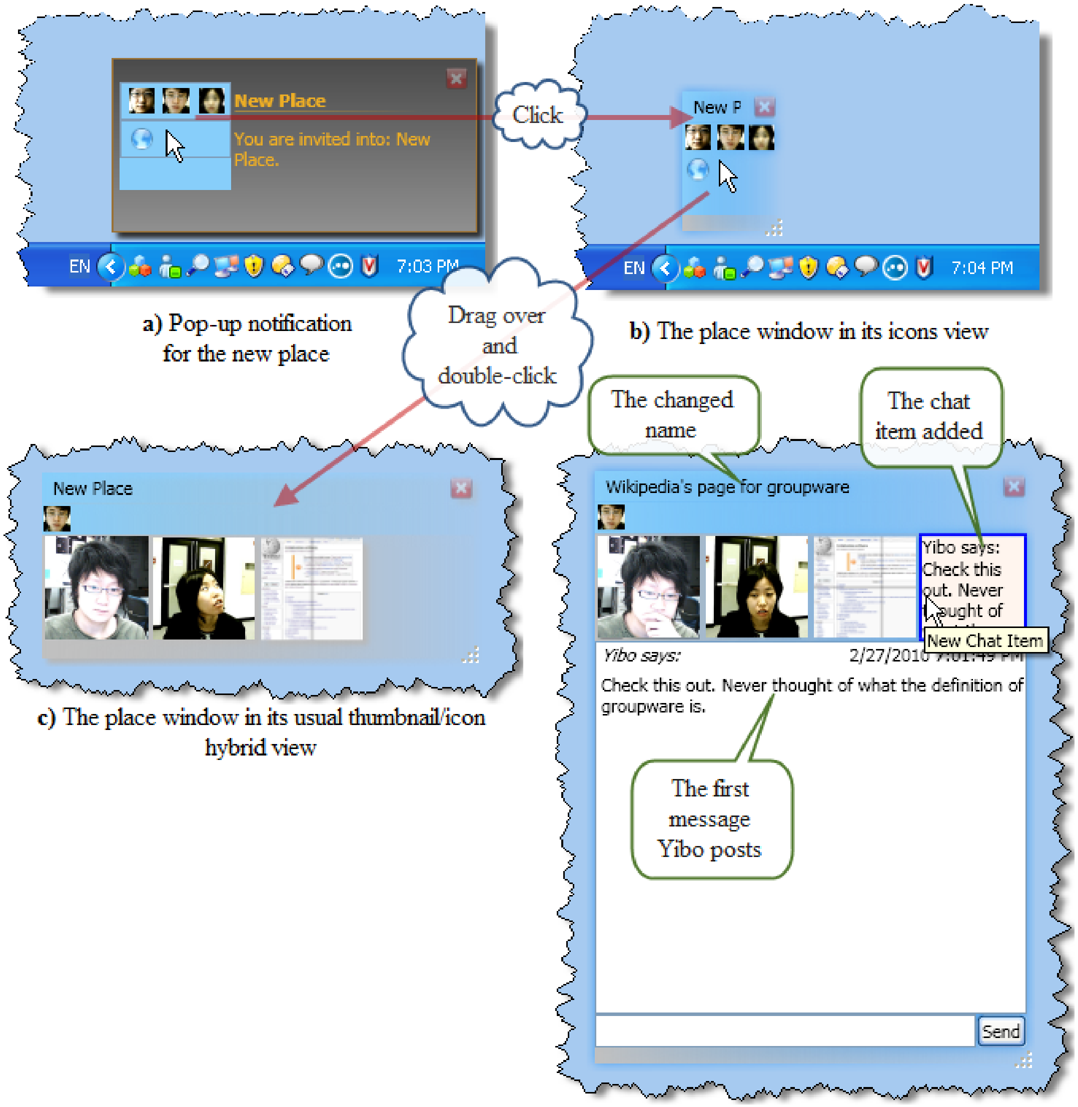
web item as an icon

Yibo says: Check this out. Never thought of what the

e) Web item moved up, staying in the icon panel

f) The place window transformed into a place strip at a vertical edge of the screen

Figure 4-13 More features: creation and customization of a place


New Place
You are invited into: New Place.
EN
7:03 PM
Click
New P
7:04 PM
Drag over and double-click
New Place
The changed name
The chat item added
Wikipedia's page for groupware
Yibo says: Check this out. Never thought of
New Chat Item
The first message Yibo posts
Send

Yibo says: 2/27/2010 7:01:49 PM
Check this out. Never thought of what the definition of groupware is.

a) Pop-up notification for the new place

b) The place window in its icons view

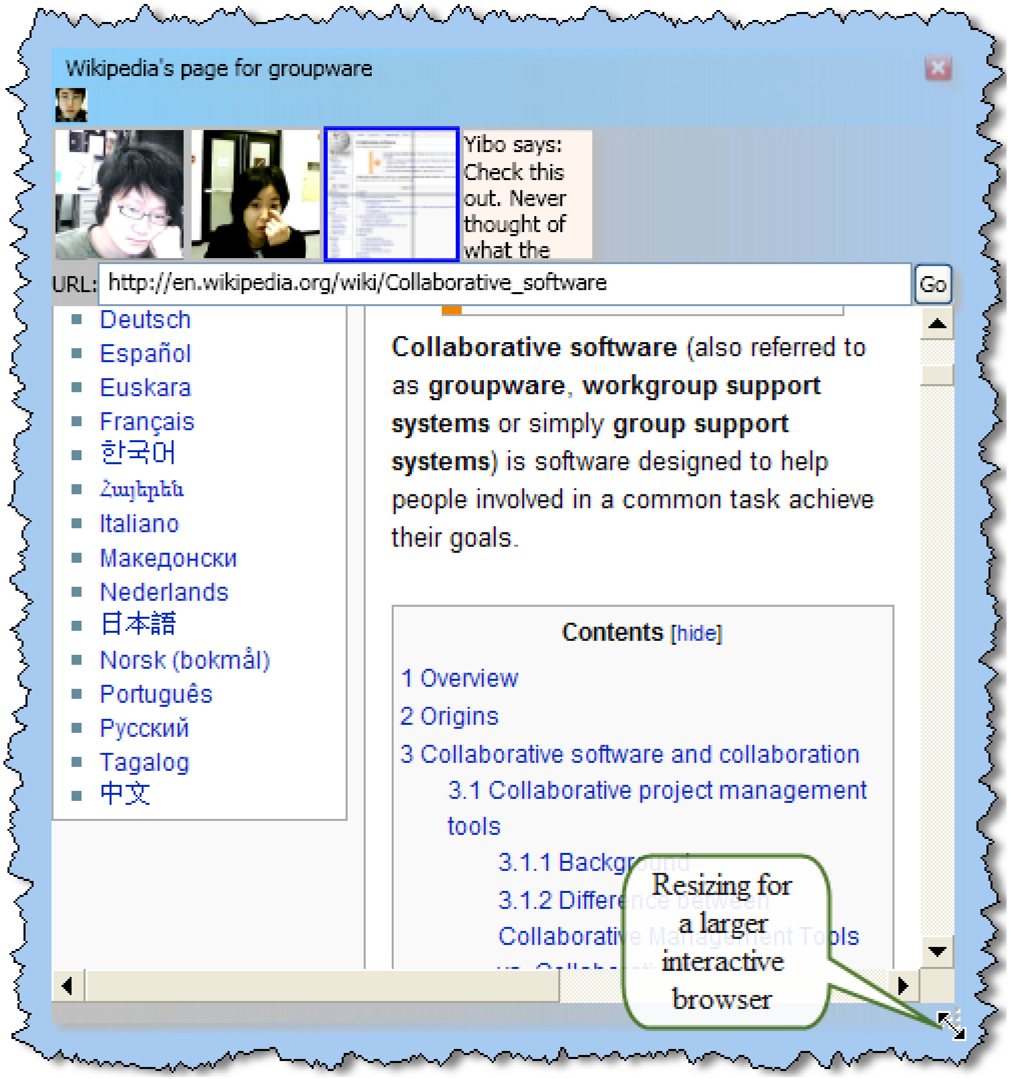
c) The place window in its usual thumbnail/icon hybrid view

d) Miaosen observing the changes Yibo makes

Figure 4-14 From a notification to a standalone place window

Now let’s consider how others can enter and use this place. Similar to the previous scenario, Miaosen sees the pop-up toast on his system notification tray telling him that he has been invited into a new place (Figure 4-14a). Miaosen could go to the Places tab and access the new place from there (as was done in Figure 4-3). Instead, he uses the faster method of just clicking on the notification: a place window appears next to his cursor, containing the icon-sized presence items of Yibo, Miasosen and Helen, and the icon-sized web item (Figure 4-14b). This representation is a compact standalone place window he was invited into, where all media items are shown in the smallest icon view *vs.* a mix of thumbnails and icons (Figure

4-13d). Miaosen drags this window to the centre of his screen and double-clicks: the icon-only view (Figure 4-14b) then grows into the regular icon/thumbnail hybrid view (Figure 4-14c). While Miaosen is doing this, Yibo is renaming the place and adding the chat item and message (Figure 4-13c). Miaosen observes these changes as they occur (Figure 4-14d). He reads the message but does not respond (Figure 4-14d) and then starts exploring the web page (Figure 4-15).



Yibo says: Check this out. Never thought of what the

URL: http://en.wikipedia.org/wiki/Collaborative_software

- Deutsch
- Español
- Euskara
- Français
- 한국어
- Հայերեն
- Italiano
- Македонски
- Nederlands
- 日本語
- Norsk (bokmål)
- Português
- Русский
- Tagalog
- 中文

Collaborative software (also referred to as **groupware**, **workgroup support systems** or simply **group support systems**) is software designed to help people involved in a common task achieve their goals.

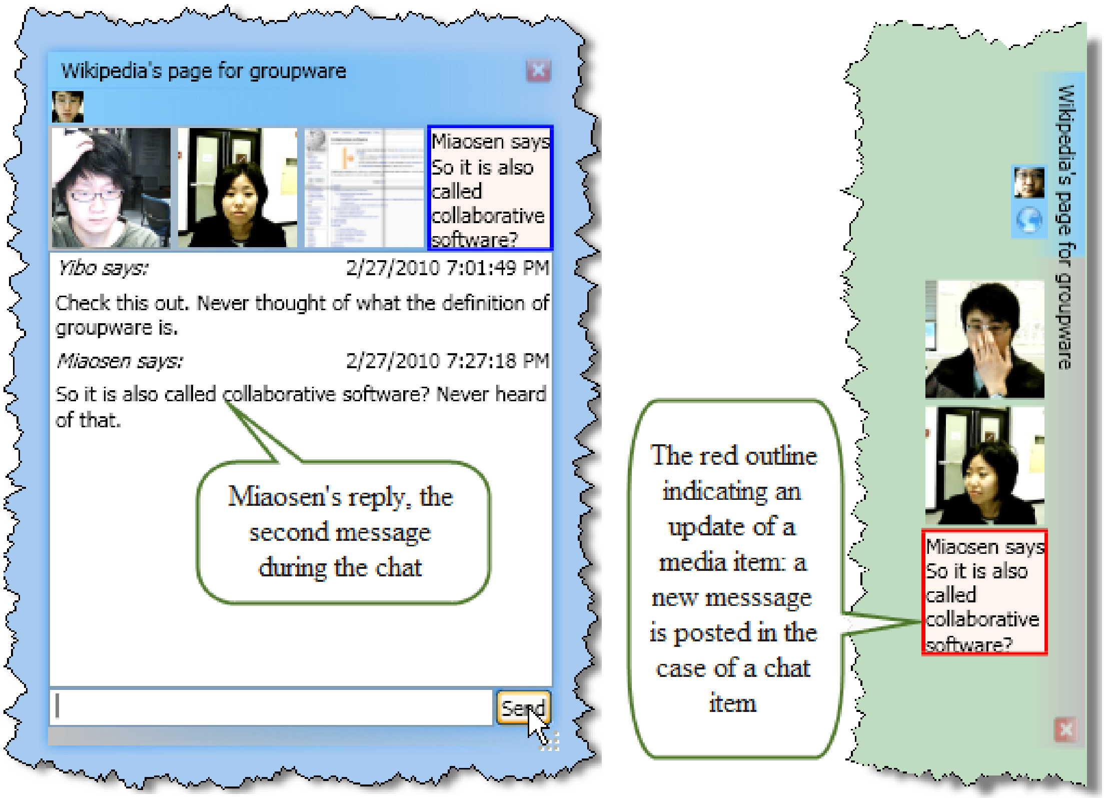
Contents [hide]

1 Overview
2 Origins
3 Collaborative software and collaboration
3.1 Collaborative project management tools
3.1.1 Background
3.1.2 Difference between Collaborative Management Tools

Resizing for a larger interactive browser

Figure 4-15 Exploring a web page from a place

More generally, like media items, a place window has multiple presentation levels. The *icon-only view* display only icon-size media items (Figure 4-14b). The *icon/thumbnail hybrid view* displays both icon- and thumbnail-sized media items (Figure 4-14c). The *interactive view* displays icon- and thumbnail-sized media items, and an interactive large version of a selected media item (Figure 4-14d, Figure 4-15). As already shown, the place strip view is a variant of the place window located at the side of the screen


Wikipedia's page for groupware
Send
Wikipedia's page for groupware

Miaosen says So it is also called collaborative software?

Yibo says: 2/27/2010 7:01:49 PM

Check this out. Never thought of what the definition of groupware is.

Miaosen says: 2/27/2010 7:27:18 PM

So it is also called collaborative software? Never heard of that.

Miaosen's reply, the second message during the chat

The red outline indicating an update of a media item: a new messsage is posted in the case of a chat item

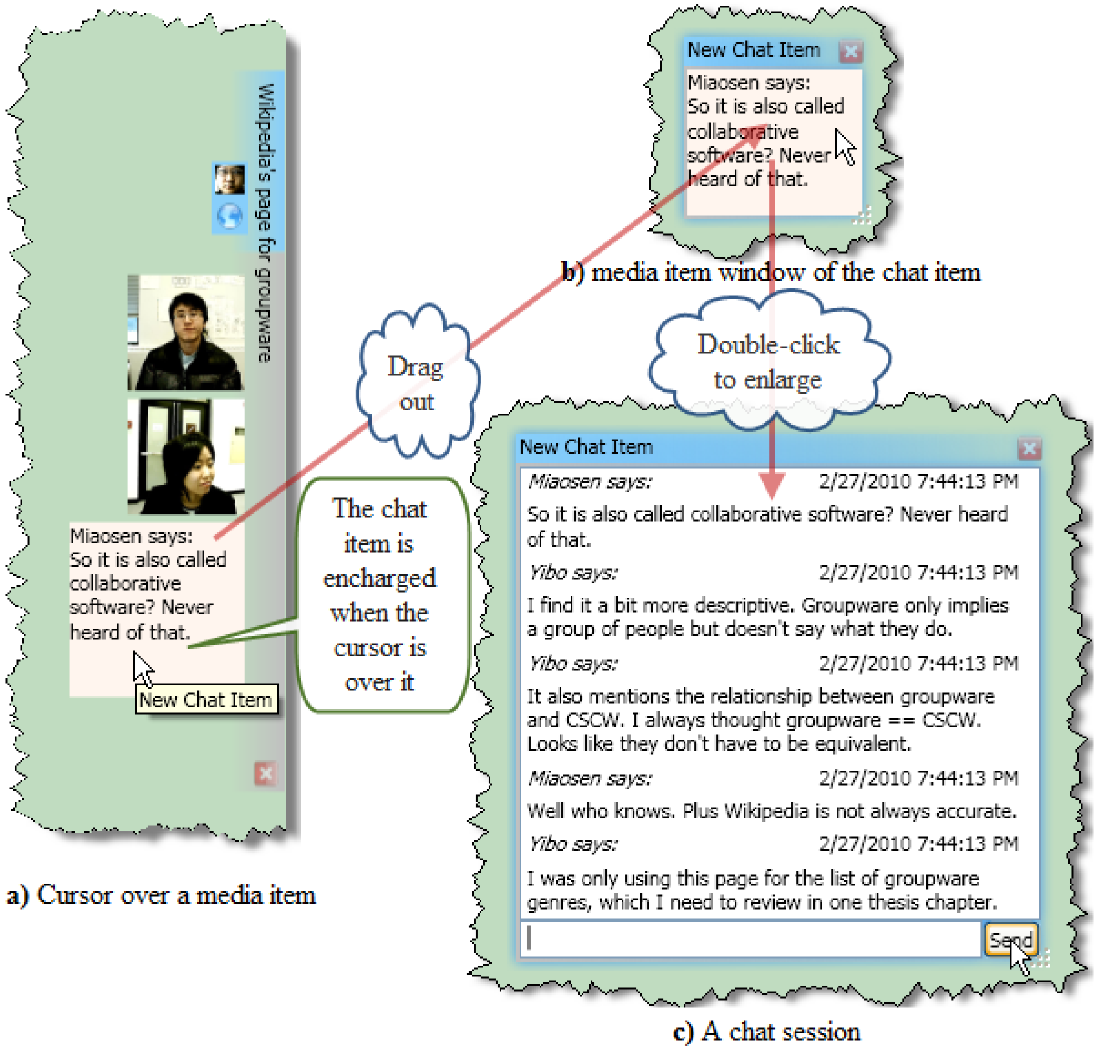
Miaosen says So it is also called collaborative software?

a) Miaosen's view of the place after posting another message

b) Yibo's view of the place, a peripheral place strip, at the arrival of a new message

Figure 4-16 Different views of the same place on Miaosen and Yibo's screen

Miaosen finished viewing the web page, and adds a message to the chat item (Figure 4-16a). While Yibo has by now moved his view of this place into the place strip view, he is still notified that there is activity in that place. In particular, the chat item is outlined in red to notify the arrival of the new message (Figure 4-16b), and the actual message (or as much as can fit) is displayed within it. Yibo notices the red border of the chat item, and moves the cursor over it. The red outline is removed and the chat item is enlarged, showing the entire message (Figure 4-17a). To read this message in full and to respond to it, Yibo drags the chat item to the centre of his screen as a standalone media item window (Figure 4-17b) and double-clicks it to expand it into its interactive view (Figure 4-17c). Yibo and Miaosen start chatting.



Miaosen says:
So it is also called collaborative software? Never heard of that.

The chat item is encharged when the cursor is over it

Drag out

Double-click to enlarge

a) Cursor over a media item

b) media item window of the chat item

c) A chat session

Miaosen says: 2/27/2010 7:44:13 PM
So it is also called collaborative software? Never heard of that.

Yibo says: 2/27/2010 7:44:13 PM
I find it a bit more descriptive. Groupware only implies a group of people but doesn't say what they do.

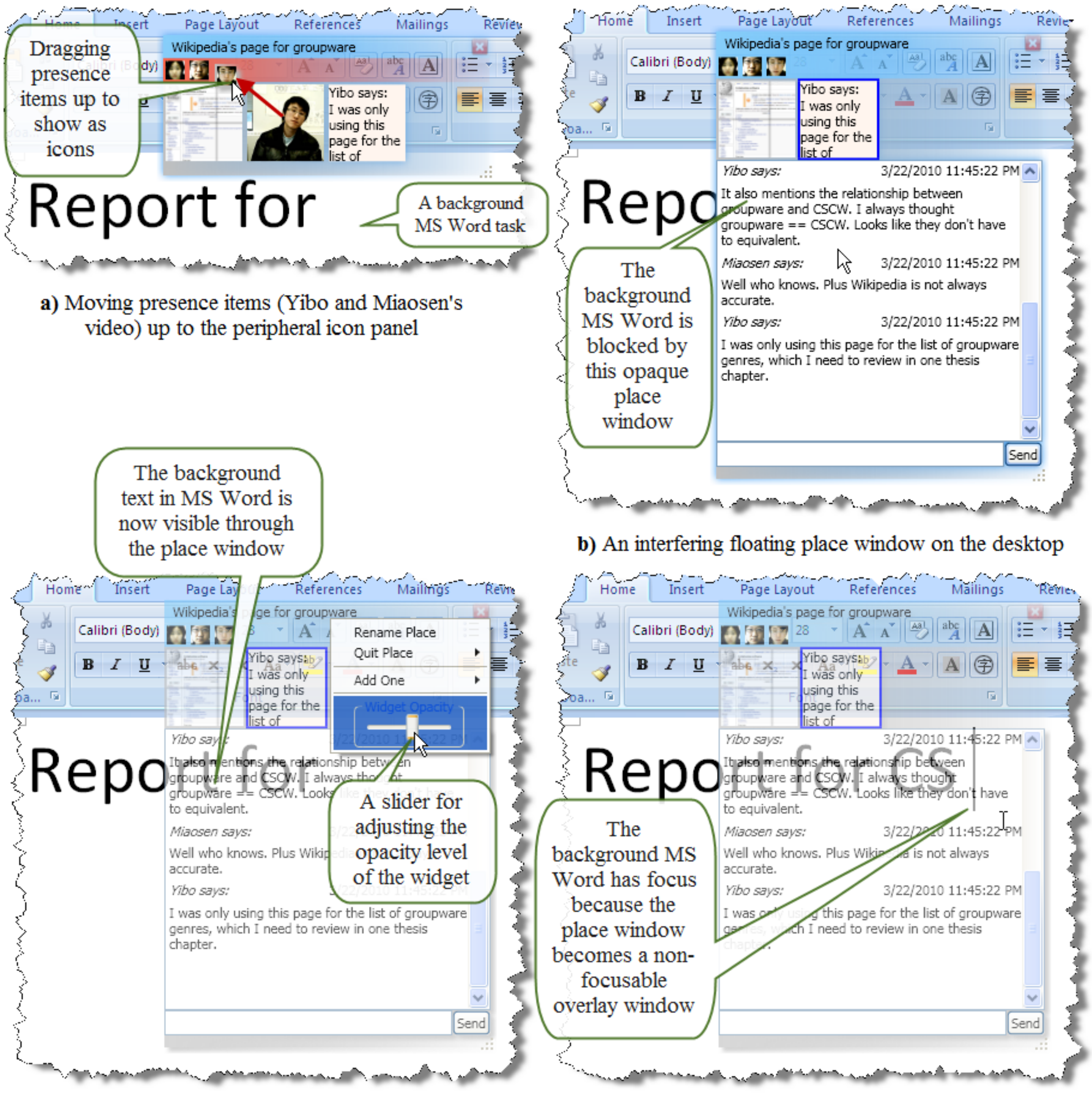
Yibo says: 2/27/2010 7:44:13 PM
It also mentions the relationship between groupware and CSCW. I always thought groupware == CSCW. Looks like they don't have to be equivalent.

Miaosen says: 2/27/2010 7:44:13 PM
Well who knows. Plus Wikipedia is not always accurate.

Yibo says: 2/27/2010 7:44:13 PM
I was only using this page for the list of groupware genres, which I need to review in one thesis chapter.

Figure 4-17 From periphery awareness to interactive chat

Concurrent to these activities, Helen is busy writing so she remains at the periphery of this place: she wants to see what is going on, but is not ready to participate. She does have the place window on her screen (Figure 4-18a), but finds the video of Yibo and Miaosen distracting. Consequenlty, she relegates Yibo and Miaosen's presence item to the icon panel (Figure 4-18a), which transforms them into static images. She monitors the conversation between Yibo and Miaosen but does not participate (Figure 4-18b).



Dragging presence items up to show as icons

A background MS Word task

Yibo says:
I was only using this page for the list of

a) Moving presence items (Yibo and Miaosen's video) up to the peripheral icon panel

The background MS Word is blocked by this opaque place window

Yibo says: 3/22/2010 11:45:22 PM
It also mentions the relationship between groupware and CSCW. I always thought groupware == CSCW. Looks like they don't have to equivalent.

Miaosen says: 3/22/2010 11:45:22 PM
Well who knows. Plus Wikipedia is not always accurate.

Yibo says: 3/22/2010 11:45:22 PM
I was only using this page for the list of groupware genres, which I need to review in one thesis chapter.

b) An interfering floating place window on the desktop

The background text in MS Word is now visible through the place window

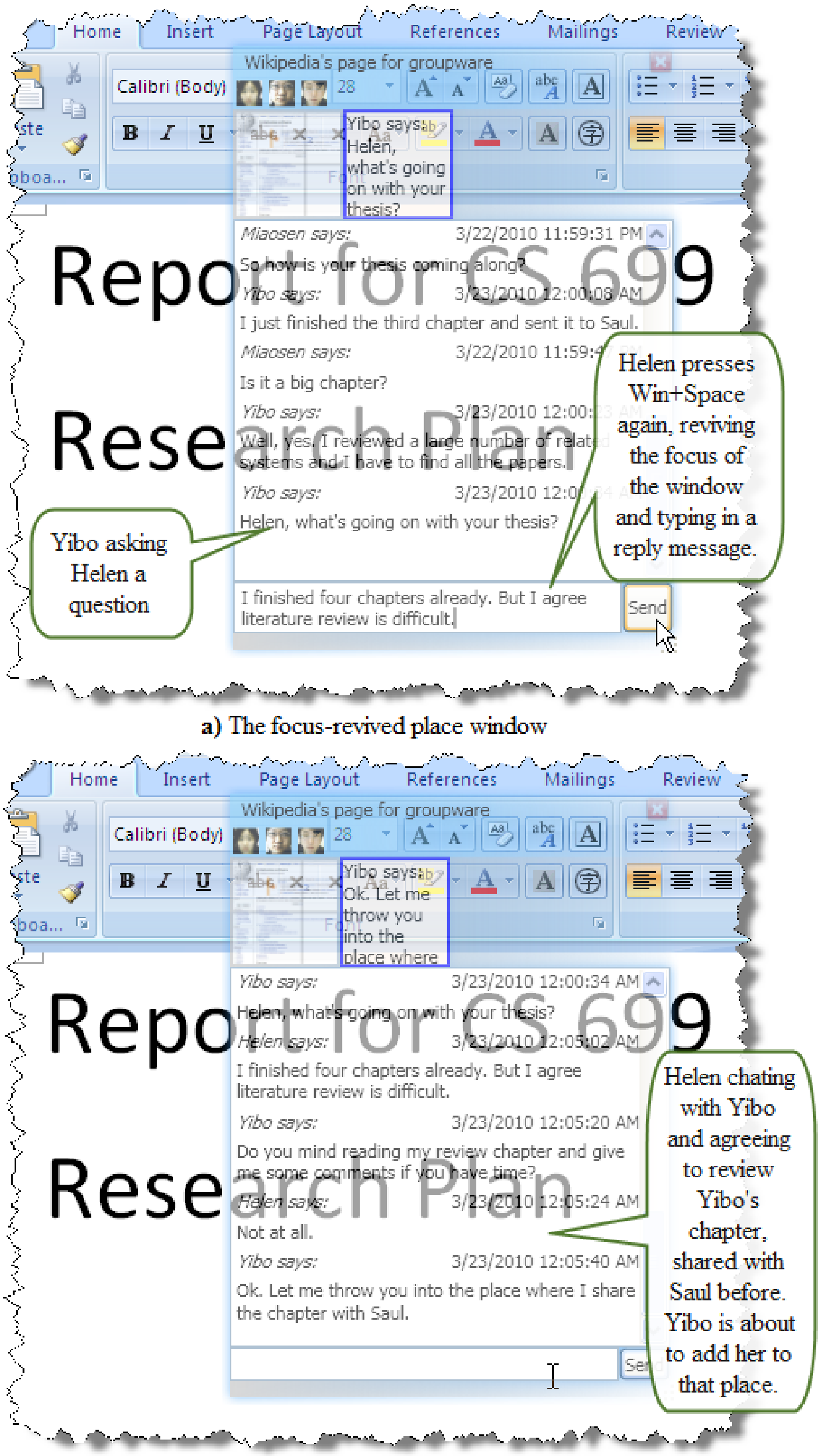
A slider for adjusting the opacity level of the widget

c) Making the place window visually transparent

The background MS Word has focus because the place window becomes a non-focusable overlay window

d) An overlay place window transparent to mouse cursor and keyboard input

Figure 4-18 Making a place window less intrusive through visual and input tranparency



Wikipedia's page for groupware

Yibo says:
Helen, what's going on with your thesis?

Miaosen says: 3/22/2010 11:59:31 PM
So how is your thesis coming along?
Yibo says: 3/23/2010 12:00:08 AM
I just finished the third chapter and sent it to Saul.
Miaosen says: 3/22/2010 11:59:4
Is it a big chapter?
Yibo says: 3/23/2010 12:00:13 AM
Well, yes. I reviewed a large number of related systems and I have to find all the papers.
Yibo says: 3/23/2010 12:0
Helen, what's going on with your thesis?

I finished four chapters already. But I agree literature review is difficult.

Yibo asking Helen a question

Helen presses Win+Space again, reviving the focus of the window and typing in a reply message.

a) The focus-revived place window

Wikipedia's page for groupware

Yibo says:
Ok. Let me throw you into the place where

Yibo says: 3/23/2010 12:00:34 AM
Helen, what's going on with your thesis?
Helen says: 3/23/2010 12:05:02 AM
I finished four chapters already. But I agree literature review is difficult.
Yibo says: 3/23/2010 12:05:20 AM
Do you mind reading my review chapter and give me some comments if you have time?
Helen says: 3/23/2010 12:05:24 AM
Not at all.
Yibo says: 3/23/2010 12:05:40 AM
Ok. Let me throw you into the place where I share the chapter with Saul.

Helen chating with Yibo and agreeing to review Yibo's chapter, shared with Saul before. Yibo is about to add her to that place.

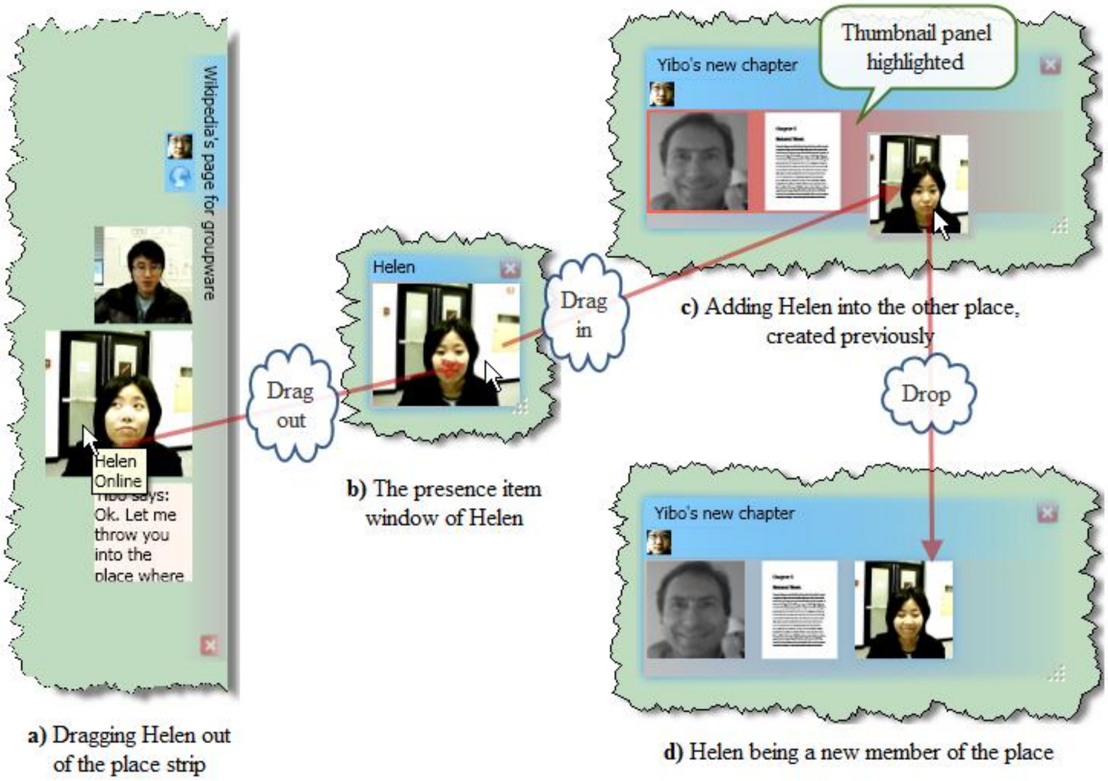
b) The conversation between Yibo and Helen

Figure 4-19 The place window regains focus

However, Helen has a small screen. Because the place window is displayed always atop of all other application winodws, it interferes somewhat with her foreground task of report-writing in Microsoft Word. To better balance awareness at the periphery, she adjusts the transparency of the place window via a slider in its context menu (Figure 4-18c) to make it just readable, while at the same time toggling a ‘input focus’ option (Win+Space) to ensure that the Place window will not receive mouse or keyboard focus (Figure 4-18d). As a result, Helen can continue writing her report, while still monitoring the conversation.

It is fairly easy for Helen to move between the center and periphery of this place. For example, Yibo conversation with Miaosen turns to their progress in their thesis writing, and Yibo aks Helen about her progress (Figure 4-19a). Helen sees this message to her, and decides to reply. Using Win+Space, she toggles the input focus and types a message (Figure 4-19a).

The converstaion continues, and as a consequence Helen agrees to review Yibo’s thesis chapter, as shared with Saul in the beginning of this scenario (Figure 4-7). To facilitate this, Yibo adds Helen to that “Yibos new chapter” place. He does this by dragging Helen’s presence item out of the current place in place strip (Figure 4-20a) and drops it onto the ‘Yibos new chapter’ place window (Figure 4-20b, Figure 4-20c). The thumbnail panel of the place is highlighted indicating Helen is about to be added (Figure 4-20c). Yibo releases his mouse button and Helen’s presence item is now in the thumbnail panel of the place window (Figure 4-20d) and she is invitied in via the pop-up toast notification. This place also appears in Helen’s “My Places” section of her Places tab (Figure 4-21).



Yibo says: Ok. Let me throw you into the place where

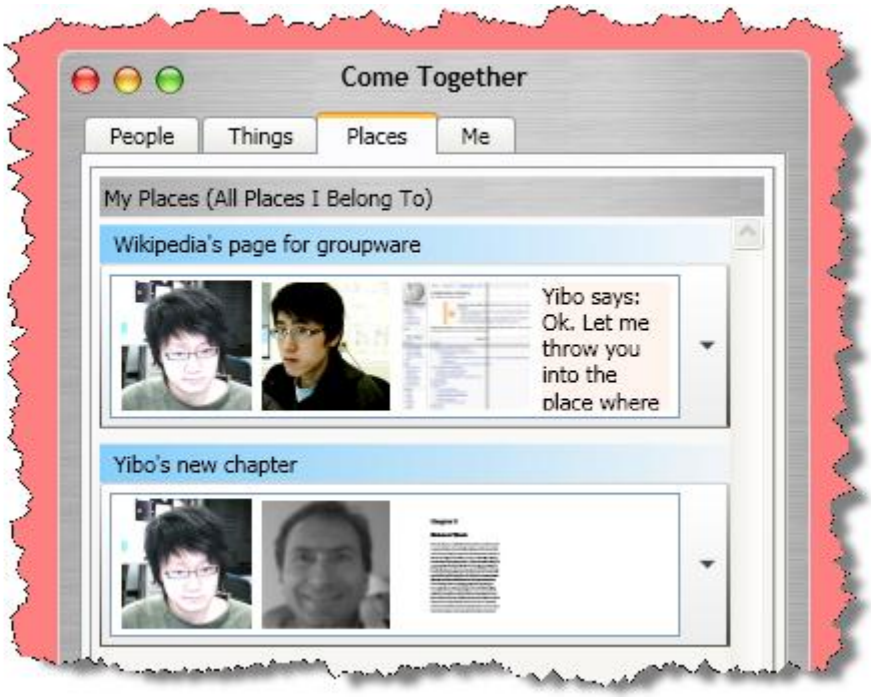
a) Dragging Helen out of the place strip

b) The presence item window of Helen

c) Adding Helen into the other place, created previously

d) Helen being a new member of the place

Figure 4-20 Inviting a person into an existing place

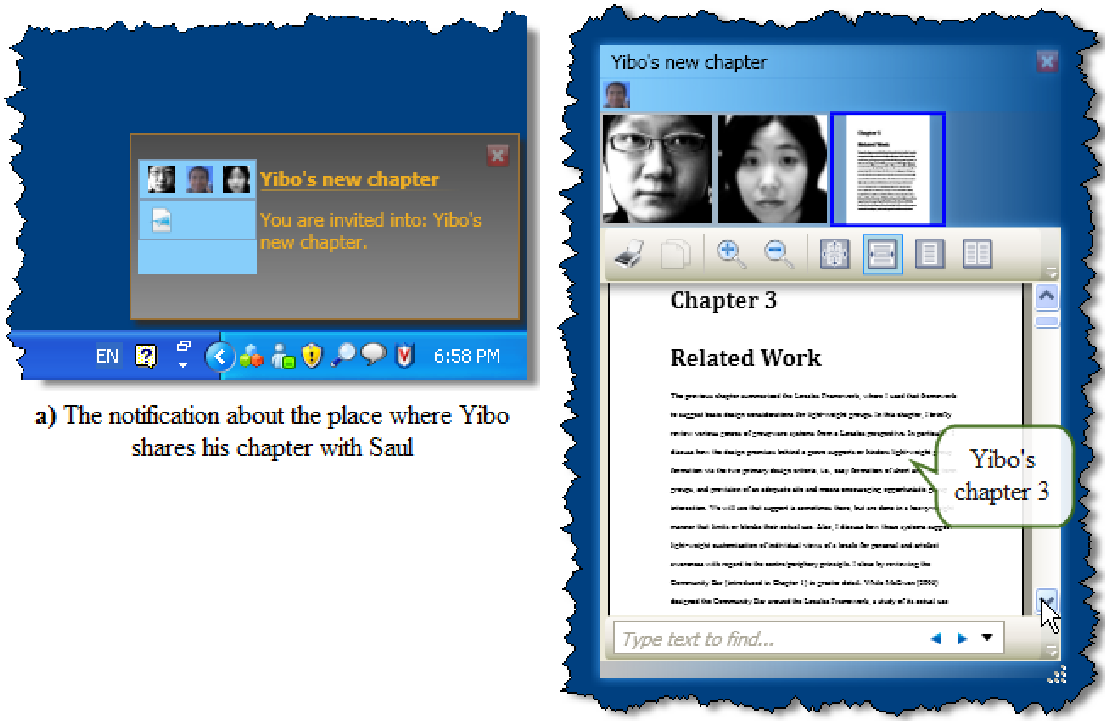

Come Together
People
Things
Places
Me
My Places (All Places I Belong To)
Wikipedia's page for groupware
Yibo's new chapter

Yibo says: Ok. Let me throw you into the place where

Figure 4-21 Helen's Places tab, showing the two places

4.4 Scenario 4. The asynchronous collaboration, in scenario 2, continued

Much later, Saul logs on. He sees the notification pop-up toast for the “Yibo’s new chapter” place via the icon-only view (Figure 4-22a), drags it onto his desktop, and starts viewing Yibo’s chapter (Figure 4-22b).



Yibo's new chapter

You are invited into: Yibo's new chapter.

Chapter 3

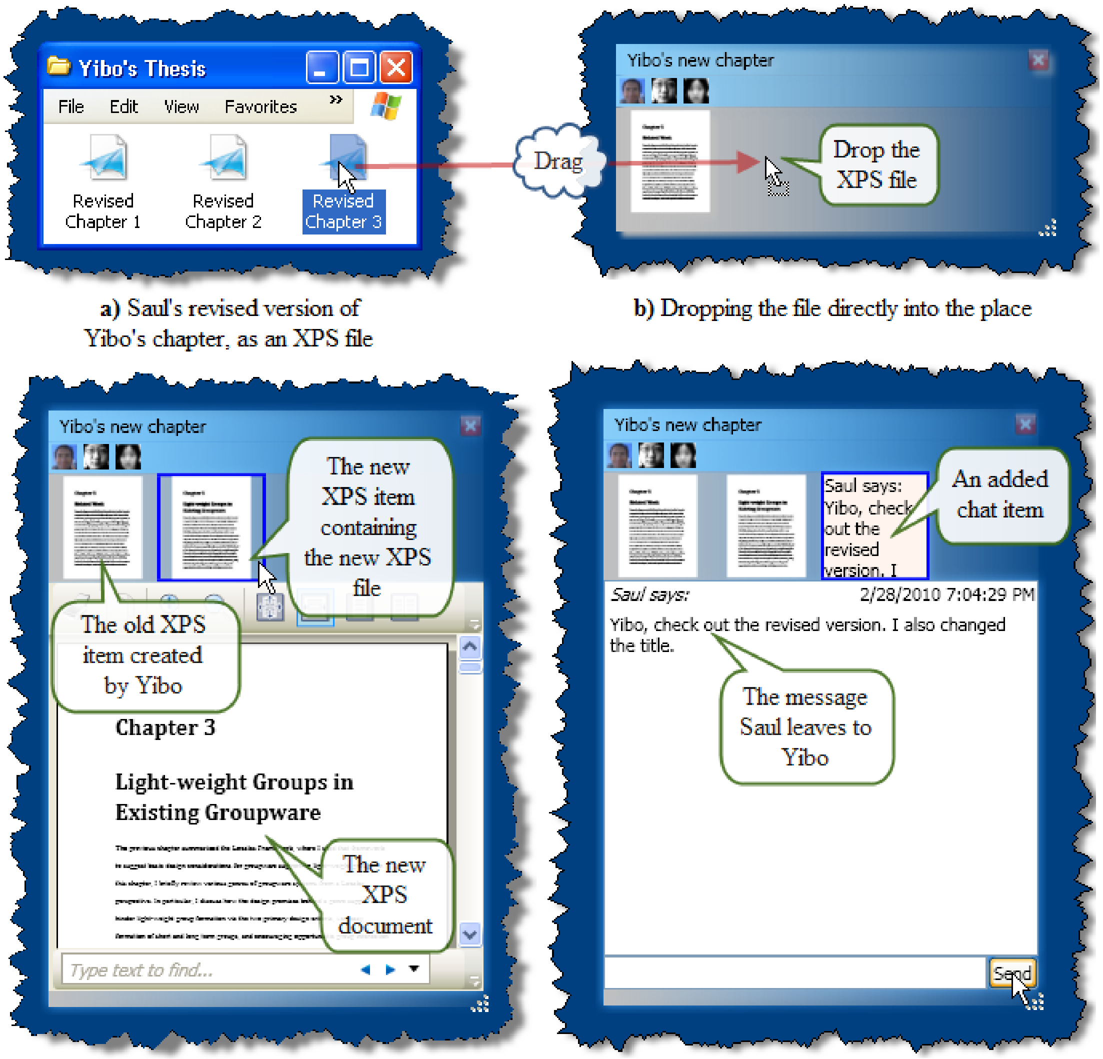
Related Work

a) The notification about the place where Yibo shares his chapter with Saul

b) The XPS document viewed in the place window.

Figure 4-22 Saul sees the place where Yibo shares the chapter

He saves this chapter to his file system, and revises it (the current document viewer item does not allow editing). He then drags and drops this new chapter version onto this place, where it appears as a new document viewer item. (Figure 4-23a, Figure 4-23b). Saul adds another chat item and leaves a message to Yibo about his changes (Figure 4-23d).



Chapter 3

Light-weight Groups in Existing Groupware

Saul says: Yibo, check out the revised version. I

Saul says: 2/28/2010 7:04:29 PM

Yibo, check out the revised version. I also changed the title.

a) Saul's revised version of Yibo's chapter, as an XPS file

b) Dropping the file directly into the place

c) The new XPS item, containing Saul's revised version of Yibo's chapter

d) Saul's posted message in a new chat item informing Yibo about the new version of the chapter

Figure 4-23 Saul adding another artefact, his revised version of Yibo's chapter

4.5 Summary

The four scenarios illustrated the following main concepts:

- Locale formation requiring no prior configuration of the site and/or means.
  - One can drag together two or more media items to form a place. Therefore, a locale container does not have to be created before adding people and artefacts.
  - When a place is created, it does not require a name. A place can be named

and/or renamed at any later time.

- Locale creation through implicit actions.
  - Dropping an artefact onto a person's media item implicitly creates a new place.
- No hard boundary: joining a locale without invitation or moderation.
  - CT does not require a user to accept an invitation to join a place. It does not impose any moderation from an administrator to allow another user to join a place either.
- No prescribed social protocol in a locale: participating without imposed limitation.
  - Technical protocols or roles are not imposed in a place. Every participant in a place enjoys the same privileges. Every member can invite people into a place, add artefacts to a place, modify the place, and reconfigure its settings.
- People and artefacts represented equally, as media items, in and out of places.
  - People and artefacts are contained uniformly in media items, with three presentation levels, collected in place windows, contained in individual place strips on the desktop edges, listed under various panels in the CT Console.
- Different presentation levels/forms of media items and places along the centre-periphery continuum
  - Media items have three presentation levels: icon view, thumbnail view, full view.
  - Places have two presentation levels—icons-only view, icon/thumbnail hybrid view—and two forms—standalone place windows, place strips.
  - One can adjust the transparency and toggle input focuses of place windows or media item windows.
- Customizable individual views of people, artefacts, and places in different

presentations levels/forms, at different locations.

- One can customize her own view of the CT Console by placing media items for people and artefacts in different presentation levels.
- One can place the media item windows and place windows at different locales of the screen, in different presentation levels and forms.
- Each user’s view is individual and does not affect the view of another user’s.

In the next chapter, we will see the other features of Come Together by providing a general description of the system through its building blocks.

.

Chapter 5. **Come Together Deconstructed**

The previous chapter introduced Come Together and illustrated its main features through four scenarios. However, they by no means provided full coverage of all CT capabilities and details. As a complement to the scenarios, this chapter deconstructs CT into its components. It also explains (when needed) how these components meet the design goals as derived from the Locales Framework and other groupware systems in the first three chapters.

I begin with a complete picture of the CT client user interface. Next, I provide an overview of how the user interface components are combined to support three main features: light-weight locales, centre-periphery differentiation, and people-artefact integration. I conclude by describing all the stock media items currently included in CT.

5.1 The Come Together Client User Interface

User interface components in Come Together belong to three main domain categories: *people, artefacts* and *places*.

- *People* are the users of the system. There are two kinds of people: the user, and the other people the user can interact with.
- *Artefacts* are of two types. First are the digital artefacts that people can create and share, such as photos, web pages, documents, and screen snapshots. Second are the communication channels that are the means that let people communicate, such as a textual chat.
- *Places* represent locales, where they provide the sites that contain the means for group interactions.

People and artefacts are represented and manipulated primarily as *media items*, i.e., a container that holds the view(s) of one person or one artefact. A place is just a container, where it can hold one or more media items and present alternate views of these media items.

Yet another aspect of Come Together is how a user sees and lays out media items and places to fit their particular individual needs. It does this through three primary mechanisms.

- *CT console* collects all people and places into a single tabbed window. It is the primary means by which a user can view, navigate and control aspects of them. It has four tabs: People, Things, Places, and Me (Figure 5-1), each which will be explained shortly.
- *The desktop* lets people drag out media items and/or places as standalone windows.
- *The place strips*, located at the always-on-top layer on the edges of the desktop (perhaps across multiple monitors), let people move places to the periphery of their attention.

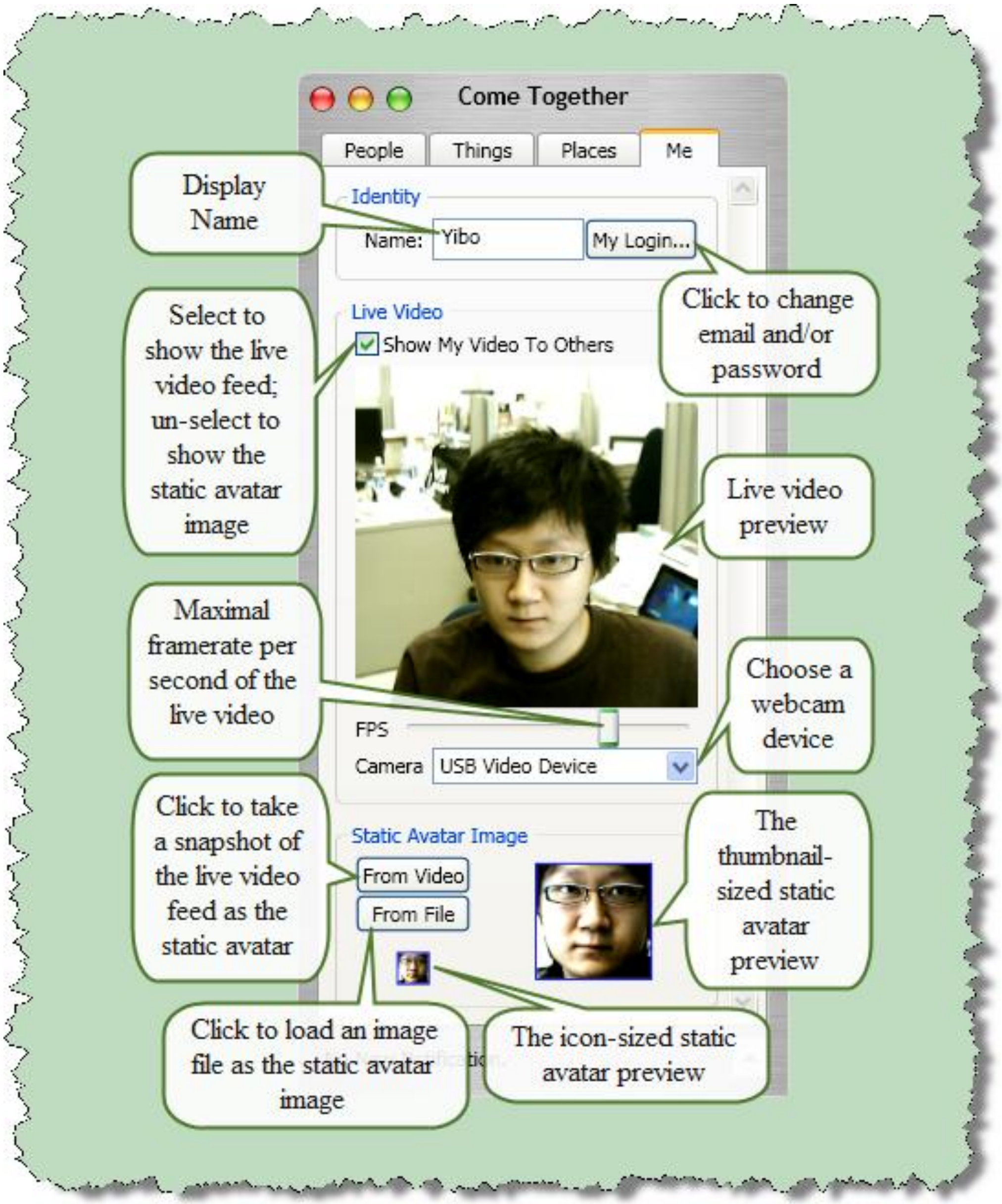
5.1.1 People

There are two types of people: the user (called 'Me') and other users on the system.

'Me' in the CT console. Come Together allows a user to adjust how others can see him or her. A user does this via the 'Me' tab on the CT console. As illustrated in Figure 5-1, the user can configure various account settings and personal profile information about oneself including:

- one's display name;
- a static 'avatar' image (a photo from either a file or captured from live video) that is displayed when video is not enabled;
- a live video feed, where one can adjust settings like maximum frame rate;
- an option to specify whether the static avatar image or live video is shown to others.

Other users in the CT console. The People tab of the CT console lets one view all people on the system, and create a buddy list of a subset of those people. Figure 5-2 illustrates this tab. Its default media item view (Figure 5-2a) displays a presence media item for every person. It is divided into two parts. The upper “My Buddies” section collects the current user’s selected buddies, while the lower “All People” section shows every registered user known to the system. One can adjust the relative sizes of the two sections by dragging the “All People” title bar splitter.



Name: Yibo

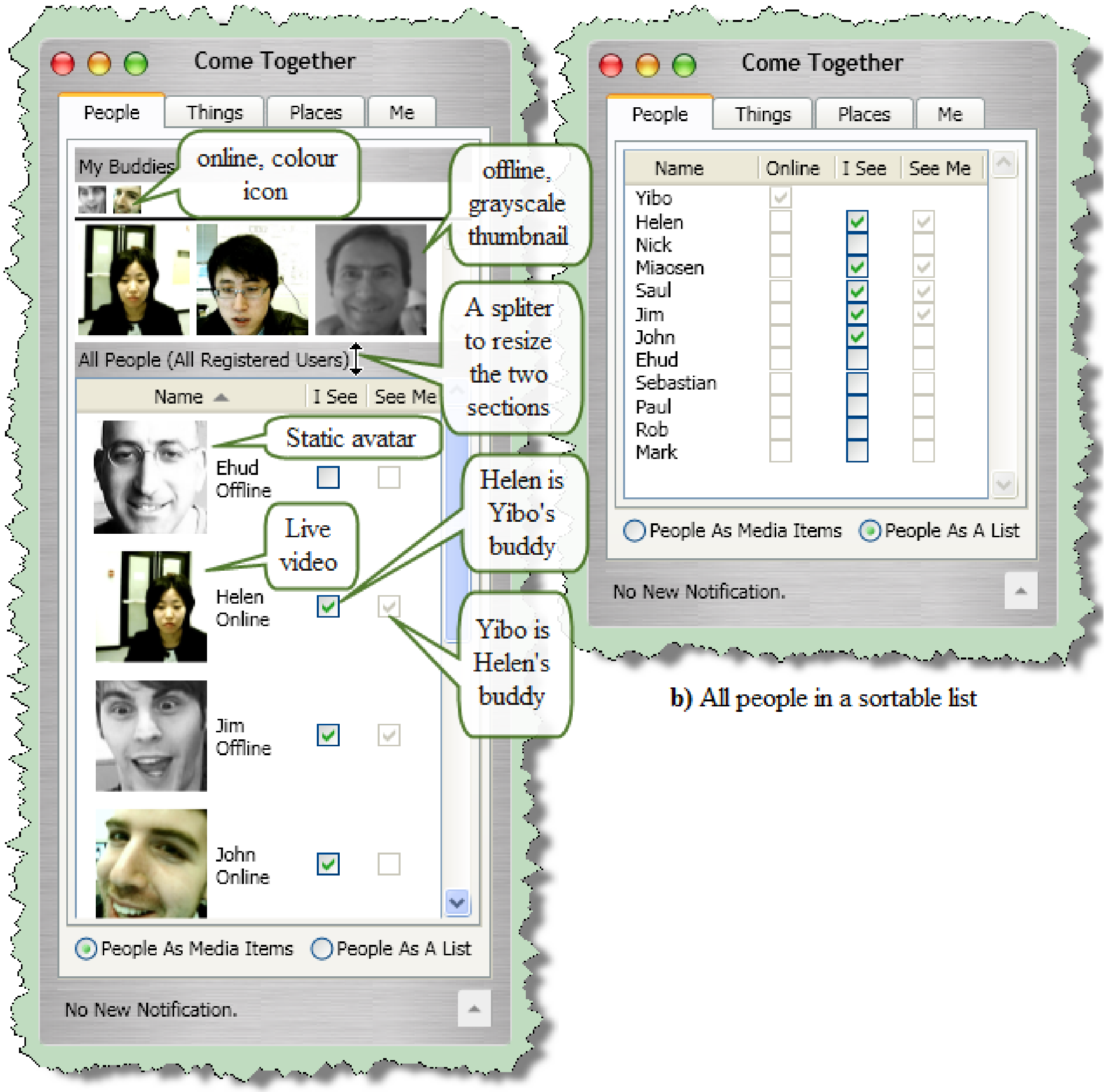
☑ Show My Video To Others

Camera: USB Video Device

Come Together
People
Things
Places
Me
Identity
My Login...
Live Video
FPS
Static Avatar Image
From Video
From File
Display Name
Click to change email and/or password
Select to show the live video feed; un-select to show the static avatar image
Live video preview
Maximal framerate per second of the live video
Choose a webcam device
Click to take a snapshot of the live video feed as the static avatar
The thumbnail-sized static avatar preview
Click to load an image file as the static avatar image
The icon-sized static avatar preview

Figure 5-1 Yibo’s Me tab

As can be seen, each row entry in the “All People” section represents a person via a person’s thumbnail-sized media item (live video or static avatar, depending on her account setting mentioned earlier) and display name. Selecting the left checkbox (in the “I See” column) changes that person into a buddy, where it would then appear in the upper section. Buddies need not be reciprocal. Thus the check box on the right (in the "See Me" column) indicates whether that person also has selected the current user as a buddy.



Name	Online	I See	See Me
Yibo	☑		
Helen	☐	☑	☑
Nick	☐	☐	☐
Miaosen	☐	☑	☑
Saul	☐	☑	☑
Jim	☐	☑	☑
John	☐	☑	☐
Ehud	☐	☐	☐
Sebastian	☐	☐	☐
Paul	☐	☐	☐
Rob	☐	☐	☐
Mark	☐	☐	☐

b) All people in a sortable list

a) Yibo's buddies and all people as media items

Figure 5-2 Yibo's People tab: two views

The "My Buddies" section (Figure 5-2a, top) contains two panes: an icon pane (at the very top), and a thumbnail pane (just below). Both show the buddies of the current user as a presence media item, i.e., all people whose "I see" checkboxes are selected in the "All People" section. By default, the presence item appears in the 'thumbnail pane', where it displays that person's live video feed or her static image as a 64 x 64 thumbnail. The thinner icon pane

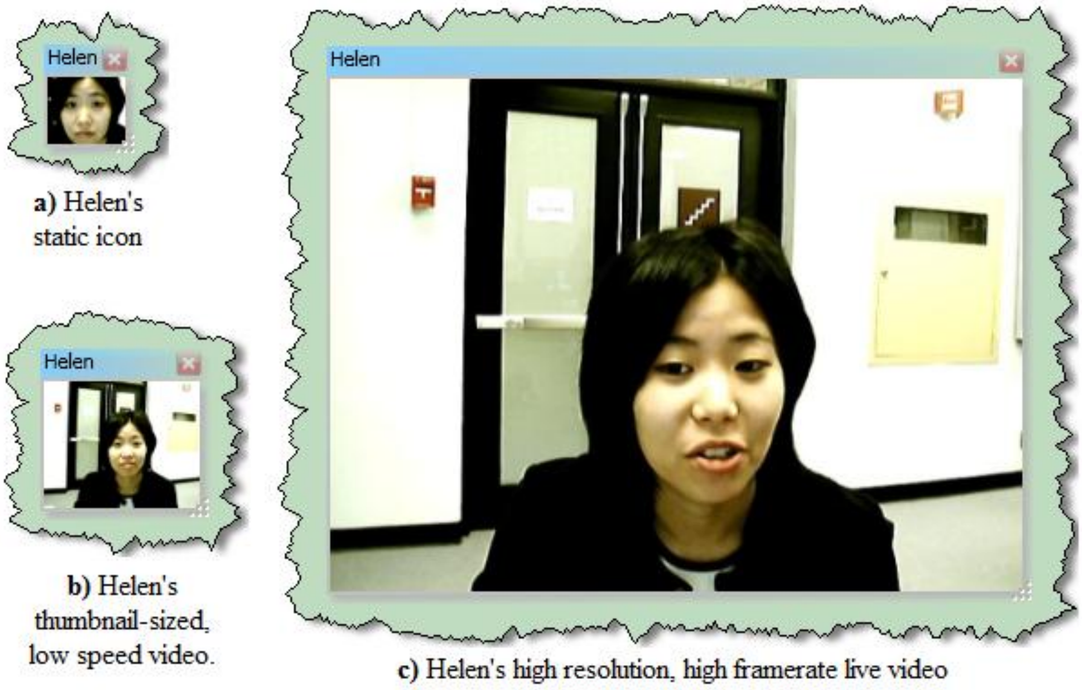
(Figure 5-2a, very top), shows buddies as a static smaller sized 16x16 image. A user can move one of their buddies between these sections by dragging and dropping a person's presence item, e.g., a person of less interest can be moved from the main panel to the upper panel. The two views in the section thus supports two presentation levels—the icon view and the thumbnail view—of people's presence items, forming an individual view of a selected sub-set of all users in the system.

Other aspects of the People tab are more routine. A person can customize where others are in the sequence of media items in either panel via drag and drop. To add a buddy, one can also drop a presence item into it. Offline people are shown as a grayscale, static image. Additionally, a user can switch to a more compact sortable list view of all people via a radio button at the bottom (Figure 5-2b).

'People' on standalone media item windows. As illustrated in the scenarios of the previous chapter, a user can drag out a person's presence item from the people's tab of the CT console onto the desktop space, where it appears as a media item within a standalone window titled by the person's display name (Figure 4-1b). As with all media items, the people media item has three presentation levels (Figure 4-12), where the presentation level depends on the size of the containing window as adjusted by the end user. When the window size is smaller than 64x64 pixels, the icon view is displayed (Figure 5-3a), while if it is between 64x64 and 256x256 the thumbnail view is shown. Both icon and thumbnail views are similar to those of the "My Buddies" panel (Figure 5-3b), except that their contents are scaled to fit. As well, when video appears in the thumbnail, it is played at a slow frame rate. When the window size is greater than 256 x 256, a third full view is displayed, usually as that person's live video shown at higher resolution and at the higher frame rate specified in that person's account settings (Figure 5-3c). For all views, the end user can adjust the opacity of the media item window via a slider in the context menu (Figure 5-4). The idea is that transparent windows move that person somewhat towards the periphery of attention, while fully opaque windows are more in the foreground.

As also illustrated in the scenarios, a person's media item window can be added into one or more places. This is done by either dropping the presence item into an existing place, or by creating a place around it via the media item window's context menu (Figure 4-20).

When in a place, the three representations of a person are dependent on other factors. As with the Person tab, the icon and thumbnail representation are displayed in the icon and thumbnail pane, while the full view is shown in a lower pane as a person mouses over the icon or thumbnail representation, where the user can fix it into place by clicking it.



a) Helen's static icon

b) Helen's thumbnail-sized, low speed video.

c) Helen's high resolution, high framerate live video

Figure 5-3 Helen's presence item window in three sizes: three presentation level

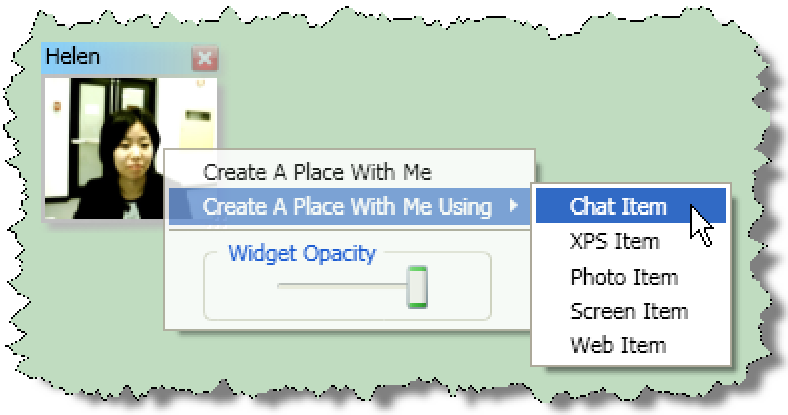
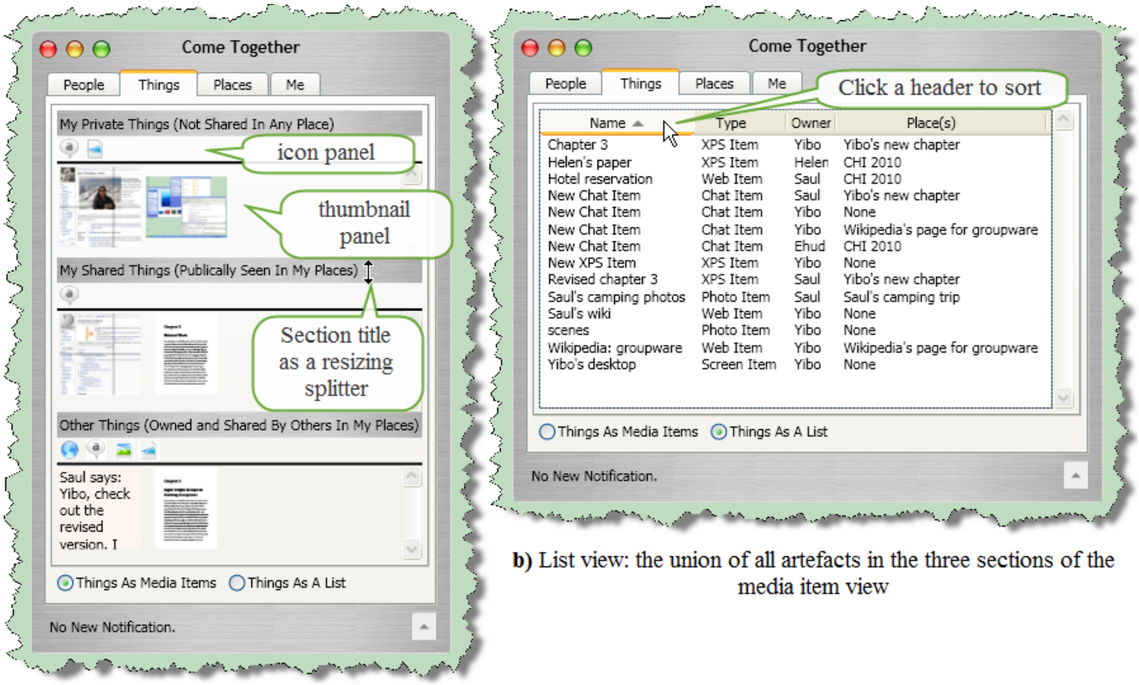


Figure 5-4 Context menu of a presence item window

5.1.2 Artefacts

An artefact is called a *thing* (its non-technical term) in CT's user interfaces, being either an artefact (be they private or shared with others), or a communication facility. Interacting with artefacts is done via media items either in the CT console or through standalone media item windows, and thus is very similar to the way a user interacts with *people.*

Artefacts in the CT console. As with the people tab, an end user can view *things* within the CT console in two ways: the default media items view (Figure 5-5a) which displays artefacts in sections, and the alternative list view (Figure 5-5b) which lists the union of all artefacts in a compact sequential and sortable list.


Come Together
People
Things
Places
Me
My Private Things (Not Shared In Any Place)
icon panel
thumbnail panel
My Shared Things (Publically Seen In My Places)
Section title as a resizing splitter
Other Things (Owned and Shared By Others In My Places)
○ Things As Media Items ○ Things As A List
No New Notification.
Come Together
People
Things
Places
Me
Click a header to sort
○ Things As Media Items ● Things As A List
No New Notification.

Saul says: Yibo, check out the revised version. I

Name ▲	Type	Owner	Place(s)
Chapter 3	XPS Item	Yibo	Yibo's new chapter
Helen's paper	XPS Item	Helen	CHI 2010
Hotel reservation	Web Item	Saul	CHI 2010
New Chat Item	Chat Item	Saul	Yibo's new chapter
New Chat Item	Chat Item	Yibo	None
New Chat Item	Chat Item	Yibo	Wikipedia's page for groupware
New Chat Item	Chat Item	Ehud	CHI 2010
New XPS Item	XPS Item	Yibo	None
Revised chapter 3	XPS Item	Saul	Yibo's new chapter
Saul's camping photos	Photo Item	Saul	Saul's camping trip
Saul's wiki	Web Item	Yibo	None
scenes	Photo Item	Yibo	None
Wikipedia: groupware	Web Item	Yibo	Wikipedia's page for groupware
Yibo's desktop	Screen Item	Yibo	None

a) Media item view: three sections

b) List view: the union of all artefacts in the three sections of the media item view

Figure 5-5 Yibo's Things tab: two views

The media items view (Figure 5-5a) has three resizable sections that define three different collections of artefacts, as explained below. Similar to the 'My Buddies' section of the people tab, each section contains two panes that allow an end user to adjust an artefact's representation level as an icon or a thumbnail, and move artefacts between them via drag and drop. For artefacts, the icon view displays a static image representing the type of artefact (e.g., that the artefact is a web page), while the thumbnail view (usually) shows a live representation of the artefact (e.g., the actual web page). The three sections are described below.

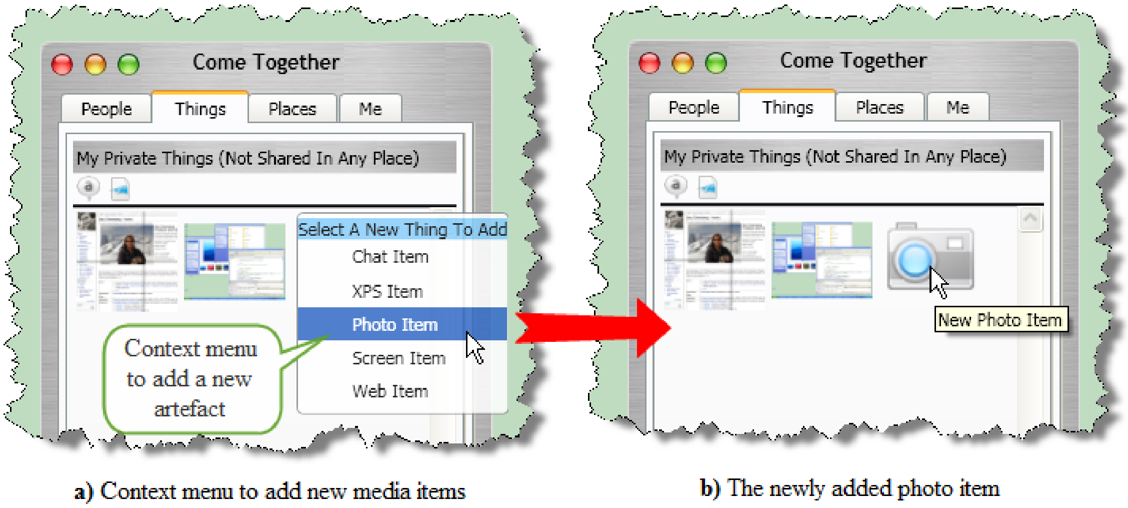
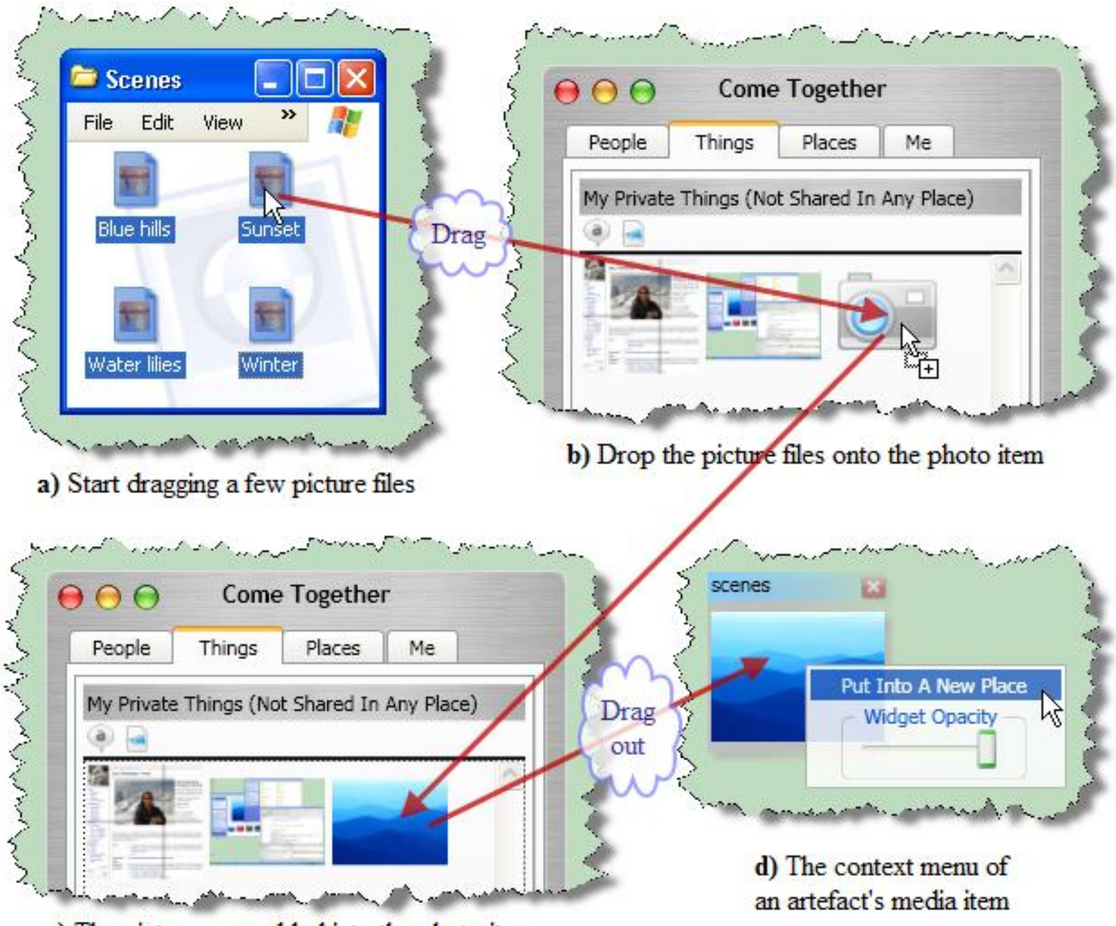

Come Together
People
Things
Places
Me
My Private Things (Not Shared In Any Place)
Select A New Thing To Add
Chat Item
XPS Item
Photo Item
Screen Item
Web Item
Context menu to add a new artefact
Come Together
People
Things
Places
Me
My Private Things (Not Shared In Any Place)
New Photo Item
a) Context menu to add new media items
b) The newly added photo item

Figure 5-6 Yibo's private artefact media items

The *My Private Things (Not Shared In Any Place)* section defines a personal space where the user can create and collect artefacts that are not currently shared with others. The user is the 'owner' of these artefacts. As illustrated previously in Figure 4-10, an end user can import a new artefact into CT simply by dragging and dropping it into this section (e.g., a .jpg file, an XPS file, a web link). CT will look at the type of artefact and – if that type is associated with a particular artefact media item – will create that media item within the section to hold it. Alternatively, the user can create and add a particular media item to this section via the context menu (Figure 5-6), and then add content to the empty media times by dragging the artefact atop of it. For example, Figure 5-7 a,b and c illustrates a sequence where the user drags an image atop of the 'Photo' media time, where that image is then displayed as a thumbnail.

As with people, an artefact media item can be dragged out of the main window onto the desktop as a standalone media item window, where its representation (icon, thumbnail or full view) depends on the window size. Alternately, the media item can be used to create a new place or dragged into an existing place (e.g., Figure 5-7d), where it is also viewable in the icon, thumbnail or full view sections of that place.



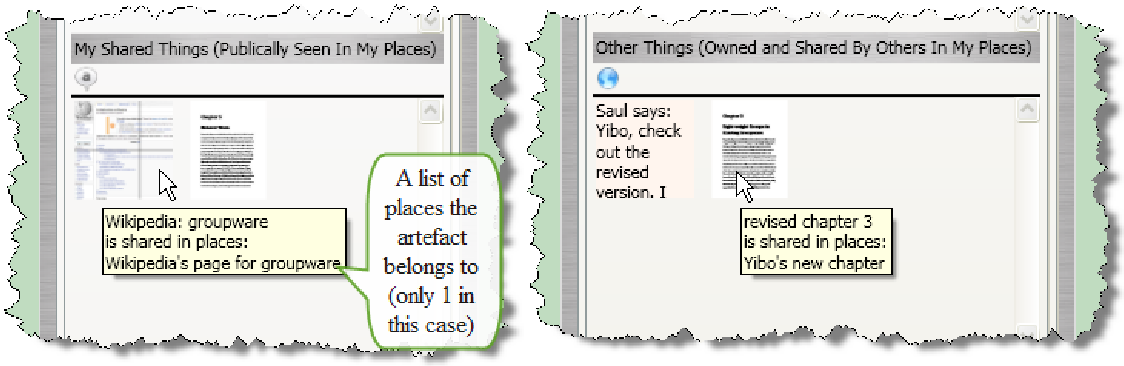
a) Start dragging a few picture files

b) Drop the picture files onto the photo item

c) The pictures are added into the photo item

d) The context menu of an artefact's media item

Figure 5-7 Adding artefacts into a empty media item


My Shared Things (Publically Seen In My Places)
Other Things (Owned and Shared By Others In My Places)

Wikipedia: groupware
is shared in places:
Wikipedia's page for groupware

A list of places the artefact belongs to (only 1 in this case)

Saul says:
Yibo, check
out the
revised
version. I

revised chapter 3
is shared in places:
Yibo's new chapter

a) Tooltip of a Yibo's shared artefact

b) Tooltip of a shared artefact by others

Figure 5-8 The other two sections: Yibo's shared artefacts and artefacts shared by others

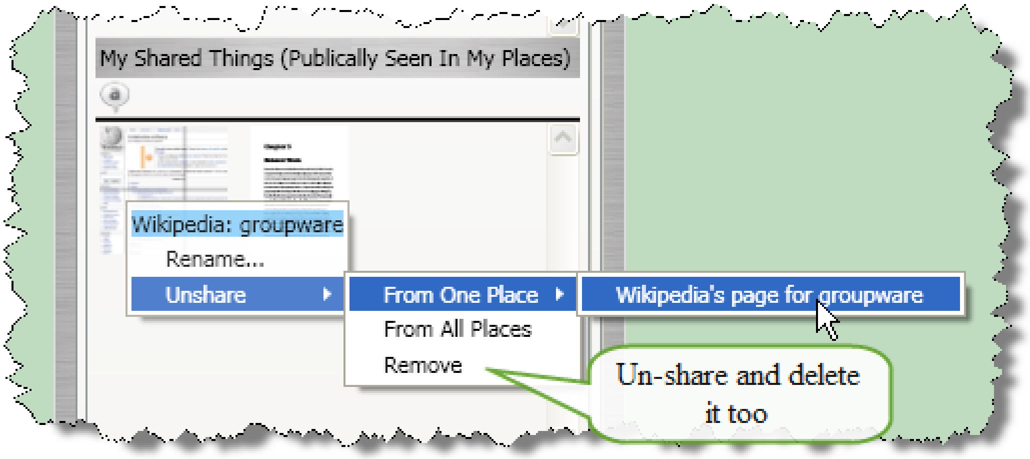

My Shared Things (Publically Seen In My Places)
Wikipedia: groupware
Rename...
Unshare
From One Place
From All Places
Remove
Wikipedia's page for groupware
Un-share and delete it too

Figure 5-9 Context menu of a media item in “My Shared Things”

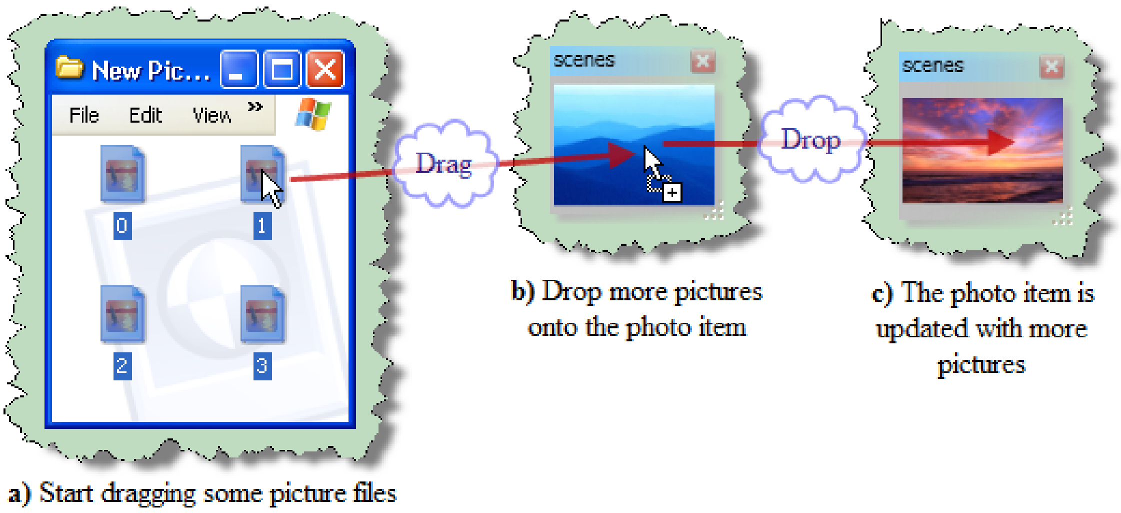
The *My Shared Things (Publically Seen In Places)* are items that the user as owner has created and then shared with others (Figure 5.5a, middle section). For example, if the owner drops an unshared media item into a place, CT will automatically move that item out of the ‘My Private Things’ section to the ‘My Shared Things’ section. When the owner mouses over an artefact in that section, a tooltip appears that describes what place(s) that item is shared in, as illustrated in Figure 5-8a. Further options are available via the context menu, where the owner can’ unshare’ the artefact from the one or more places it belongs to, or even remove it altogether (Figure 5-9).

Finally, the *Other Things (Owned By Others And Shared In My Places)* contains all artefacts in all the places the current user is in that are not owned by that user (Figure 5.5a, bottom section). The artefact media item’s tooltip behaves similarly, displaying which places it belongs to (Figure 5-8b). However, it offers no context menu in this section.

The content of the three sections above are not uniform across all users, as the placement and selection of media items depend on that particular person’s perspective. That is, the sections comprise the current user’s individual view of all their artefacts of interest.

Artefacts in the media item window. As with a people media item, an end user can drag an artefact’s media item out of the CT console. The resulting media item window (Figure 4-12) also has three size-dependent representation levels—the icon, thumbnail, and full view. Other interface details are similar: e.g., an artefact’s context menu also offers options to create a new place or set the opacity level of the window (Figure 5-7d). Similarly, an arterfact’s media item window can be added into one or more existing places by dragging and dropping it onto a chosen place window or place strip.

However, there are several differences. First, the appearance of the representation levels is determined by the particular media item. For example, the thumbnail views of a web item and photo item show a miniature of the web page and a photo respectively. Details for the stock media item types are provide in Section 5.3. Second, artefacts and people media items react differently as drag-and-drop targets. Dropping an object onto a person’s presence item implies that one wants to share the object with that person (Figure 4-7). Dropping an object onto an artefact’s media item implies that one wants to add or update the media item’s content with that artefact as long as the types match. For example, Figure 5-10 shows how dropping an image onto a photo item adds that photo to it and makes it the currently displayed photo in the thumbnail.



a) Start dragging some picture files

b) Drop more pictures onto the photo item

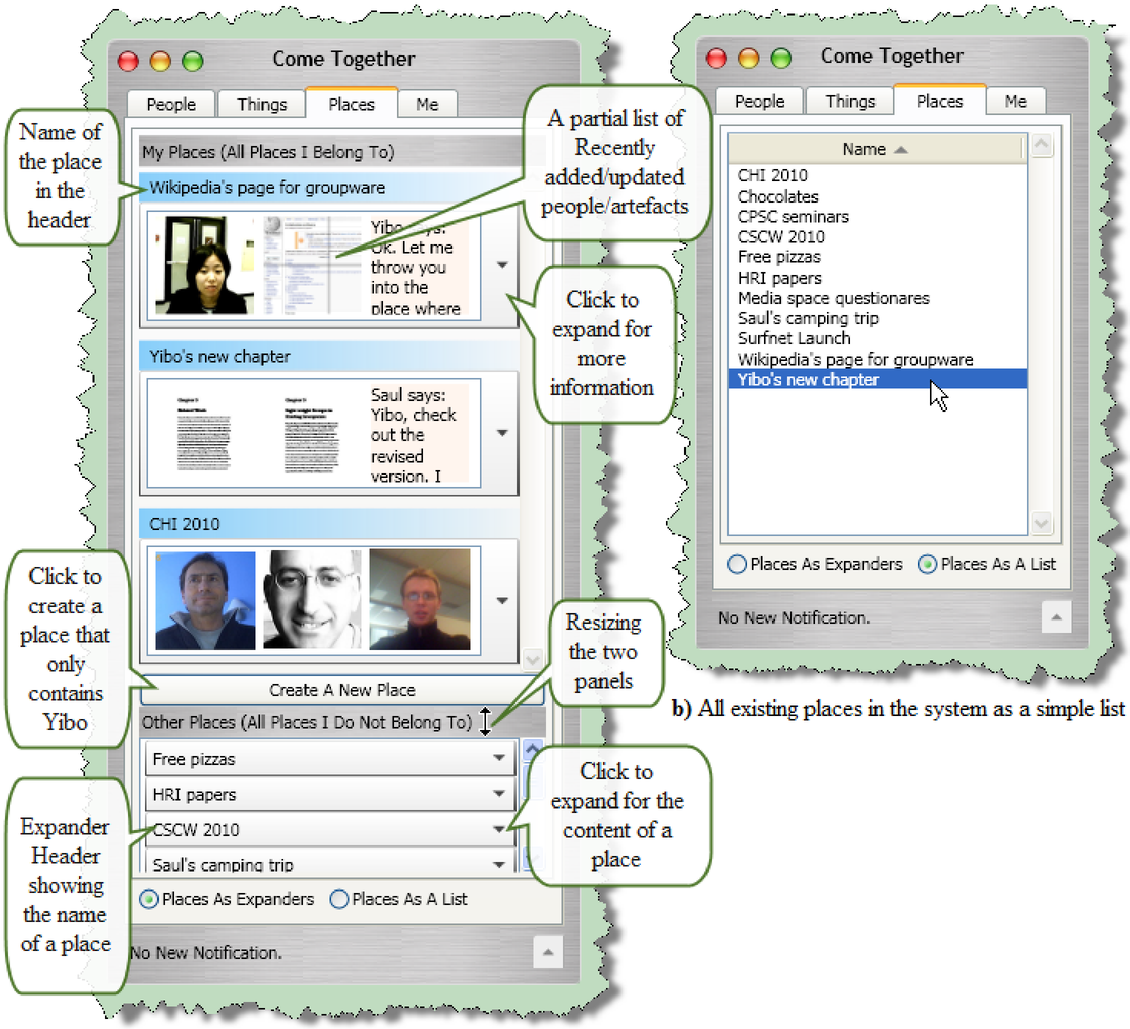
c) The photo item is updated with more pictures

Figure 5-10 Dropping artefacts onto a media item to upate its content

5.1.3 Places

Places were briefly mentioned above in the context of people and artefacts. This section describes CT places in more detail.

Generally speaking, a Come Together place represents a locale from the perspective of the Locales Framework. The place provides the site (a container) and means (a number of artefacts and/or people media items held by the container) conducive to group interactions. It also includes various interface mechanisms to make place construction, manipulation, persistence, and deconstruction simple and rapid.



Yibo says: Ok. Let me throw you into the place where

Saul says: Yibo, check out the revised version. I

b) All existing places in the system as a simple list

a) Yibo's places and all other places as expanders

Figure 5-11 Yibo's Places tab: two views

Places within the CT console. The CT console contains a Places Tab (Figure 5-11), which lists all existing, public places (publicity of places are explained shortly) known to the CT server. Similar to the People and Things tabs, the Places tab offers two views selectable via radio buttons, “Show places as expanders” and “Show places as a list”.

The default view (Figure 5-11a), which lists all places as expandable entries, has two resizable sections. “My Places” lists all places the current user is in, while “Other Places” lists all other public places, i.e., the places where the current user is not a member. The

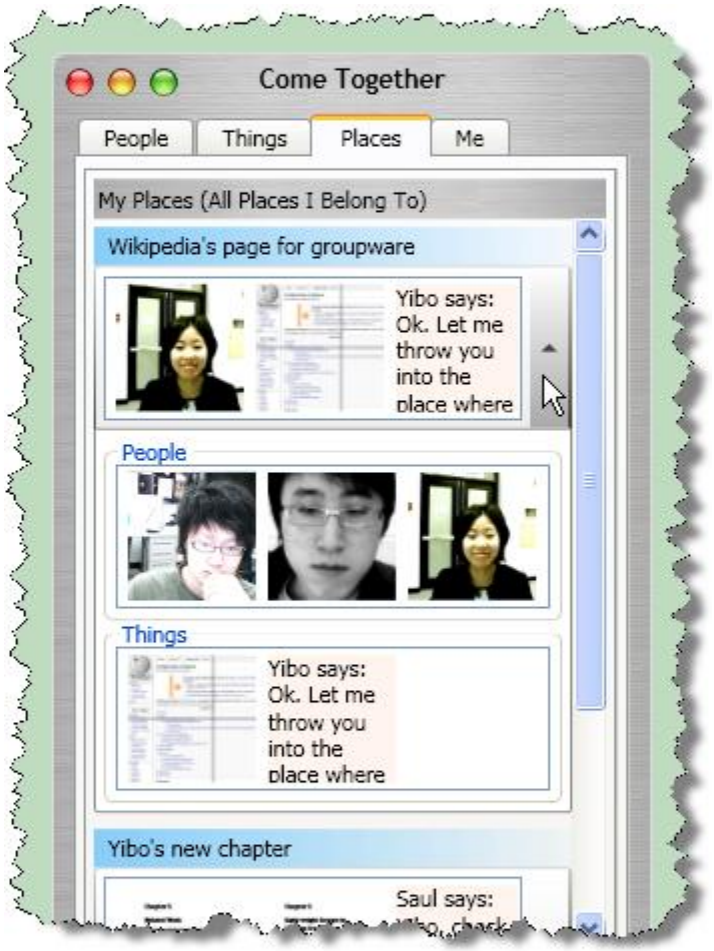
“Other Places” list lets an end user discover and peruse other places that they can optionally join into. A place can also be designated by a participant as a private place; while these places are seen by participating members in their “My Places” section, they are not displayed in the “Other Places” section. Finally, as with other tabs, the end user has the option of viewing all places in the system as a sortable list (Figure 5-11b).

We now describe the appearance of the various sections. In the upper “My places” section (Figure 5-11a, top), each listed place is displayed within a titled graphical row. The title is the name of the place. Each horizontal row contains a subset of media items in that place (i.e., as many as can fit in the row), where the most recently updated or added media items of people and artefacts are preferentially displayed. When there are more media items than can fit in that row, the end user can click the arrow button at the row’s right: the row expands to include another two rows, with one containing people, and the other the artefacts in that place (Figure 5-12).

The context menu of a place (Figure 5-13) provides various options to the user, e.g., to rename a place, set its public or private status, set the public visibility of the place (i.e., whether non-members are limited to see the icon, thumbnail or full view levels of media items in that place), or even to quit a place. For convenience, an end user can also create a new place from an existing place by clicking the “Create a new place” button (Figure 5-11a): a new, empty place with the current user as the only member will then appear.

The “Other Places” section (Figure 5-11a) lists all public places created by all other users. Those places designated as ‘Private Place’ in a place’s context menu as visible in Figure 5-13 will not appear here. Because there can be many “other places”, each with a variable public visibility setting, CT uses a somewhat simplified presentation of each place. A place can be displayed by only its title (e.g., ‘Free Pizza’ in Fig. 14a, top). A user can expand that place to reveal further details, although what is seen depends on the public visibility levels for media items of people and artefacts as mentioned previously. For example, Figure 5-14a shows two places with people and artefacts visible as thumbnails, while Figure 5.14b shows how those places would appear if a place participant had limited the view to only icons. Figure 5.14c shows the same place, but this time no media items are visible as the media items were set to ‘not visible’.

Importantly, any person can join into a public places simply by selecting the “join this place” button seen at the bottom of each expanded place (Figure 5-14). Permission to join the place is provided implicitly by others by making that place public. As a side effect, that place is moved from the user’s “Other Places” into the “My Places” section.


Come Together
People
Things
Places
Me
My Places (All Places I Belong To)
Wikipedia's page for groupware
People
Things
Yibo's new chapter

Yibo says: Ok. Let me throw you into the place where

Yibo says: Ok. Let me throw you into the place where

Saul says: Yibo, check

Figure 5-12 The expanded area of a place shows its details: all people and artefacts.

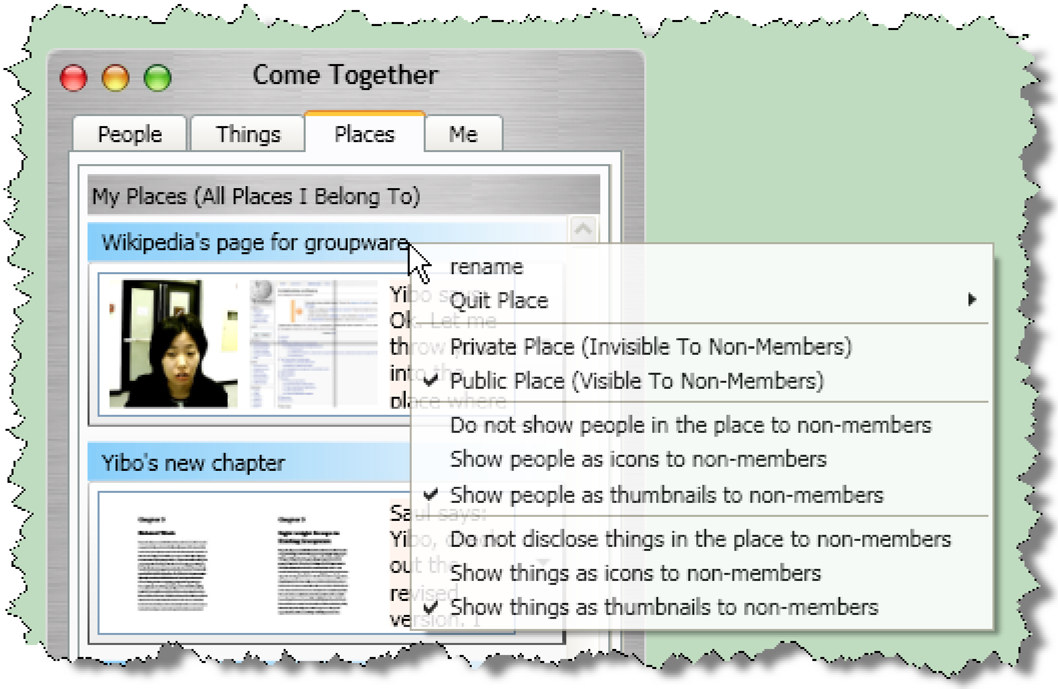
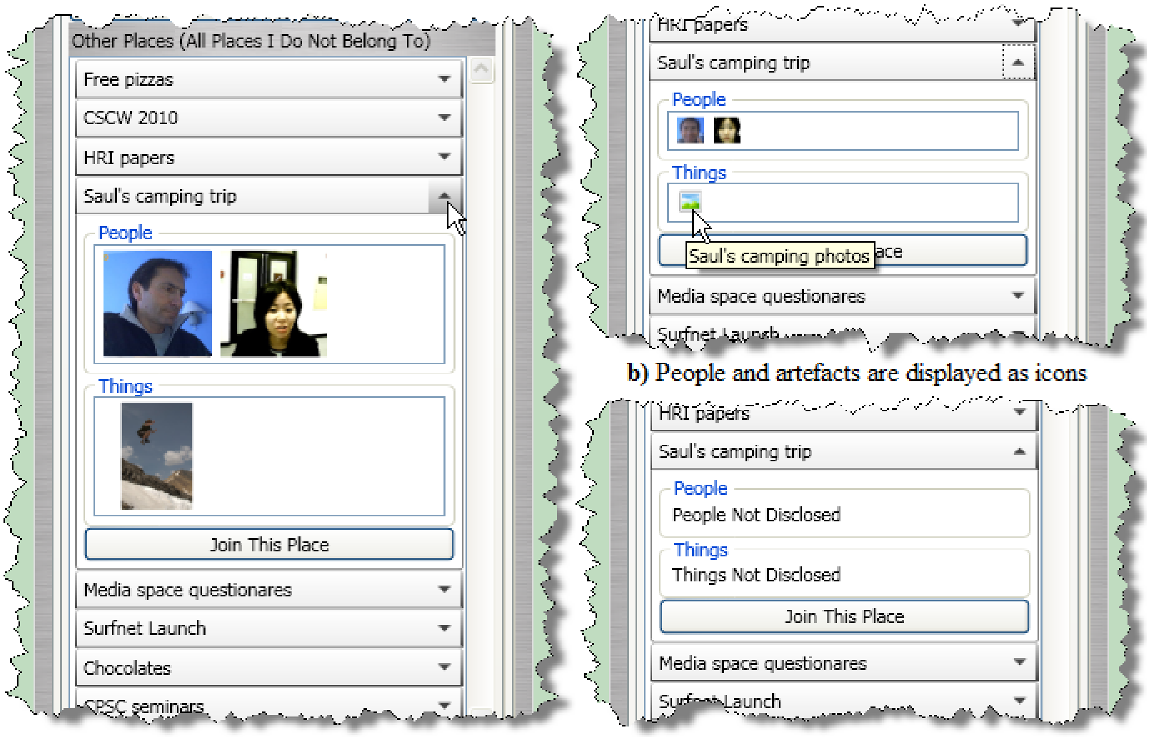

Come Together
People
Things
Places
Me
My Places (All Places I Belong To)
Wikipedia's page for groupware
Yibo says: Ok. Let me throw into the place where
Yibo's new chapter
Saul says: Yibo, check out the revised version. I
rename
Quit Place
Private Place (Invisible To Non-Members)
✓ Public Place (Visible To Non-Members)
Do not show people in the place to non-members
Show people as icons to non-members
✓ Show people as thumbnails to non-members
Do not disclose things in the place to non-members
Show things as icons to non-members
✓ Show things as thumbnails to non-members

Figure 5-13 The context menu of a place to set the publicity and visibility of a place



b) People and artefacts are displayed as icons

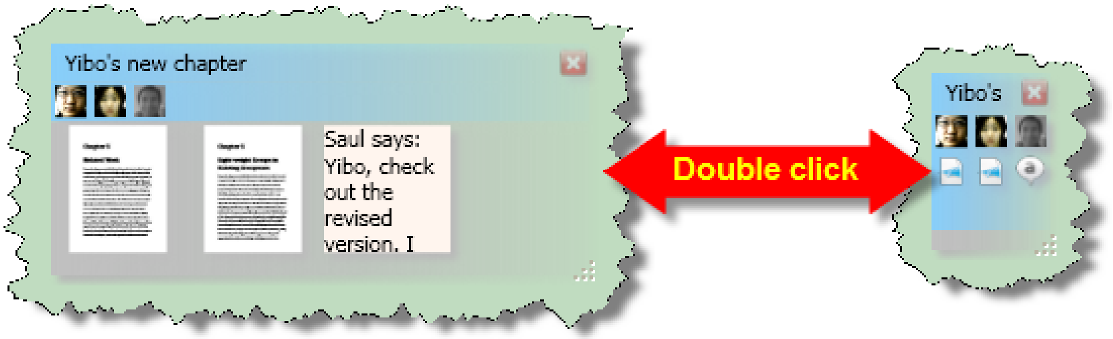
a) People and artefacts are displayed as thumbnails

c) No information is displayed

Figure 5-14 An expanded "other place"

Places as stand-alone windows. As illustrated in Figure 4-3 in Chapter 4, a person can drag a place out of the CT console, where it then appears as a titled window containing all media items of its people and artefacts. The user can customize the display of media items within three sections: the icon, thumbnail and full view. This is an individual view, as that arrangement is unique to that person.

To support center/periphery placement of a place, the places window can appear in different forms. For example, a user can rapidly shrink the normal view of a place to show its icon-only view by double-clicking (Figure 5-15). Within either view, however, the user can still rapidly preview icon-sized media item contents by hovering the mouse cursor over that icon: a semi-transparent transient tooltip containing the thumbnail-sized media item will appear (Figure 5.16).

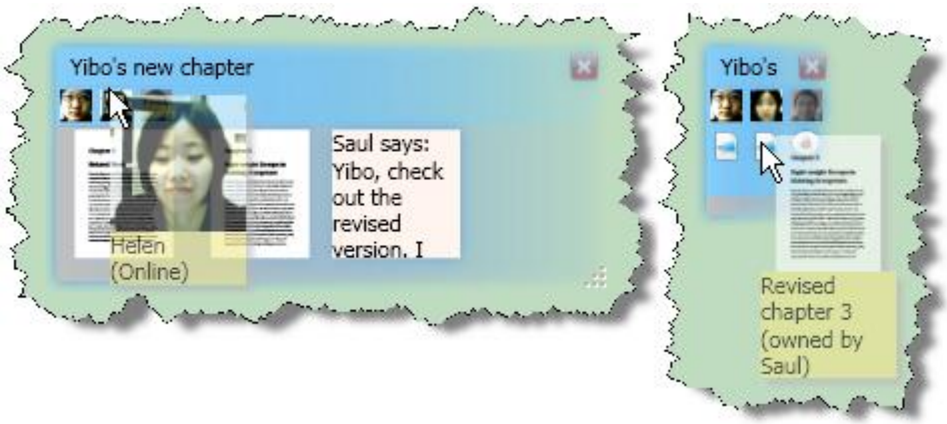


Saul says: Yibo, check out the revised version. I

a) icons/thumbnails hybrid view of a place

b) icons view of a place

Figure 5-15 Two presentation levels of a place



Saul says: Yibo, check out the revised version. I

Figure 5-16 Mouse-over preview of a icon-sized media item

5.1.4 The Place Strips

A unique interface element of Come Together is the place strips located on the desktop edges. Its purpose is to let an end user customize their individual view of their places by moving them even further to the periphery, while still allowing them to get subtle notifications of ongoing changes to a place and opportunity to explore elements of a place.

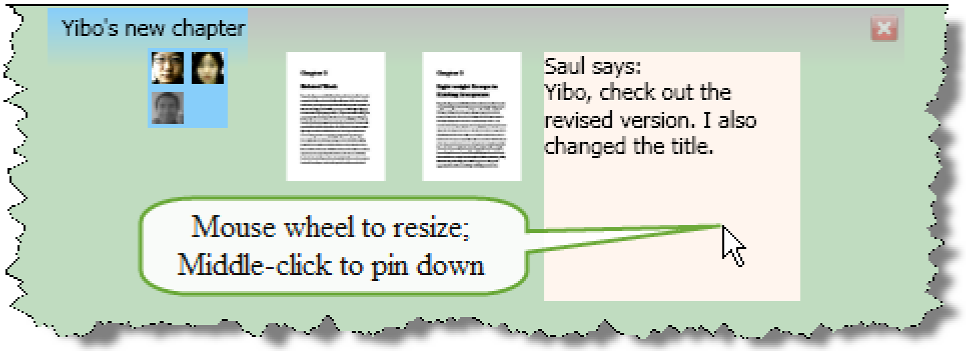
Desktop edges are controlled by CT. They are initially empty. At any time, the end-user can move a place to the edge periphery by dragging the place window to the horizontal or vertical edges of the monitor(s). The place is then transformed into a strip on the edge of the display, where its media items are docked within that strip. Figure 4-13f and Figure 5-17 both illustrate this strip.

Within this strip, the end-user has considerable control over how media items appear, i.e.., they can fine tune whether a media item is deeply on the periphery (e.g., as an icon), somewhat on the periphery (as a thumbnail), somewhat more to the center (by raising a limited full view, resizing it, and even pinning it to keep it permanently at that size).

Specifically, the place strip supports two presentation levels—the icon view and the thumbnail view—of its media items. The left/upper part of the place strip stacks media items as their icons while the right/lower part arrays media items in their thumbnail views. The end user can drag a media item between these two parts to change its representation.

In addition, the user can control these representations. First, moving the cursor over an icon-sized media item will bring up a transient thumbnail-sized preview (as in Figure 5-16). Second, and as illustrated in Chapter 4 (Figure 4-17a), moving the cursor over a thumbnail item causes the media item to enlarge to fit more content (a transient, larger thumbnail view). Third, the end user can make this already enlarged thumbnail view even larger by adjusting its size via the mouse scroll wheel (Figure 5-17). Fourth, the user can make that resized thumbnail view permanent by pinning it to the strip (a middle mouse button press). Finally, and similar to a place window, a place strip can be rapidly switched from a hybrid view containing icons, thumbnails, and customized thumbnail views (Figure 5-18a) to a very compact icon-only view (Figure 5-18b). Double-clicking toggles between these views.

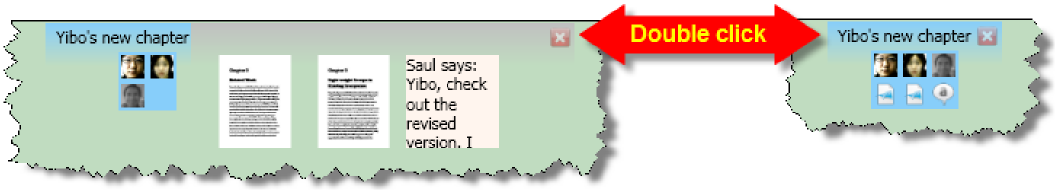
The strips can be relocated to provide a sense of civic structure as described in the locales framework, albeit in a limited way. Strips can be moved adjacent to one another to connect related places as a cluster of places. One such cluster of two places is illustrated in Figure 5-19. However, this is an individual view; only the end user that created this adjacency pair will see that. The current system implementation provides no means to connect place windows.



Saul says:
Yibo, check out the revised version. I also changed the title.

Mouse wheel to resize;
Middle-click to pin down

Figure 5-17 A horizontal place strip and its resizable media item.

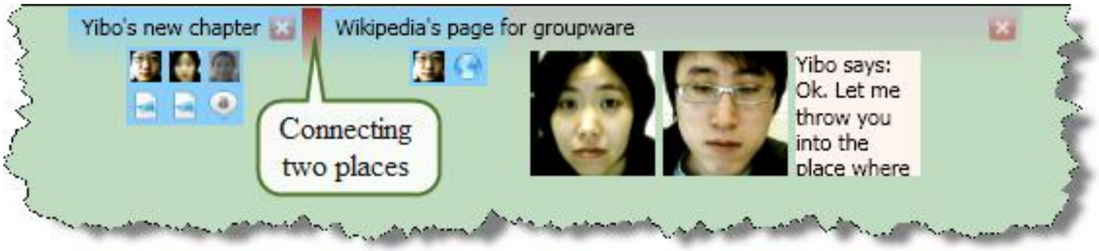


Saul says:
Yibo, check out the revised version. I

a) icon/thumbnail hybrid view of a place strip

b) icon-only view of a place strip

Figure 5-18 A place strip

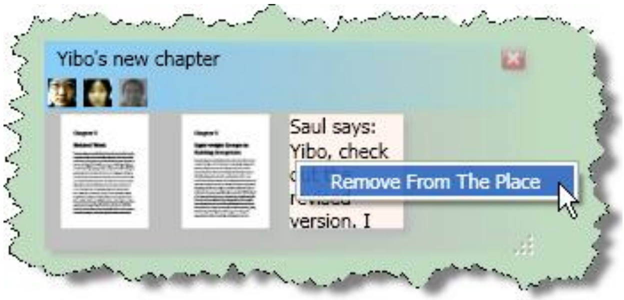


Connecting two places

Yibo says:
Ok. Let me throw you into the place where

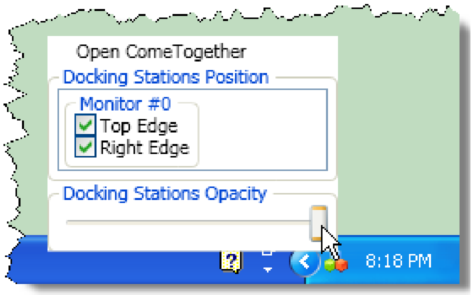
Figure 5-19 Two connected place strips

Other facilities of the strip resemble that of the places window. An end user can drag out a media item out of its place, where that item is duplicated on the display as a standalone media item window. Any existing media item of a person or an artefact can be added to a place by drag-and-drop (Figure 4-20c). File types can also be drag-and-dropped directly into a place, where a new media item matching that file type is immediately created and added to that place and to the CT console under the Things tab (Figure 4-23b). The context menu of that place also offers options to create and add new media items (Figure 4-4a). The display order of media items can be adjusted by dragging and moving in a place. A person or an artefact can be removed from a place via the context menu (Figure 5-20). While the opacity of a single place on the strip cannot be set individually, the overall opacity of all places on the strip can be set in the context menu of the system icon of Come Together (Figure 5-21). Similarly, the input focus of all places on the strip can be toggled by selecting Win+Space.



Saul says:
Yibo, check
revised
version. I

Figure 5-20 Context menu of a media item in a place



☑ Top Edge
☑ Right Edge

Figure 5-21 Context menu of Come Together's system icon

Other more routine interface facilities are available. As with the place's window, a place strip can be removed from a desktop edge by clicking the "x" button in the corner. This does not delete the place: it can still be found in "My Places" tab of the CT console, and it can be dragged out of it at any time.

5.1.5 Places as individual views on the center/periphery spectrum.

To summarize, a place has the three basic forms along the centre/periphery continuum:

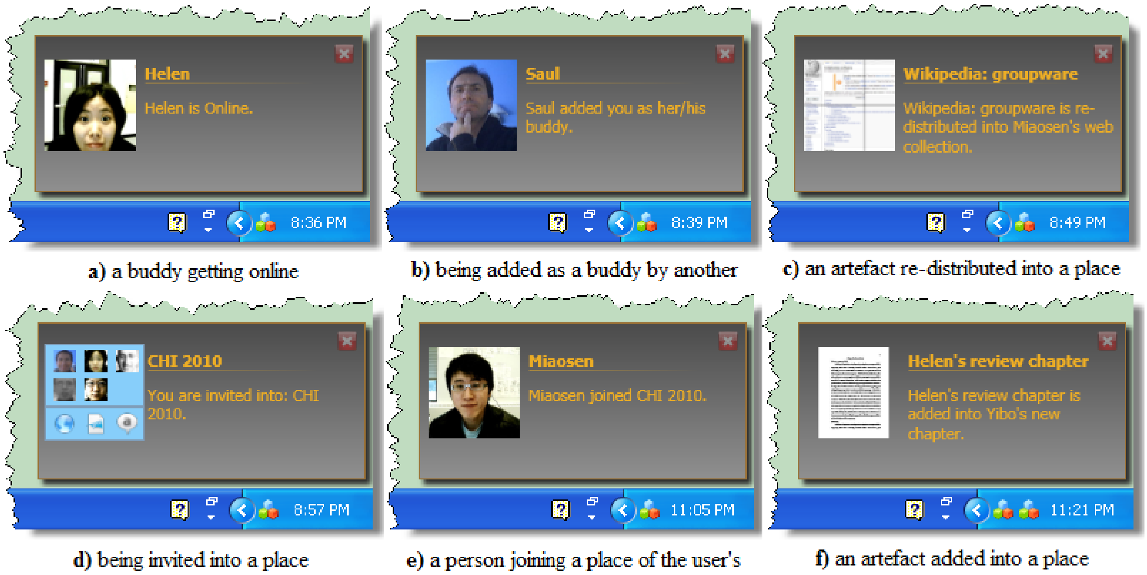
- a place window on the main part of the display (most engaged and most centered);
- a place strip on a display's edge (less engaged, more peripheral); and as
- an expander entry in the Places tab of the CT console (least engaged and most peripheral).

Each of these forms allow further customization of a particular media item's centre/periphery relationship, e.g., between its icon, thumbnail, limited full view, and full view manifestations, as well as by its opacity. The idea is that compact icons and semi-transparent views are more peripheral, while full views and opaque views are more central. As well, the input focus on semi-transparent items can be toggled: as these items are on the periphery, they do not recognize input (and thus do not interfere with a user's interaction on other windows above it) unless explicit action is taken by toggling input state.

The above center-periphery spectrum is part of an individual's view set of their places, where they can customize their view to reflect their particular interests and engagements across all places and media items within a place. In practice, we expect people to use the simpler.

5.1.6 System messages

Places and media items are not static. Other people can create new places, enter or leave an existing place, change their online status, add media items into a place, and change the contents of a particular media item. For a dynamic system like this, we consider it important to keep people aware of these activities while still maintaining balance.



Helen

Helen is Online.

a) a buddy getting online

Saul

Saul added you as her/his buddy.

b) being added as a buddy by another

Wikipedia: groupware

Wikipedia: groupware is re-distributed into Miaosen's web collection.

c) an artefact re-distributed into a place

CHI 2010

You are invited into: CHI 2010.

d) being invited into a place

Miaosen

Miaosen joined CHI 2010.

e) a person joining a place of the user's

Helen's review chapter

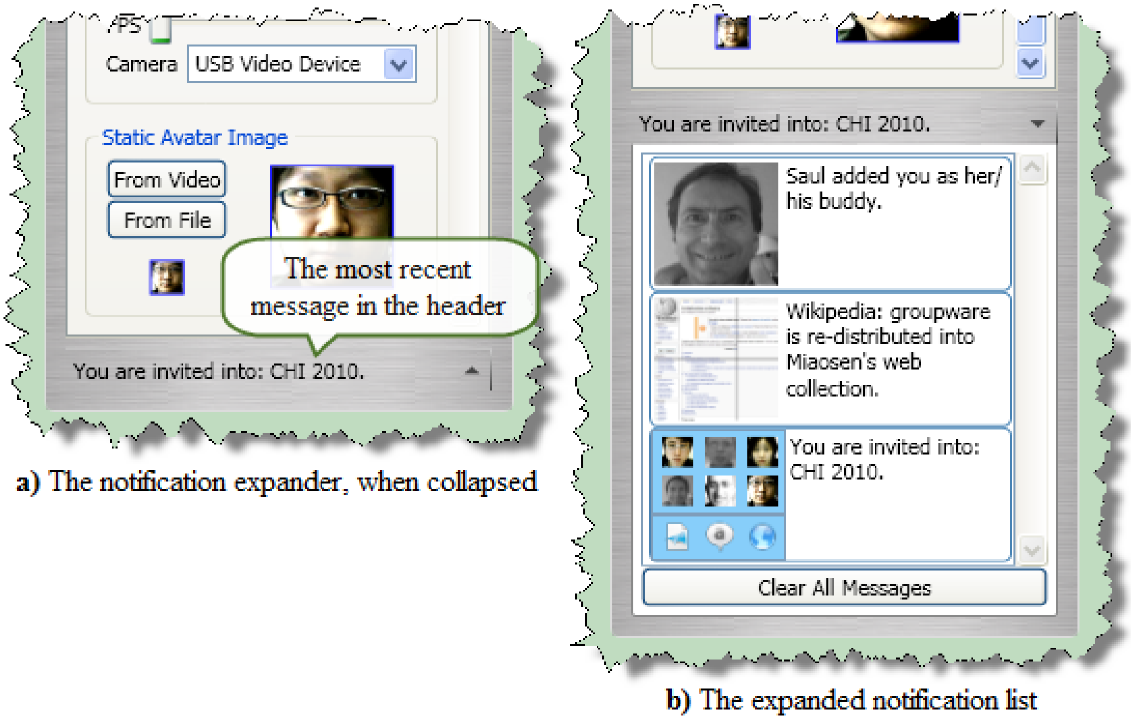
Helen's review chapter is added into Yibo's new chapter.

f) an artefact added into a place

Figure 5-22 notification pop-up toasts

Come Together tracks these events, and notifies each user of events that may concern them. These include things like: a buddy appears online; the person is added as a buddy by another user; an artefact within a media item is updated; an artefact owned by the current user is re-distributed into another place by another user; the current user is invited into another place; another person joins a place the current user is in; another user adds an artefact into a place inhabited by the current user, and so on. Several means are used to notify the current user of these and other events.

First, CT outlines the newly added (Figure 4-5a) or updated (Figure 4-16b) media items in a place's window or strip in red. This is used as a visual indication that something has changed since the person last interacted with that media item. Second, some events will appear as transient graphical and textual popup messages via the notification toasts on the windows status bar (Figure 5-22). Third, important messages are logged into a persistent expandable view at the bottom of the CT console so they can be viewed at any time. These are messages that appear in a pop-up toast but which may have been otherwise missed (e.g., if a user was not attending the display during a notification pop-up or because the user was offline at that time). To avoid excessive messages in this list, only a subset of message types are persistent: the current user is added as a buddy of another user; an artefact owned by the user is redistributed into another place by another user; the user is invited into a place.



You are invited into: CHI 2010.

a) The notification expander, when collapsed

You are invited into: CHI 2010.

Saul added you as her/his buddy.

Wikipedia: groupware is re-distributed into Miaosen's web collection.

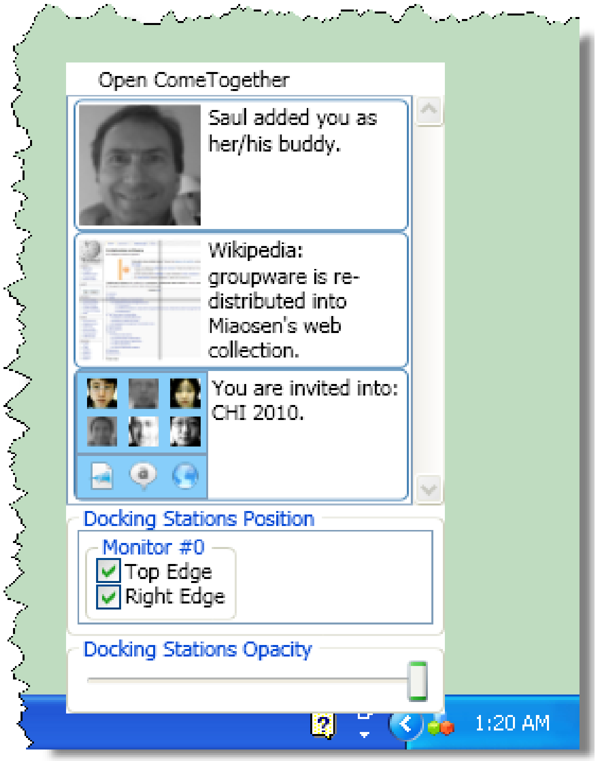
You are invited into: CHI 2010.

b) The expanded notification list

Figure 5-23 The notification expander at the bottom of the main window

Messages are displayed in recency order. The most recent message is always visible in the message header (Figure 5-23a), while older messages are visible if the message area is expanded (Figure 5-23b). The same message list is also available in the context menu of the Come Together system icon in the system's notification tray (Figure 5-24).

Finally, messages have rich content regardless of whether they appear a pop-up message toast or an entry in the message list. They typically include a textual descriptor and a thumbnail-sized image the message is about, which can be either a thumbnail-sized media item (Figure 5-22a) or the icons view of a place (Figure 5-22d). Finally, the user can click on the message for an immediate media item window for the person/artefact/place the message is about (Figure 4-14a, 4-14b).



Open ComeTogether

Saul added you as her/his buddy.

Wikipedia: groupware is re-distributed into Miaosen's web collection.

You are invited into: CHI 2010.

Docking Stations Position

Monitor #0

☑ Top Edge

☑ Right Edge

Docking Stations Opacity

Figure 5-24 The notification list in the system icon context menu

5.2 Stock media item types

Media items are generalized components, where they are used to represent people and artefacts in the world of Come Together. Like the Community Bar, media items are programmable plug-ins that a programmer can use to create items for various purposes. Thus CT is an open system. While a stock set of media items are included (described below), these are illustrative of potential media items rather than restrictive. Chapter 6 will delve into the technical aspects of developing a custom media item. In this section, I will briefly review the capabilities of the stock media items distributed with the existing version of CT. These include: a presence item representing people, a communication-oriented item including a text chat item, and several other artefact-oriented items including a web item, a photo item, a screen item, and a document (XPS) item.

5.2.1 Presence item

The presence item provides awareness of particular users of Come Together. The icon view shows a user’s static avatar (Figure 5-3a) set in her account (Figure 5-1). Depending on the binary option “show my video to others” in an account (Figure 5-1), the thumbnail view (Figure 5-3b) shows either the user’s static image, or her slow video (1 frame per second). The full view (Figure 5-3c) is similar to the thumbnail view, but shows either the user’s high speed video (in the maximal frame rate specified in her account), or her static image if her video is not public. If the user is offline, all three views show her static avatar image in grayscale.

The presence item is currently the only stock media type for people. However, it would be possible to create and substitute custom types of presence items for the stock one. For example, it would be possible to create a presence item that scraped information from a person’s facebook account and use that to form the information within it.

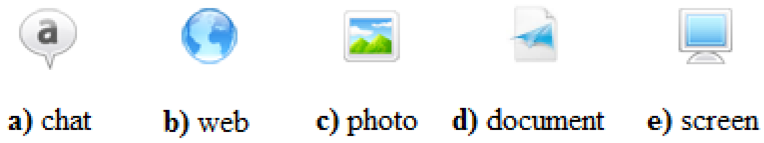


Figure 5-25 representation icons for artefact media items

5.2.2 Chat item

Textual Chat is a common means to communicate in many groupware systems. Come Together offers a simple chat item. Its icon view is the static icon of Figure 5-25a. The thumbnail view shows the most recent message in the conversation (Figure 4-16b). The full view provides an interactive panel where a user can type in a textual message and see the scrollable chat history (Figure 4-16a). As groupware, messages composed in the text chat are immediately viewable by all users.

5.2.3 Web item

The web item hosts a web page so that it can be previewed, viewed or interacted with directly as a media item. Its icon view is the static icon of Figure 5-25b. The thumbnail view (Figure 4-12a) comprises the top part of the web page, scaled to fit the available size. The full view (Figure 4-12c) is a full, interactive browser hosting the web page. As with a normal browser, the user can navigate links on that page to display other pages in place. As groupware, all users in that place will see that updated page in their thumbnail or full view.

5.2.4 Photo Item

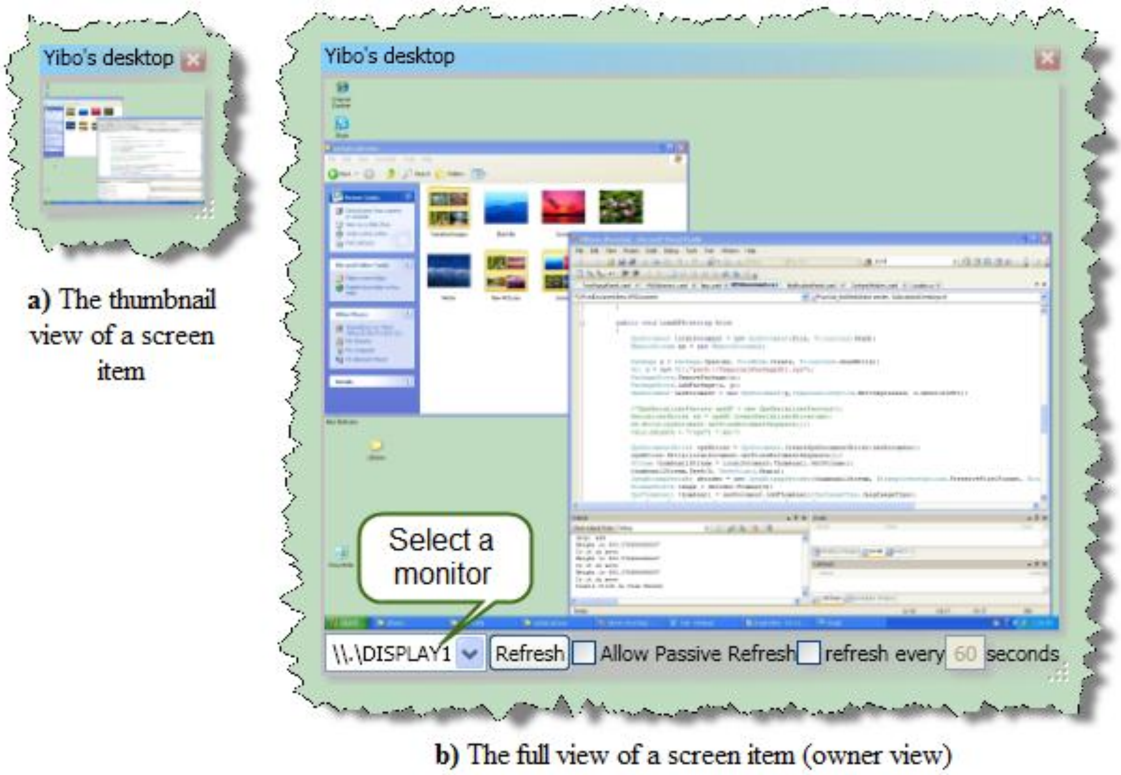
A photo item is a container for a shared set of photos. Users sharing the item can flip through all the photos in the same set, either manually or as a slide show. Its icon view is the static icon of Figure 5-25c. The thumbnail view shows the thumbnail of the currently selected photo in the set (Figure 4-2). The full view shows a selected photo in its full size (Figure 4-2), as well as various interactive controls as a panel on its bottom. Through this panel, the user can manually flip through the set, or click the “play” button to start a slideshow with a specified “display duration”. A user can add additional photos to the set simply by dropping image files onto a photo item (Figure 5-10). As groupware, all users have immediate access to photos added by themselves and others. Users also have the option to go through their slides individually, or to link their views so that all can see the slides simultaneously as they are being played.

5.2.5 Document (XPS) item

The document item allows people to share an XPS file, a specific type of document file somewhat akin to a PDF file (e.g., a Microsoft Word file can be saved as a WPF file). This particular media item type only supports viewing an XPS file *vs.* authoring of the file. Its icon view is the static icon of Figure 5-25d. The thumbnail view (Figure 4-7d) shows the embedded previous image of the XPS file. The full view shows the entire content of the XPS file with controls to change display layout (Figure 4-8). As an aside, the choice of an XPS viewer had more to do with the ease of programming within the current capabilities of Windows WPF. While programming would be more complex, other file types and even a proper groupware annotations and/or authoring system could be created within a media item.

5.2.6 Screen item

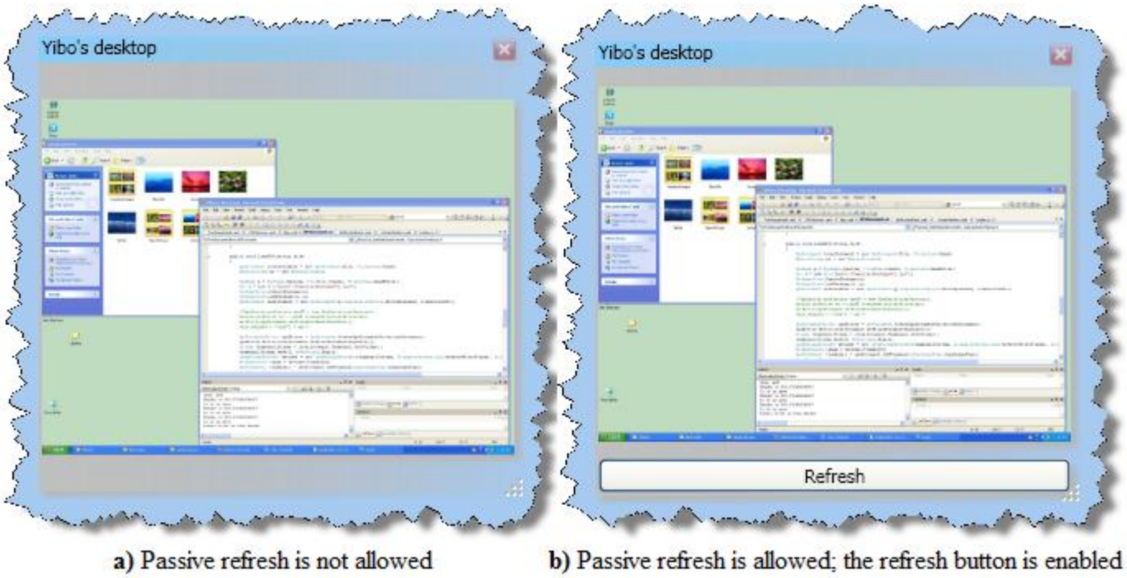
Screen sharing is a long-standing and useful feature of many groupware systems. The screen item inspired by Tee (2007) allows a user to share his screen activities so that others can see it. However, the current CT version does not allow others to control that screen remotely. The screen item captures its owner's screen as a bitmap, and shares the captured image with others. The owner can set the screen item to manually update the captured image (so that only particular moments in time are captured); or can configure the screen item to capture and automatically update the image at a particular frame rate. Its icon view is the static icon of Figure 5-25e, while the thumbnail view shows the thumbnail of the most recently captured image (Figure 5-26a). The full view shows the screen capture in full size, but with two versions of the control panel at the bottom. The owner's version, seen in Figure 5-26b, lets the owner manually capture an image of a selected monitor via the refresh button, or lets the owner specify the automatic update via a time interval. The owner can also specify if other viewers are allowed to request an update of the image if, for example, the manually updated image has gone stale and the viewer wishes to see the latest version (via the "Allow passive refresh" option). The viewers' version, seen in Figure 5-27, also has a "refresh" button (if passive refresh was allowed, Figure 5-27b) that requests that the owner's screen item manually update the image.



a) The thumbnail view of a screen item

b) The full view of a screen item (owner view)

Figure 5-26 A screen item



a) Passive refresh is not allowed

b) Passive refresh is allowed; the refresh button is enabled

Figure 5-27 the viewer's version of the screen item

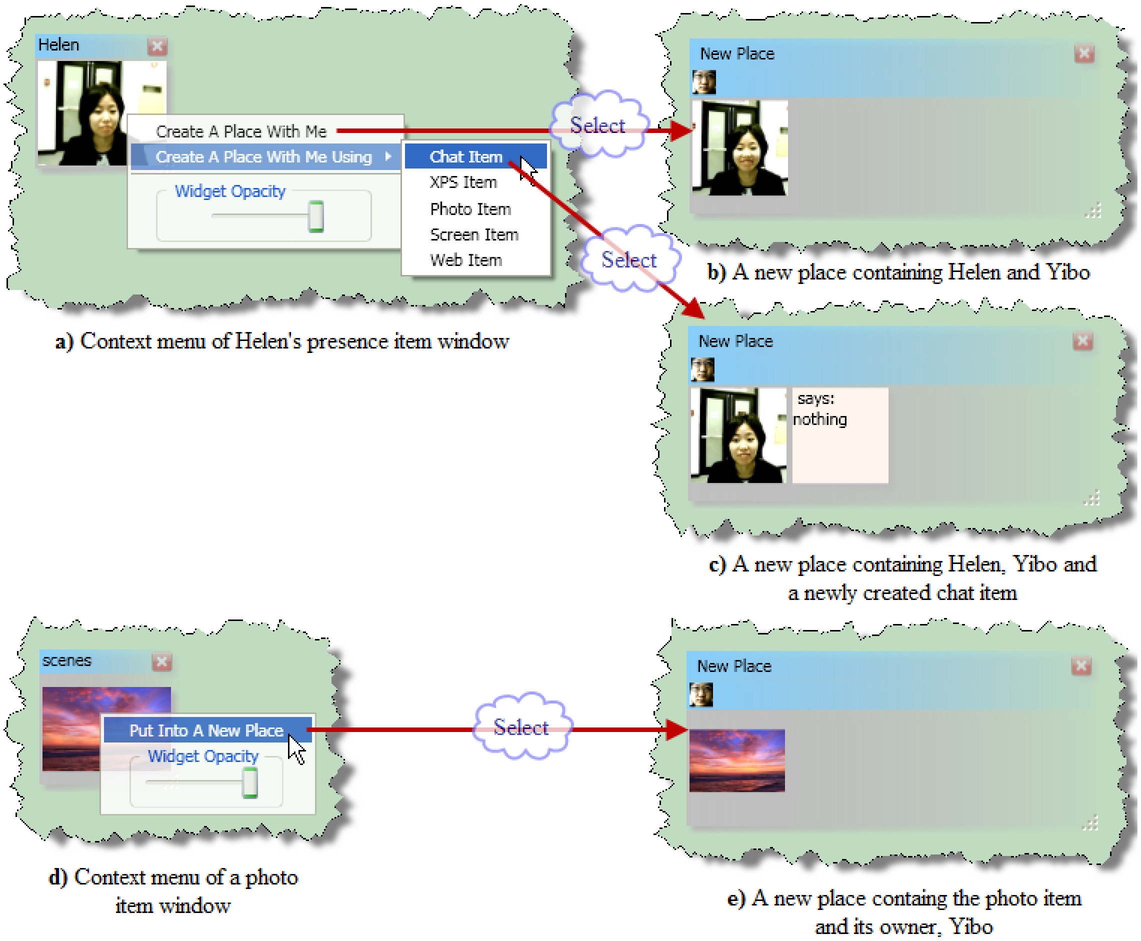
5.3 Revisiting CT as a Design Rationale

Come Together's user interface as described here and in the previous chapter are designed around three main design considerations that emerged from chapters 2 and 3: supporting light-weight locales through places, reflecting the centre-periphery continuum on multiple dimensions, and treating people and artefacts equally.

5.3.1 Light weight locales

Chapter 2 provides a number of design considerations to support light-weight work practices of groups. Come Together's place is designed around these considerations to emulate a light-weight locale.

Importantly, places are very lightweight to create and modify. An unnamed empty place can be created with a click of a button (the "Create a new place" button in Figure 5-11a). Within that, the user can easily rename it. By dragging and dropping in people, artefacts, and files, the user can easily invite others into the place, incorporate sharable artefacts, and add communication facilities. This becomes the site and means of the locale. This method affords the traditional way of creating a locale in other groupware systems, i.e., where the groupware entity has to be created first, and then other entities have to be added to it. We already saw that this was a limitation of the Community Bar, as it means that one has to create a groupware setting a priori vs. having it emerge from individual existing artefacts. In contrast, Come Together also allows people and artefacts to exist outside of a place (e.g., as stand-alone media item windows), where places can be created and people and artefacts added (or removed) within them opportunistically. The simple act of dragging two unconnected media items near each other (of people and/or artefacts), or of dropping an artefact atop a person (scenario 2, Figure 4-7), or of creating a place from within a media item window via its context menu (Figure 5-28) implicitly creates a place (see also Figure 4-1e, 4-13a). Places can evolve. People and artefacts can be added or removed as a place evolves. They can be named and re-named at any time. People can come and go (e.g., as members, or as off-line / on-line participants).



a) Context menu of Helen's presence item window

b) A new place containing Helen and Yibo

c) A new place containing Helen, Yibo and a newly created chat item

d) Context menu of a photo item window

e) A new place containg the photo item and its owner, Yibo

Figure 5-28 Using a media item's context menu to create new places

Come Together also makes places lightweight by making places and their contents easy to view and join by non-members, unless explicitly restricted by the current participants of a place. That is, CT places do not have hard walls that restrict what others can see. Non members - the outer-most periphery – can see the thumbnail media times of a place in the “Other Places” section of the Places tab, although this depends on its visibility setting (Figure 5-13 and 5-14). Users can join a public place by one click (Figure 5-14) without an invitation or without moderation (moderation can be added as an optional feature in future versions). Of course, places can be made private so that the public cannot see or join it, by un-checking the option “public place” in the context menu of a place (Figure 5-13). The public visibility, presentation levels for people and artefacts shown to non-participants, of a public place is also set in the same context menu (Figure 5-13).

The primary idea is that technical protocols or roles are not imposed in a place. Instead, people use their own social processes to mediate how they view and join a place. Every participant in a place enjoys the same privileges. Every member can invite people into a place, add artefacts to a place, modify the place, and reconfigure its settings. While this model may not fit large groups or ill-behaved groups, it is appropriate for casual interactions and intimate collaborators, which is the target of CT. Of course, Come Together can be extended to support more formal protocols, e.g., roles of participants (moderators, chairs, etc.), access control (of places and media items), security (login / password) if desired. Our stance, however, is that these should be optional rather than imposed by the system design if place creation is to be lightweight.

5.3.2 Centre/Periphery and individual views

Another key design concern of Come Together is support for the centre/periphery principle, and for individual views. Come Together inherits and extends the Community Bar approach to how this is done via individual views of locals and media times through different sizes and presentation levels.

We saw how a user can adjust and choose between the three presentation levels of a person or artefact to reflect their level of engagement. The premise is that the full view is the most centered, and the icon view is the most peripheral. Places have the same features, and more. When seen as a window or as a strip, it can be sized down to its most peripheral icon-only view, or sized quite a bit larger (including how media items within it are displayed) for a more central view. In addition, the centre/periphery level of a media item or a place can be further adjusted by its location. Moving media items or places from the CT console to a window or to an edge allows one to tune the place as a whole on this center/periphery spectrum. In addition, end users have the ability to adjust the opacity and input focus of the media item or the place window/strip.

All the above is user-specific, where they form an individual view set of everything in the world of Come Together.

5.3.3 Integration of People and Artefacts in and out of places

In Come Together, people and artefacts are treated as equal entities that can exist in and out of places. This is important, and was previously hinted at above. Media item of either a person or arterfact can be created outside of a place. For people, this means that they can behave as a buddy in IM (i.e., no explicit sharing is done, but that person has a preferred status). This also means that artefacts can be used for individual purposes. Indeed, there is no requirement in CT for the end user to actually use its groupware aspects; they may instead use media items and places solely for individual work.

People and artefacts can be brought into one or more places, which automatically makes them group-aware and shareable. They can also be moved out of places, where they can be used again for individual purposes. This behavior is quite different from Community Bar, which limited media items to appear only within a pre-constructed place.

5.4 Summary

Chapters 2 and 3 suggested a number of groupware design considerations based on the Locales Framework and from a review of existing systems. Come Together provides one possible design solution based on those considerations. This chapter along with the scenarios of Chapter 4 introduces the interface to Come Together, as well as a rationale of how it supports the various design considerations.

The next chapter goes under the covers of CT. It briefly covers CT’s technical aspects, including how media items are developed and how its client/server architecture.

.

Chapter 6. **System Implementation**

This chapter describes the technical aspects of Come Together. The first section gives an overview of Come Together's system architecture. Following that, we describe how it interoperates with the GT/SD library (de Alwis, Gutwin, & Greenberg, 2009; Boyle & Greenberg, 2005), a third party research system developed in our laboratories that Come Together uses for its backend data store, data distribution and networking. Subsequent sections explain Come Together's major architectural software components: its server, client, and media item plug-ins. We close with a technical critique, where we identify issues – and possible solutions – that remain to be addressed if Come Together is to be deployed in a realistic setting.

6.1 Architectural Overview

This section provides an overview of the Come Together Architecture and the interplay between its components. Figure 6-1 provides a bird's eye view, and will ground the discussion. Later sections will provide details.

General. Come Together is a distributed client/server architecture. It was developed using the C# programming language on the Microsoft .Net platform, with its interface created in WPF. This technology lends itself to the Model-View-Presenter (MVP) pattern (Eisenberg & Bennage, 2008, p. 177). As a distributed system, Come Together uses the GT/SD toolkit (de Alwis et al., 2009) to handle its networking and data distribution needs via a distributed Shared Dictionary, which we use to implement the *model* in the MVP pattern. Similar to its predecessor—the Community Bar (McEwan, 2006)—Come Together allows third party developers to extend its functionalities by writing custom media items through a plug-in system.

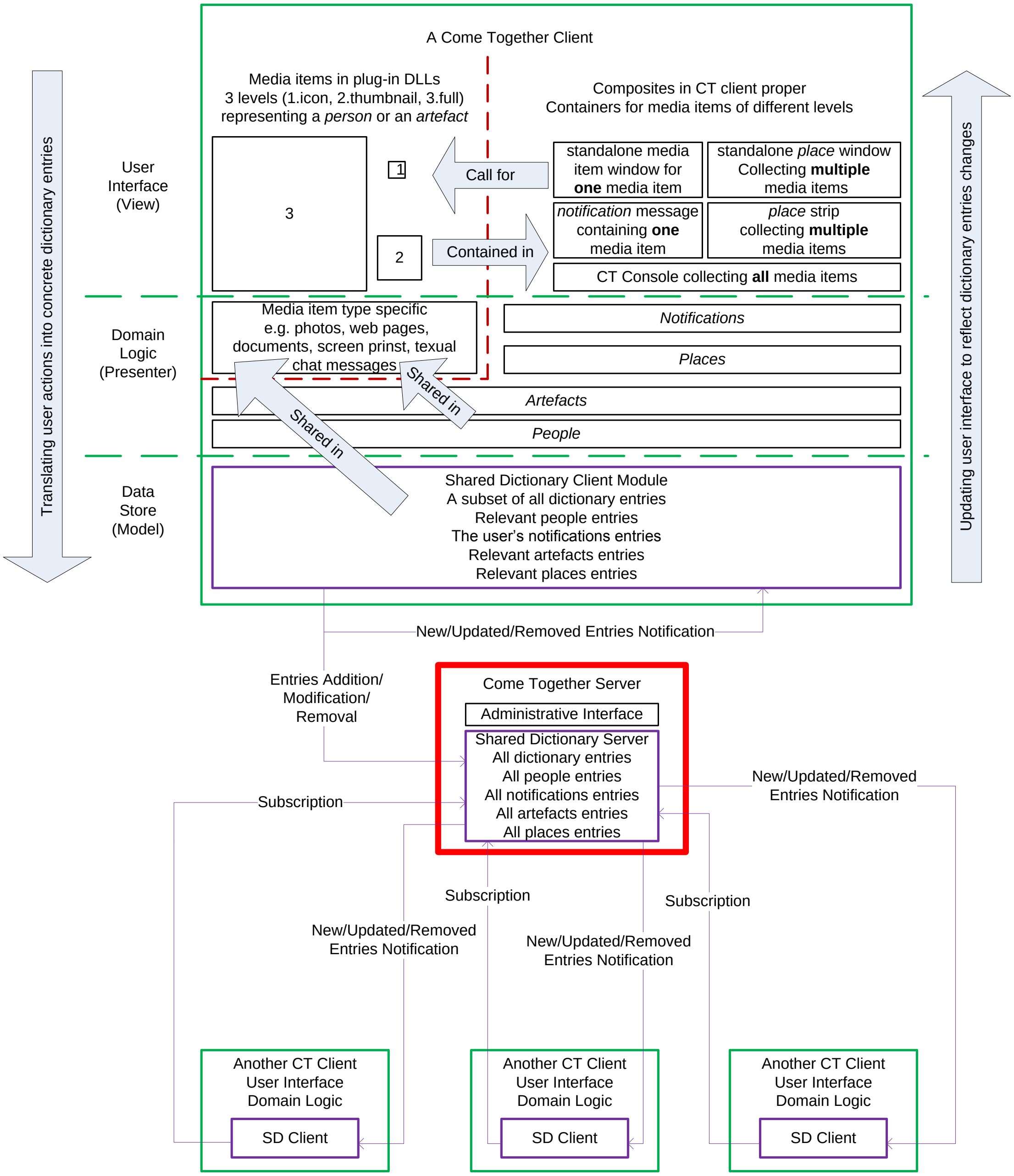


Figure 6-1 Come Together software architecture overview

Client Server Architecture. Come Together is a client-server distributed system. Figure 6-1 shows this. The server is at its center (outlined in a thick red border), and four clients are shown connected to it (outlined in a thinner green border) . The client at the top is shown in greater details than the others.

Shared Data Model and Networking. As a distributed system, Come Together relies on third party software for all its networking and data sharing and distribution needs. Specifically, it uses the Groupware Toolkit / Shared Dictionary software package (GT/SD for short) (de Alwis et al., 2009; Boyle & Greenberg, 2005). The Groupware Toolkit (GT) layer provides networking facilities. The Shared Dictionary (SD) layer, built atop the GT layer, provides a data store, a means to selectively share and distribute data between server and clients, and a notification mechanism via a publish/subscribe method[1]. Collectively, both layers of GT/SD ease the difficulty of rapidly prototyping data-intensive groupware applications via a reasonably powerful API running atop a runtime architecture.

To summarize how this all works, Come Together maintains, via the Shared Dictionary, a data model that defines an abstraction of all of its places, people and artefacts. The CT server hosts the Shared Dictionary server, which contains the entire data model (Figure 6-1, center). Each CT client hosts a Shared Dictionary client (e.g., shown in detail in Figure 6-1, top), where each client can subscribe to particular fields held by the server. Thus each client will subscribe to only the subset of the data held by the server that is relevant to that particular client. Clients can then add, change and remove data in the shared dictionary. The typical sequence of events is:

1. The Come Together client software adds/changes/removes a particular data element in the Shared Dictionary via the SD API.
2. That change is automatically transmitted to the server.
3. The server automatically updates its model to reflect that change.
4. For those clients that have subscribed to that data element, the server automatically generates a notification of that client’s and propagates that notification to the shared dictionaries held by those clients
5. Those clients’ shared dictionaries automatically store that changed data in its local cache.

[1] These SD’s event notification mechanism are not the same as the notification messages that the user sees that were presented in the previous chapter.

6. The client software then receives, from the SD client, a notification of that change, which invokes a Come Together event handler (if one is specified) to take action on it. Data associated with that change can be retrieved either from the event handler or directly from the local cache held by the client shared dictionary.

Figure 6-1 illustrates the above sequence. The thin arrows between the shared dictionary components of the various clients and the server illustrate how the clients propagate data changes to the server, and how the server propagates those changes back to the various clients.

Of importance is that the programmer need only specify the first and last step; all other aspects are handled automatically by the GT/SD software. Because of the importance of the Shared Dictionary and the way its various components interoperate, its use will be discussed in detail in its own section shortly.

Model/View/Presenter. Come Together clients follow the Model-View-Presenter (MVP) pattern, a variation of the well-known Model-View-Controller (MVC) (Greenberg & Roseman, 1999). The roles of the model, the view, and the presenter are illustrated in Figure 6-1 (see annotations on left side).

The primary difference is that the MVP pattern's view layer does not directly communicate with the data model layer and in MVC, i.e., it goes through the presenter layer. Specifically, the *model* layer (data store and networking) defines the abstract model whose data contents and structure defines people, artefacts and places. It is solely managed by the shared dictionary, as introduced above. The *view* layer comprises the user interface views—the main system window, the three presentation levels of media items, the place windows, the places strip, and the pop-up notification message toasts. The *presenter* layer is the domain logic and software that sits between the model and the view. It translate user actions from the interfaces (the views) down to the data model (i.e., by adding, changing or deleting shared dictionary data elements), and it reflects the changes in the data model up to the user interface views (i.e., by responding to SD notifications) (Figure 6-1, 6-2).

Media Items as DLL Plugins. As mentioned earlier, Come Together supports a broad variety of media items, each encapsulated as a DLL plug-in that can be added to Come Together on the fly. Specifically, each plug-in is a stand-alone piece of software that can be developed independently from Come Together, and compiled as a DLL. The main Come Together client program executable looks for these DLL plugins in a pre-defined location, and then hosts that plugin. Under the covers, the plugin must obey a defined interface, and its software is also expected to follow conventions that implement the MVP pattern. That is, it must populate the shared dictionary model held by the client (and thus propagated to the server), and must construct the view to fit and match icons, thumbnails, and full views (and any size in between).

Places windows, Places strip, Console, Notification messages, Media item windows. Come Together clients host various containers that generate particular views of the data model, and that contain various media items. Containers are thus interface components that communicate to the presenter (i.e., to update their contents or to indicate changes), and that provides space for the media item interfaces.

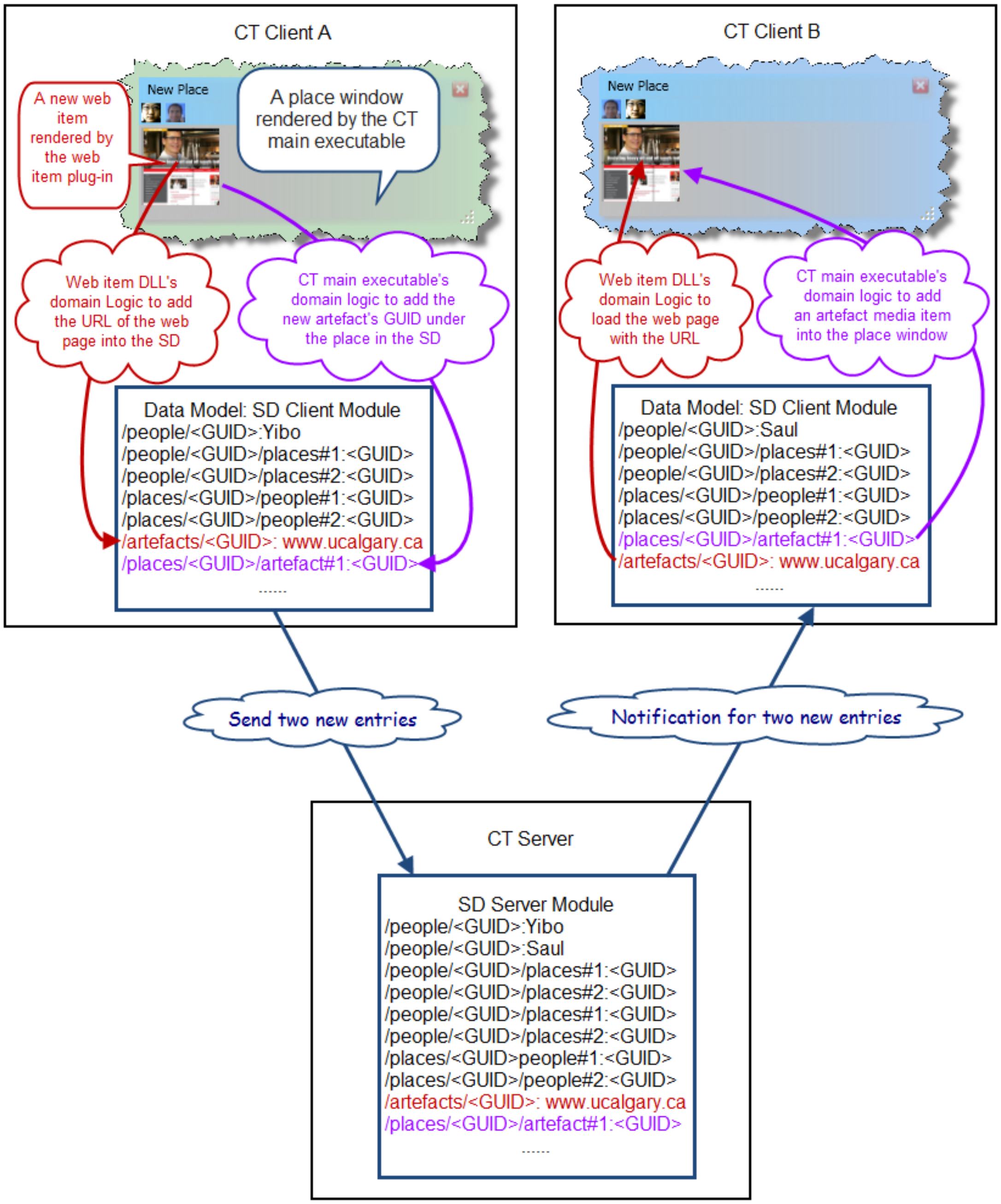
This completes the overview of the Come Together architecture. Because it is a complex system, the various sections below provide additional details and discussions of its particular architectural components.

6.2 The Shared Dictionary

The Shared Dictionary is best described as a centrally coordinated hash table that provides shared access to data by clients via subscriptions, notifications, and the ability of clients to add, change or delete particular data elements. The dictionary is hierarchical, where each element (or path in the dictionary) is defined as a key-value pair. *Keys* are strings, with hierarchical elements distinguished by '/' delimiters. *Values* can be any serializable data type, e.g., integers, strings, video frames, objects, etc. For example, the current video frame of a specific person could be denoted as /people/<GUID>/liveVideoFrame = < a serialized jpeg image > where the <GUID> uniquely identifies that person. The exact structure of these key / value pairs will be discussed shortly.

Each Come Together client includes an SD client that subscribes to a subset of keys (using regular expression pattern matching), where the SD client will automatically maintain a locally-cached copy of the subscribed-to key-value pairs. For example, to subscribe to any person’s video feed, one could add SD.subscribe(“/people/*/liveVideoFrame”, VideoFrameEventHandler). The first parameter specifies the pattern used to match a key, and the second is the event handler to be invoked if a key matching that pattern is added, deleted, or changed. In general, a dictionary client will subscribe to a key/value pattern. If another client adds, modifies, or deletes that subscribed-to key pattern, the server will immediately notify the subscribed client of that particular key/value creation, modification, or deletion. For example, the event handler “VideoFrameEventHandler” is invoked, where its arguments will contain the actual matched key (e.g., “/people/<GUID>/liveVidedFrame)” and the current value of that key as it appears in the client cache. The client can then retrieve the data associated with that key. However, clients that have not subscribed to that pattern will not be notified of that key/value update, nor will the key/value patterns be maintained in the local cache.

Figure 6-2 adds detail to the Shared Dictionary portion of Figure 6-1. It illustrates how Come Together exploits the Shared Dictionary to create its abstract data model containing (amongst other things) people, places, and artefacts, as seen by the key path prefixes. For now, the data model shown in Figure 6-2 is a much simplified form; the actual data model will be described shortly. Figure 6-2 illustrates what happens when one user uses the client at the top left to add a new artefact – in this case a web item - into an existing place, and what happens at another user’s client (top right). Starting at the top left, Figure 6-2 shows the client just after the user dropped a URL into his Place Window. The CT main client executable responds by creating a web media item (from the DLL) containing that URL. Two operations then happen in this client’s copy of the Shared Dictionary. First, the web item DLL adds an /artefacts/<GUID> key/value pair under the artefacts hierarchy: the first part of the path defines that its an artefact, the second defines a unique ID for this web item (the GUID), while the value holds the URL of the current web page contained in that web item. Second, the Come Together main executable adds a key/value pair under the /places hierarchy. The first part of the path defines that it’s a place, the second provides the GUID that uniquely identifies that particular place, while the third identifies not only that an artefact is added, but that it is the first artefact in a potentially longer list of artefacts (the #1). The value points to the unique GUID of that artefact, which in essence links to the artefact key created in the first step.


CT Client A
New Place
A new web item rendered by the web item plug-in
A place window rendered by the CT main executable
Web item DLL's domain Logic to add the URL of the web page into the SD
CT main executable's domain logic to add the new artefact's GUID under the place in the SD
CT Client B
New Place
Web item DLL's domain Logic to load the web page with the URL
CT main executable's domain logic to add an artefact media item into the place window
Send two new entries
Notification for two new entries
CT Server

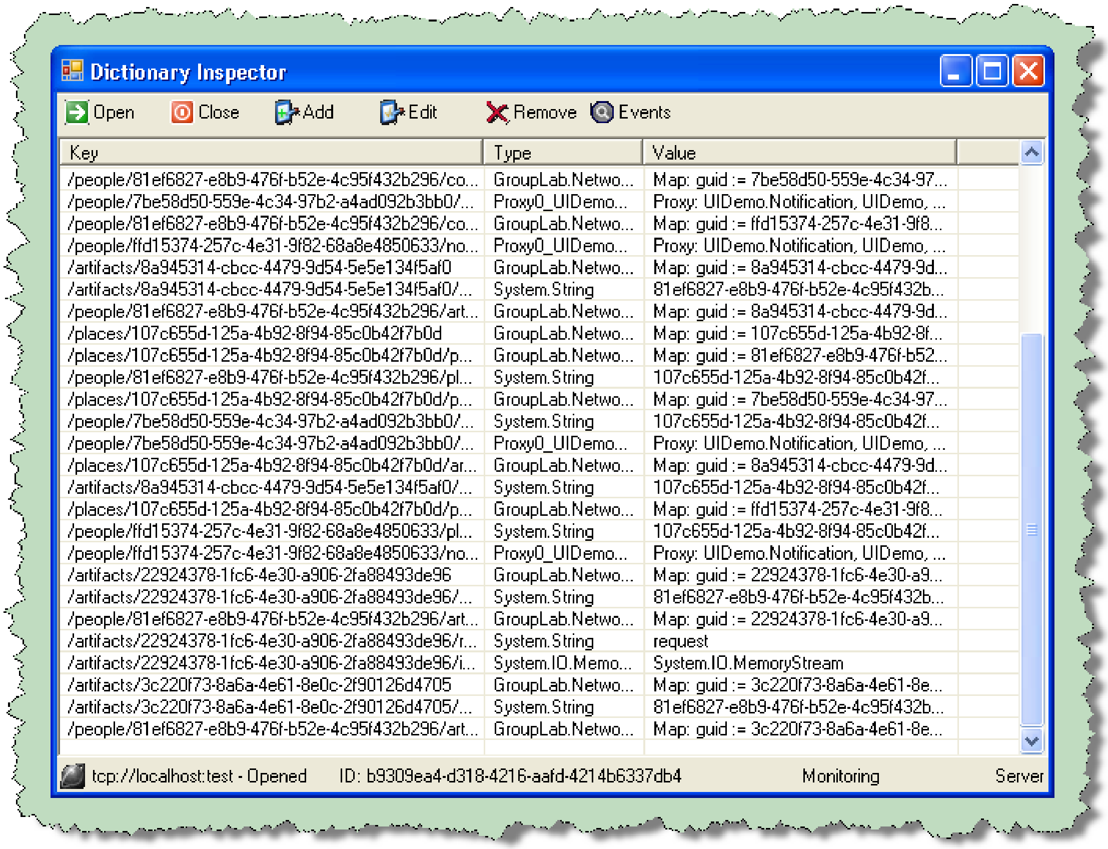
Data Model: SD Client Module
/people/<GUID>:Yibo
/people/<GUID>/places#1:<GUID>
/people/<GUID>/places#2:<GUID>
/places/<GUID>/people#1:<GUID>
/places/<GUID>/people#2:<GUID>
/artefacts/<GUID>: www.ucalgary.ca
/places/<GUID>/artefact#1:<GUID>
......

Data Model: SD Client Module
/people/<GUID>:Saul
/people/<GUID>/places#1:<GUID>
/people/<GUID>/places#2:<GUID>
/places/<GUID>/people#1:<GUID>
/places/<GUID>/people#2:<GUID>
/places/<GUID>/artefact#1:<GUID>
/artefacts/<GUID>: www.ucalgary.ca
......

SD Server Module
/people/<GUID>:Yibo
/people/<GUID>:Saul
/people/<GUID>/places#1:<GUID>
/people/<GUID>/places#2:<GUID>
/people/<GUID>/places#1:<GUID>
/people/<GUID>/places#2:<GUID>
/places/<GUID>people#1:<GUID>
/places/<GUID>/people#2:<GUID>
/artefacts/<GUID>: www.ucalgary.ca
/places/<GUID>/artefact#1:<GUID>
......

Figure 6-2 A concrete example of CT's client-server communication

The Shared Dictionary client then sends those keys to the server (Figure 6-2, arrow, left side), which updates it central data store (Figure 6-2, bottom). The server checks those keys against a list of clients subscribed to key patterns that match these just-added keys. Because Client B matches, it generates added entries to Client B (Figure 6-2, arrow, right side). The Shared Dictionary of the 2nd user's Client B receives that notification and updates its local cache (Figure 6-2, top right). The notification invokes the event handler in the Come Together main executable, where as a consequence it adds a new web media item to that particular place. The web item, in turn, receives the artefact notification through its event handlers, and thus updates its contents to retrieve and show that web page in the 2nd user's interface.



Key	Type	Value
/people/81ef6827-e8b9-476f-b52e-4c95f432b296/co...	GroupLab.Netwo...	Map: guid := 7be58d50-559e-4c34-97...
/people/7be58d50-559e-4c34-97b2-a4ad092b3bb0/...	Proxy0_UIDemo...	Proxy: UIDemo.Notification, UIDemo, ...
/people/81ef6827-e8b9-476f-b52e-4c95f432b296/co...	GroupLab.Netwo...	Map: guid := ffd15374-257c-4e31-9f8...
/people/ffd15374-257c-4e31-9f82-68a8e4850633/no...	Proxy0_UIDemo...	Proxy: UIDemo.Notification, UIDemo, ...
/artifacts/8a945314-cbcc-4479-9d54-5e5e134f5af0	GroupLab.Netwo...	Map: guid := 8a945314-cbcc-4479-9d...
/artifacts/8a945314-cbcc-4479-9d54-5e5e134f5af0/...	System.String	81ef6827-e8b9-476f-b52e-4c95f432b...
/people/81ef6827-e8b9-476f-b52e-4c95f432b296/art...	GroupLab.Netwo...	Map: guid := 8a945314-cbcc-4479-9d...
/places/107c655d-125a-4b92-8f94-85c0b42f7b0d	GroupLab.Netwo...	Map: guid := 107c655d-125a-4b92-8f...
/places/107c655d-125a-4b92-8f94-85c0b42f7b0d/p...	GroupLab.Netwo...	Map: guid := 81ef6827-e8b9-476f-b52...
/people/81ef6827-e8b9-476f-b52e-4c95f432b296/pl...	System.String	107c655d-125a-4b92-8f94-85c0b42f...
/places/107c655d-125a-4b92-8f94-85c0b42f7b0d/p...	GroupLab.Netwo...	Map: guid := 7be58d50-559e-4c34-97...
/people/7be58d50-559e-4c34-97b2-a4ad092b3bb0/...	System.String	107c655d-125a-4b92-8f94-85c0b42f...
/people/7be58d50-559e-4c34-97b2-a4ad092b3bb0/...	Proxy0_UIDemo...	Proxy: UIDemo.Notification, UIDemo, ...
/places/107c655d-125a-4b92-8f94-85c0b42f7b0d/ar...	GroupLab.Netwo...	Map: guid := 8a945314-cbcc-4479-9d...
/artifacts/8a945314-cbcc-4479-9d54-5e5e134f5af0/...	System.String	107c655d-125a-4b92-8f94-85c0b42f...
/places/107c655d-125a-4b92-8f94-85c0b42f7b0d/p...	GroupLab.Netwo...	Map: guid := ffd15374-257c-4e31-9f8...
/people/ffd15374-257c-4e31-9f82-68a8e4850633/pl...	System.String	107c655d-125a-4b92-8f94-85c0b42f...
/people/ffd15374-257c-4e31-9f82-68a8e4850633/no...	Proxy0_UIDemo...	Proxy: UIDemo.Notification, UIDemo, ...
/artifacts/22924378-1fc6-4e30-a906-2fa88493de96	GroupLab.Netwo...	Map: guid := 22924378-1fc6-4e30-a9...
/artifacts/22924378-1fc6-4e30-a906-2fa88493de96/...	System.String	81ef6827-e8b9-476f-b52e-4c95f432b...
/people/81ef6827-e8b9-476f-b52e-4c95f432b296/art...	GroupLab.Netwo...	Map: guid := 22924378-1fc6-4e30-a9...
/artifacts/22924378-1fc6-4e30-a906-2fa88493de96/r...	System.String	request
/artifacts/22924378-1fc6-4e30-a906-2fa88493de96/i...	System.IO.Memo...	System.IO.MemoryStream
/artifacts/3c220f73-8a6a-4e61-8e0c-2f90126d4705	GroupLab.Netwo...	Map: guid := 3c220f73-8a6a-4e61-8e...
/artifacts/3c220f73-8a6a-4e61-8e0c-2f90126d4705/...	System.String	81ef6827-e8b9-476f-b52e-4c95f432b...
/people/81ef6827-e8b9-476f-b52e-4c95f432b296/art...	GroupLab.Netwo...	Map: guid := 3c220f73-8a6a-4e61-8e...

tcp://localhost:test - Opened ID: b9309ea4-d318-4216-aafd-4214b6337db4 Monitoring Server

Figure 6-3 The stock shared dictionary monitor as the CT server

Details of the data model. The Shared Dictionary includes a special client, called the Inspector; it is a standalone monitor that dynamically shows all the live key/value contents, including the data type, held by the server. A portion of the data held by a running Come Together system is visible in the Inspector of Figure 6-3. While it should be evident the data model is fairly extensive, its structure is actually based upon a handful of simple naming conventions of the key hierarchy defined by Come Together.

In particular, all CT dictionary entries are structured into three branches of the hierarchy that identify the primary CT components: people, artefacts and places. These branches are specified by the identifier in the first part of the key path. For example, In Figure 6-3, we see keys beginning with /artefacts/ that collect information relating to artefacts, /people for people, and /places for places. Of course, there are myriads of people, places and artefacts. To distinguish between them, the second part of the path is a globally unique identifier (a GUID), that uniquely identifies a particular person, place, or artefact. For example, the unsorted list in Figure 6-3 has multiple references to "/people/62dc…005…" which identifies data associated with a particular person.

The People branch. Table 6-1 discloses a subset of people information and describes what a particular key/value combination does. For example, the "/people/GUID key's value contains a map (akin to a record) that stores basic account and configuration information as collected from that person's 'Me tab': the person's display name, whether to show one's video feed to others, and the frame rate of that video. Other child keys can capture more complex data types, such as the person's display image (under "/people/<GUID>/staticAvatar"), and the most current video frame of that person ("/people/GUID/liveVideoFrame"). People entries can also reference other specific people and artefacts. For example, the "/people/<GUID>/buddies" key contains a list of GUIDs identifying other people. By knowing those GUIDs, the software can construct a key (e.g., /people/<GUID>) and retrieve data about that particular buddy. Similarly, the "/people/artefacts" and "/people/places" collect a list of the artefacts associated with that person and the places that person belongs to. Finally, people entries also contain various sub-root entries that can be used to construct toast notifications. For example, /people/<GUID>/notifications/<GUID> at the bottom of Table 6-1 identifies a specific entry that identifies the notification type (a 'Place Invitation') and other information specific to that notification type (e.g., the GUID of a place) that will be used at some point to compose a notification message.

Key	Value	Description
/people/<GUID>	#displayName = Yibo #showVideo = true #maxFrameRate = 5	A map storing a person's account setting: display name, whether to show the video feed, the maximum video frame rate
/people/<GUID> /staticAvatar		The person's static image
/people/<GUID> /liveVideoFrame		The current video feed frame of the person's
/people/<GUID> /buddies	#0 = UserGuid0 #1 = UserGuid1 #2 = UserGuid2…	A list of Guids of the person's buddies
/people/<GUID> /artefacts	#0 = ArtefactGuid0 #1 = ArtefactGuid1 #2 = ArtefactGuid2…	A list of Guids of the person's private artefacts
/people/<GUID> /places	#0 = PlaceGuid0 #1 = PlaceGuid1..	A list of Guids of the places the person belongs to
/people/<GUID> /notifications/<Guid>	#notificationType = PlaceInvitation #place = PlaceGuid2	A map storing a notification sent to the person
Other notifications…		

Table 6-1 Dictionary entries for a person and the notifications he received

The Artefact branch. Similar to the above, entries with key paths prefixed with "/artefacts/<GUID>" store information about an artefact. Table 6-2 shows a subset of an artefact's key/value pairs. The top level includes information common to all artefacts, such as its name, the GUID of the person who created it (its 'owner') and so on. All artefacts also have a type "/artefacts/<GUID>/type", which is the name of the DLL file that implements that artefact and which the system loads as needed. Similarly, all artefacts have a list of places that it belongs to, stored as a map in "/artefact/<GUID>/places", and thus an artefact can also retrieve (via that place) what participants can access it. Finally, the DLL can add any keys under the /artefact/<GUID> path to compose key/value pairs specific to its own need. Table 6-2 shows a sample, where the webitem.dll added a key "/artefact/<GUID>/url"that contains the path of the web page it should display. Of course, there are many other custom fields included with the web items that are not shown in the table.

Key	Value	
/artefacts/<GUID>	#name = "Wikipedia" #owner = aUserGuid	A map storing an artefact's name and the Guid of its owner
/artefacts/<GUID>/type	webitem.dll	The file dll name associated with that artefact
/artefacts/<GUID> /places	#0 = PlaceGuid0 #1 = PlaceGuid1…	A list of Guids of the places the artefact belongs to
/artefacts/<GUID>/url	http://www.wikipedia.org	A custom field for a web item, created by the dll
/artefacts/<GUID>/… for example, artefacts/<GUID>/url	<any custom data> for example, http://www.wikipedia.org	Other custom key/value pairs specific to various artefact types as created by the dll for example, a custom field for a web item

Table 6-2 Dictionary entries for an artefact

The Places branch. Places follow the same pattern, as illustrated in Table 6-3. The top level key is a map that includes basic properties of the place (e.g., its name, its access level, the visibility of its contents to others). Other keys include a list of all people and artefacts (via their GUIDs) that belong to that place.

Key	Value	
/places/<GUID>/…	#name = "New Place" #isPublicPlace = true #peopleVisibility = Thumbnail #artefactVisibility = Thumbnail	A map storing configuration of a place: its name, its publicity, its public visibility.
/places/<GUID>/people	#0 = UserGuid0 #1 = UserGuid1…	A list of Guids of the place's participating members
/places/<GUID>/artefacts	#0 = ArtefactGuid0 #1 = ArtefactGuid1…	A list of Guids of the place's artefacts

Table 6-3 Dictionary entries for a place (locale)

In summary, the three hierarchical branches cleanly separate out the three primary entities of Come Together: "/people/" capturing people (and notifications they should see), "/artefacts/" capturing artefacts, and "/places/" capturing places. Figure 6-4 illustrates this, where our abstract, virtual hierarchical data model is translated to the flat hashtable records of the Shared Dictionary. Each branch defines standard names of its children, so that Come Together can both populate and retrieve specific data. Each branch also specifies its relationship to other branches, by collecting pointers (via GUIDs) to those other branches (e.g., a place collects pointers to the people within it). Mutual, bi-directional relationships are maintained by storing the respective GUIDs under each other's entries.

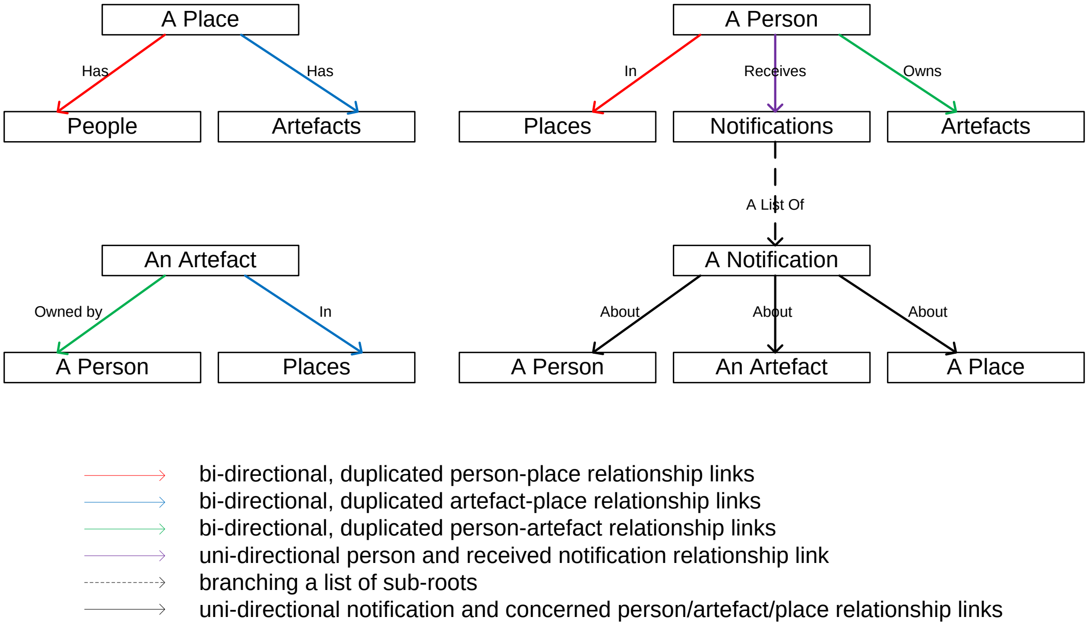

A Place
Has
Has
People
Artefacts
A Person
In
Receives
Owns
Places
Notifications
Artefacts
A List Of
An Artefact
Owned by
In
A Person
Places
A Notification
About
About
About
A Person
An Artefact
A Place
bi-directional, duplicated person-place relationship links
bi-directional, duplicated artefact-place relationship links
bi-directional, duplicated person-artefact relationship links
uni-directional person and received notification relationship link
branching a list of sub-roots
uni-directional notification and concerned person/artefact/place relationship links

Figure 6-4 The abstract hierarchical data model of the CT system

6.3 Come Together client application

The Come Together client is what the end user normally sees, where its interface was illustrated in the previous two chapters. As mentioned, it is implemented as an MVP (Model-View-Presenter) architecture (Figure 6-1). The model layer (data store and networking) is solely managed by the shared dictionary client module as described above and illustrated in Figures 6-1 and 6-2. The user interface elements—the Console, the three presentation levels of media items, the media item window, the places windows, the places strips, and the pop-up notification message toasts—are coded as the view layer. The presenter layer is the domain logic: the software components that translate user actions from the interfaces down to the data model, which they present, and reflect the changes in the data model up to the user interfaces (Figure 6-1, 6-2).

Come Together supports different types of media items with DLL plug-ins that are compiled separately (Figure 6-1). However, the client and plugins interoperate, where it is the responsibility of a plug-in to store/retrieve the custom data it needs and to draw the contents held by the media item. First, the DLL does not create its own shared dictionary. Rather, the main Come Together client executable hosts the singleton shared dictionary. It passes a path within the shared dictionary to the DLL plug-in, which the DLL can use to populate that branch with information specific to its needs (as illustrated in Table 6-2). Second, the user interfaces provided by the media item DLLs are passed to and embedded into the user interface (i.e., the empty windows and containers) provided by the CT client proper.

The user interfaces of the main Come Together client and its stock media items are developed with Microsoft's Windows Presentation Foundations (WPF) (Eisenberg & Bennage, 2008). WPF works well with an MVP architecture. In particular, the WPF UI components in the view layer handle user inputs and interpret them into abstract domain logic functions provided in the presenter layer. In turn, this layer modifies dictionary entries, which are transmitted to other clients as notifications.

The presenter layer of the main Come Together client program consists of domain entity objects: a person, an artefact, a place, a notification. Stock media item DLLs include type-specific domain objects, for example a list of photos (for a photo item), a textual chat message (for a chat item), a web page (for a web item), a screen image (for a screen item), a document (for a document item). The shared data layer—the singleton shared dictionary client module—is controlled by the presenter layer—the domain logic—of the Come Together Client program proper and its media item DLLs. Each domain logic object presents and manipulates its related set of key-value pairs.

These domain logic objects separate the data model—the shared dictionary client module—and the user interface components in the view layer. They receive abstract commands from the view layer, and translate them into concrete changes applied to the data layer—the shared dictionary (Figure 6-1). On the other hand, other shared dictionary client modules in other Come Together clients see the changes made and in turn notify their presenter layer of the domain logic objects. Then they translate these notifications back to concrete instructions applied to the user interfaces in the view layer (Figure 6-1). For example, Table 6-4 below lists a partial commands/actions pairs:

Commands from one client	Actions on other clients
create/remove/rename a place	Create a place as a container for media items; remove it from the place list; update its title.
create/update/remove/rename an artefact	Create/remove a media item; set/update its content; update its title.
add/remove a person/artefact to/from a place	Add/Remove a media item to/from a panel in a place window or strip
create/configure a user's account settings	Update the displaying name of a person/artefact/place
update a person's live video frame	Refresh the live video frame/static avatar image of a person
send a notification to a person's user account	Pop up a notification message toast

Table 6-4 command/action pairs for clients' communication

Further details about how media item plug-ins are developed, compiled, and integrated into Come Together are in Appendix B.

6.4 Technical issues and possible solutions for future work

The current version of Come Together is still at the prototype stage. While fully functional and demonstrable, it is not stable enough for general deployment or for ongoing regular use. The reason is largely due to the difference of creating a research prototype as a one-person effort vs. constructing a robust groupware system. The system, as is, suffices to illustrate the concepts.

Specifically, there are flaws in the implementation that I generally attribute to the way I constructed the software using the latest Microsoft .Net 3.5 framework with the WPF technology and the SD/GT toolkit.

6.4.1 WPF issues

I decided to adopt the new Microsoft .Net 3.5 technology with WPF, largely because Come Together's design requires modern user interface features such as full graphics support, transparency, gradients, animations, transforms, a better layout system to arrange UI components and various special visual effects. WPF also supports the MVP software architecture more consistently by its data binding feature to couple domain objects directly to UI components without extra controlling logic.

The problem is that the WPF version I used was relatively new, where it exhibited many (often unexpected) issues in terms of robustness and performance. Since I exploited most of WPF features during the development of CT, I experienced many known and unknown problems. Some of them are very fundamental and have not yet been resolved by the vender. For example,

- Data-binding between UI components and domain objects breaks during run-time. The consequence is that some user actions on the user interface become ineffective, or that UI components do not always update their contents.
- Fake native WPF control (claimed to be native by the vender), wrapping legacy WinForm controls, causes incompatibility issues with other genuine WPF controls.
- Hit testing does not work in certain situations.
- Desktop/always-on-top layers are rendered reversely in certain situations.
- There are various known building issues in Visual Studio for WPF projects, e.g. resources fails to compile into binary, and using certain package/file names causes building to fail without a clue.
- Re-sizing UI components may cause a dead loop of resized events.
- The UI focus may shift randomly.

Of course, issues in software development tools are not uncommon, and often known work-arounds are developed. However, WPF had too many issues for me to address in a timely way, and most of them are unknown. I spent a large amount of time tracking these issues down, where I had to write simple programs to isolate and reproduce these issues (although some issues proved non-reproducible outside my program). Some issues had no simple workaround. WPF also proved too resource-intensive, where loading a large number of WPF controls in CT’s somewhat complex user interface could cause an “out-of-memory” exception. Therefore, the current version of CT cannot guarantee robustness for practical use.

That said, WPF is still very promising. As it matures, the current robustness and performance issues will likely disappear.

6.4.2 SD/GT issues.

The SD/GT toolkit is designed for rapidly prototyping groupware applications with the Microsoft .Net framework. The main advantage of using the shared dictionary for a client-server system is the ease of deployment and simplicity of usage. The server module and the client module of a shared dictionary system provide a unified programming interface—Grouplab.Networking.SharedDictionary. The first instance of a SharedDictionary object results in the creation of a server. Additional instances of the SharedDictionary objects are created as clients connecting to that server, forming a distributed system without additional configuration. The SharedDictionary interface—either the server or client module—works as a normal hashtable and thus is fairly simple for average programmers to understand and use. Developers are spared the effort of coding the networking of the entire distributed system, because to them it looks like all CT clients access the same data repository in a local shared memory.

However, the shared dictionary design is not particularly conducive to the complex data model held by CT. Figure 6-5 is an entity relationship diagram for the four CT domain entity types: people, artefacts, places, and notifications. As we can see, CT has a somewhat complex relational data model, with seven relationships between four entities. The best use of the shared dictionary, as a flat hashtable, is to store data in an abstract, virtual hierarchical model (Section 6-2). We have to use four types of virtually hierarchical trees rooted at each entity of one of the four domain entity types. Due to the subscription-notification mechanism, a shared dictionary client only keeps a partially and locally cached copy of the entire data store on the server. Thus a join operation to query the relational data model is not possible through a thorough search. As a result, bi-directional, duplicated links must be maintained between the four trees (Figure 6-4). This makes every update of a relationship a non-atomic operation.

The reason why this is a problem is that the current version of the shared dictionary does not support concurrency control regarding race conditions. Yet the relationships between entities can be easily corrupted by non-synchronized events, generating such race conditions. Therefore, deploying the existing CT on a large scale is not practical in that its data integrity will become compromised over time. In addition, the version of the SD/GT used was not yet fully tested, and was still being worked on by a contract software engineer. My use of the toolkit exposed many problems, where that engineer subsequently released more than 10 versions to fix those bugs. Even so, the most up-to-date version of the toolkit may still throw un-reproducible exceptions from time to time. This compounds the robustness and deployment issues of CT.

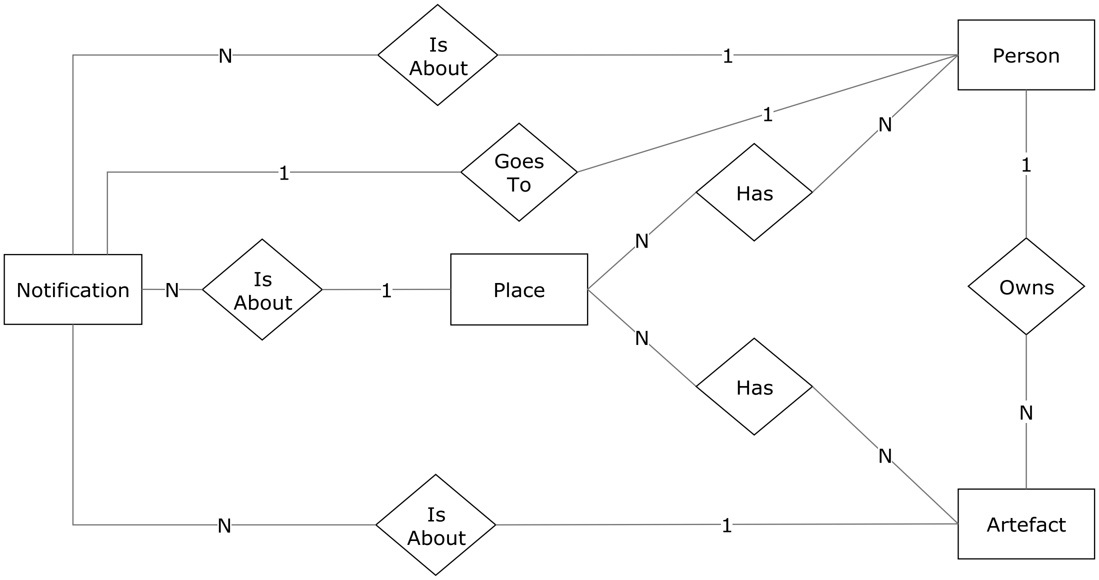


Figure 6-5 the Entity Relationship Diagram of for the four CT domain entity types

While the shared dictionary toolkit is excellent for prototyping groupware systems with a somewhat simple data model, it is not as good for robust complex systems. Due to its lack of support for complex relationships, hierarchical data models as used by SD/GT has been replaced by relational data model in modern database management systems. I recommend that the next version of CT should be developed over a true relational database management system (at the cost of easy deployment) or other means that supports a distributed relational data model. Of course, this is also fraught with issues, as most databases are not set up to handle the real time distributed performance required by interactive groupware systems..

Chapter 7. **Evaluating Come Together**

This thesis so far primarily concerns the motivation, design and implementation of Come Together. This chapter adds a preliminary evaluation of Come Together. In particular, I demonstrated Come Together to 12 individual participants (9 of them are specialists in Human-Computer Interactions), where they were able to try out and comment on its design, features, and perceived usability. Although their praises and critiques are not a replacement for a formal usability study or field deployment[2], they do indicate basic matches and mismatches between our design philosophy vs. CT's actual user interface design, and suggest how CT should be redesigned in the future. This is appropriate for a 'first pass' evaluation of the first CT design (Greenberg & Buxton, 2008).

7.1 Evaluation methodology

The basic method is an informal evaluation of Come Together, where our method went through a formal ethics approval process (see Appendix D). We recruited 12 participants, and gathered their initial reactions to our design of CT. In the first step, we used the Expert Walkthrough technique (Nielson, 1993), from an HCI perspective (9 participants), although we also included 3 non-HCI participants with computer expertise to gather their reactions. Each individual walked through various short CT scenarios running on the actual live CT system deployed as a real distributed environment consisting of three computers side by side, where she/he saw and experienced CT's features. Each individual was also given a review of CT's design goals and how CT's interface tried to meet that goal. Based on both, the individual was asked to provide his/her general impressions of system, as well as critique the

[2] As mentioned in the previous chapter, limitations in the implementation would make any real-world deployment premature.

specific features demonstrated in the walkthrough. Data collected included notes by the observer, and a video / audio record for some of the sessions.

7.1.1 Walkthrough scenario

Our first step solicited participant feedback by showing them how the system worked using a combination of think-aloud and discussion. Participants were placed in control of CT. Participants then followed various scenarios of use that collectively covered most of the CT capabilities (similar to those in Chapter 4). In particular, participants were instructed to activate different features as they stepped through the scenario. We did not follow a strict scripted scenario walkthrough, as we also allowed participants to explore user interface functions on their own. During the course of this exercise, participants were questioned regarding their opinions of CT and its features, including praise, criticisms, ease of use, and other relevant factors.

• Supporting light-weight locales with “places”
○ Implicit formation of places.
○ Persistent places.
○ No imposed hard walls on a place to allow non-members view its content.
○ No imposed invitation/acceptance mechanism to add a person to a place.
○ No imposed request/authentication mechanism to join a place.
○ No imposed social protocol or roles in a place.
• Supporting the notion of centre-periphery.
○ Media items for people and artefacts: the three presentation levels.
○ Places: window vs. stripe; icon-only view vs. hybrid view; opaque vs. transparent.
○ Desktop space vs. CT console.
• Treating people and artefact equally.
○ In terms of presentation: all as media items.
○ In terms of their being: all as independent entities, able to exist outside any locale, inside one or multiple locales.

Table 7-1 CT evaluation aspects

While walkthroughs usually concern themselves with usability issues, our walkthrough specifically probed participants for their comments on CT’s key design aspects. As summarized in Table 7-1, we introduced these design aspects at appropriate times during the walkthrough.

7.1.2 Post demonstration discussion

After the walkthough, participants were asked to reflect on CT as a whole. In particular, I revisited each design goal in the above bullet list, where I repeated the design philosophy and how CT tried to meet that in its implementation. The goal here was to confirm comments previously given by the participant during the walkthrough, and prompt a more detailed discussion about her/his rationale and/or suggestions. During this phase, participants usually provided comments from a holistic design perspective.

7.2 Analysis of comments

Analysis was via a fairly informal review of notes and of the video/audio record. Because this is an early system, we tended to look for 'big effects' that would be easy to ascertain from the commentary. We organized the detailed analysis of comments around the design goals summarized in Table 7-1. While our discussion below often highlights any criticisms of CT, we should stress that CT was generally received favorably. For example:

> "Very cool stuff. Right in the middle of your demonstration there, I think actually I could really go for something like this."

7.2.1 Support light-weight locales with "places"

Implicit formation of places. CT allows one to create a place implicitly by bringing people and artefacts together. This is the primary feature of the system and it seems to be successful at large. People found the various ways to create an implicit place intuitive. When asked to try to create a place with specific people and artefacts, they all took the right actions: dropping artefacts on a person, and dragging together people and artefacts' media items. Participants' responses echoed their success:

> "It makes a lot of sense to me. I agree. Very nice."

> "Yeah, you don't need to pre-configure a container."

All agreed with the idea that CT does not require a name upfront for a new place, as this made place creation light-weight. However, most people suggested that a new place should be assigned a better default name rather than "New Place".

> "I would have the name of a place be more intelligent, based on who is in and what I am trying to do with it."

> "The default [name should be] just a list of what it is. Just a list of the things."

We also heard suggestions to support more implicit actions to create a place besides drag-and-drop or using the context menu of a media item. For example:

> "I think the chat item can always be there. Basically this can reduce to an IM system. For here (People tab), this is like a quick access to all people I could create a place with. And to create a place, I'd like to double-click and that way I created a place [with the person]."

> "When I drag anything (a media item of a person or an artefact) out [of the CT Console], it [should form] a place."

Participants were also positive that CT could support conventional ways to create and pre-configure a place, such as when people needed to prepare a more formal meeting:

> "For example, I want to share with the whole group; I want to share with 10 people or something. I could then first create this place about the new project."

Persistent Places. Despite the IM-styled light-weight creation of a place, CT also supports long-standing locales by allowing places to persist. This allows people to come and go. In general, people were positive about persistent places. They liked the way CT allowed them to add or remove people and artefacts in a place as it evolved.

However, most participants were critical about the relatively heavy-weight interface actions (vs. the light-weight place creation actions) required when leaving a place. In particular, they have to explicitly raise a context menu of a place to control what happens when one leaves it, i.e., whether it should persist or whether it should completely disappear.

> "There should be an easy way to kill [a place if we don't want it to persist]."

> "It seems to me a lot of work to manage, the context menu [of a place]. Like (I think the intention is) you are trying to create something lightweight to deal with."

> "[Instead, maybe] you can put something (a button) there (on the title bar of a place), as long as there are two levels, like a dialog [box for confirmation to quit a place]."

Because the default is for places to persist, people thought that many persistent, stale places would arise by accident. This adds to the overall burden of managing the collection of places over time. To ease this, some participants suggested prioritizing all the places in the CT console, and providing an easier, more light-weight way to 'garbage collect' stale places.

> "I am seeing this becomes unwieldy. It will be difficult to tell what would be important to pay attention to."

> "I think it's nice it's hard to kill them. I can't imagine: over time this will get bloated; lead to management problems. Maybe a trash can or something [to make them easier to remove]."

> "You could really have something like a trash bin in here, in some hidden place you can still access."

Some saw a disparity between the light-weight creation of places and the relatively heavy-weight withdrawal process.

> "I wonder if it (implicit place formation) is too light-weight (so that you can easily create too many places). I am thinking about garbage-collection on a day to day basis."

No imposed hard wall on a place to allow non-members view its content. By default, CT makes a place open and publically viewable to non-members. It assumes an intimate community where people know and trust each other. Yet CT allows places to become private if desired. Some participants agreed with the idea of public places:

> "Well that makes sense. Say people have a conversation in the background. You hear they are talking. You can capture a part of what they are talking about. I think it is kind of reflective of real life."

> "I mean if you make it public by default, you can turn it private before you start adding private content to it."

Yet other participants pointed out that we should have been more careful about 'public by default':

> "I think the default, whether the default is public or private, could depend on what the scenario is."

> "I would say I wouldn't have expected it. I think it's not a default behavior I would feel comfortable with. At least I would like the option to say by default make it private or by default make it public. Because I think this is a little bit critical."

> "I think this is not a good default setting. I would say if you just make [the decision to], by default, show those icons in here and it would be totally sufficient. You see the name; you see little icons; don't show content; just show the type of media items. But I agree showing these places is a good thing. You still have awareness of all the places. I would say the headline, the

> participants are public but not the content. It is a good idea that you just browse quickly through [the places] from time to time to see the other places that are around, maybe something like an interesting conversation that I might be interested in. I think that's a good way."

Another common suggestion was that the system should provide better cues to participants to indicate a place's public and open state:

> "[to be public or private] depends on the context. In this case, you do want the ability to turn that off (turn a place private). But more than that, people need to be aware of what they are getting when they start this program at the first place."

> "Somehow make it clear [that a place] is public, even adding to the title 'public', [it needs] some awareness mechanism."

> "I wonder if there is something more visible, like a little icon in the top corner to say it is public."

> "I wonder if you can use background colour or something to indicate [a place's] privacy [state]. You know, you can have an image of a lot of people."

A few participants noted that a virtual world is different from the real world. In the real world, people can modulate their behaviors and actions to finely control what is public *vs.* private, e.g., by whispering or by placing content out of sight. In contrast, CT provides only a few options to specify how a non-member can view a place and its contents.

> "For instance, you and I, at a table, have a discussion that can be overheard by people. If I was saying something I don't want other people to hear, I can whisper. That kind of thing is possible, and in a computer system it is difficult to replicate that exactly because in order to create those different levels, you'll necessarily add that overhead. I think it would be nice that you can make a system in which you can be whispering. But I don't see how you make it as easy as whispering is. "

No invitation/acceptance mechanism to add a person to a place. Most participants agreed that having to accept an invitation to be added into a place is not necessary for intimate collaborators. The caveat is that this assumes that the group is composed of intimate collaborators:

> "I think it works for small intimate groups."

> "For small groups, absolutely. It depends on how small it is."

Yet a few disagreed, where they thought people needed to actively decide to join groups rather than having others just add them to it. For example,

> "If you'd like somebody to be in there (a place), I would say you can't just drag him in. I think you only send the pointer: 'Hey Bob, we have this locale happening and we would like you to join us. Do you minding coming?' I would not like to come in one morning and I see I am a member of five groups I don't really [want to] actively join because I think my commitment to the groups of my interest is only there if I joined the group. And not [because] somebody else put me in the groups. It is not about the invitation. It is about the active decision that I am joining a group. It seems to me the maintainer is the boss having their employees (dragging people in) and I think that's not appropriate. What you could do is that you send an invitation somehow maybe even by dragging this person in or maybe having them as an adjacent bubble. [The dragged in person is] not yet in there but [in] this way you know this person can access the resources or maybe the preview of the resources. Maybe not a dialog box but some visual representation of what the group looks like. Do you want to be in there?"

No imposed request/authentication mechanism to join a place. Compared to the mixed reactions towards the invitation/acceptance mechanism mentioned above, all participants agreed that there is no need for a request/authentication mechanism in an intimate community, and that CT provides enough awareness for others to socially understand when people join.

> "This is ok. This is a kind of social protocol."

> "I instantly get awareness [that] Bob joins. So it is very clear."

Yet, some participant actually pointed out that CT needs the ability to "knock on the door", where people not in a public group can still request to join it .

> "I agree. However, sometimes maybe you want to knock."

> "You already have the capability to make a place private. If I really don't want to come in, I'll just do that. But I am trying to think the middle ground where you can knock on the door [to get into a place]. I think so long I have the option to restricting things, that also works."

No imposed social protocol or roles in a place. Most participants agreed with the way CT does not impose social protocols or explicit roles.

> "I think it makes sense. We can leave it at the social level."

> "People form their own social convention [which] you don't have to enforce."

> "It certainly is a valid argument. It is up to people to be polite. The system does not enforce it. To impose it, you will easily make the wrong choice as a system designer. So leave it up to the users to decide."

Yet again, this does assume that the group is composed of intimate collaborators, which may not necessarily be the case.

> "There are whole bunch of assumptions before you can say it is ok. I don't even know what all those are."

7.2.2 Support the centre-periphery continuum and individual views

Reactions were generally positive to the three presentation levels of media items for people and artefacts adapted from the Community Bar (McEwan, 2006), as were CT's other novel features supporting centre-periphery and individual views.

> "I think they work really well."

CT provided support for a place to move along the centre-periphery continuum via: two presentation levels (the icon-only view and the regular hybrid view); two forms (the standalone place window and the place strip); and visual opacity. While participants collectively thought these were all nice features, particular participants commented that some features were likely more useful than others.

> "I don't really get it how to use transparency to support the centre-periphery [continuum]. Not sure what I gain from it. [Transparent windows are] still in the way; still in front."

> "There are rare instances where you want to see through a window. I think it's worth having it there."

> "I think it is good to have the feature (icon-only view) there but I am not sure how often I can use those."

> "I find the docking more useful than the shrinking thing (icon-only view)."

> "I think it's useful: the border, the dock. Others: not so much."

In the CT console, the contents of the three tabs (People, Things, Places) are also designed to support the centre-periphery continuum. The People tab shows buddies in two presentation levels. The Things tab shows all artefacts in two presentation levels. The Places tab shows more content of "My places" than that of "Other places". Yet participants found some of these distinctions too heavy-weight and perhaps uneccessary. For example, some critiqued the two presentation levels for artefacts:

> “I am sure about this little icon thing [icons for artefacts in Things tab]. If I have two items, you have to drag them all up here and back again. I would say a little slider in the corner that I can [use to] change the thumbnail size. It would make much more sense.”
>
> “I don’t find the two level thing [icon/thumbnail hybrid view in Things tab] useful in CT console. In a Place tab, [the two level presentation are] useful. I don’t see the value to have it in here pThings tab in CT Console]. The drag-and-dropping doesn’t work that well.”
>
> “I think [the two presentation levels] makes more sense with people. Here [People tab] it is [useful]. Here [Things tab] it isn’t.”

Somewhat opposite to the above, a few participants found the discrete steps used to distinguish aspects of the centre-periphery continuum not enough to distinguish between a more centered *vs.* a more peripheral state.

> “I think this (three presentation levels layout for media items in a place) could be more gradual because I think this makes it explicit. They are still discrete.”
>
> “If I have two [places docked as strips], which is more peripheral?”

In addition, a few participants suggested sharing the layout of media items, where a person could emphasize the different levels of a place to other members, i.e., by sharing individual views.

> “It would be interesting if there is some cross-user way of specifying the layout of a place. So I’m gonna make the preference be that this (one particular media item) would be large when you join the room, as oppose to that you have to go through all the documents and see which is important. When I first join, I want to see the big one. As a default [media item layout of a place], they can change it if they want.”
>
> “We have a shared set of resources in the view (a place) but we don’t have a shared view. So having a shared view could have value. I feel this is very powerful because if you don’t have a shared layout, and shared positioning, it’s not a shared locale I think. It’s just a shared list of items.”

7.2.3 Treating people and artefacts equally

In CT (as in the Community Bar), artefacts and people are presented similarly, i.e., both are wrapped in media items, which are blended into a place. CT takes this one step further by treating people and artefacts equally in terms of how both can exist as an independent media items in and out of any places. Our participants found this very innovative and useful:

> “It’s an interesting way to go about it.”
>
> “Having them (people and artefacts) together in a place as in real life is good. I think it is useful to have them visually together.”
>
> “I think it makes totally sense. For example, I am just adding my sketches; I am making this a nice collection. From unsharing things, they (shared artefacts) can come to the private things and they stay there; they don’t disappear; then I can share them again.”

However, most participants wanted a visual cue to distinguish people and artefacts:

> “I think it is an advantage to have them (people and artefacts) as the same thing (same presentation in media items). But one thing I was noticing is that sometimes we have to look at it for a little while before you understand which is which. I wonder if there is a way that can help distinguish them a little bit better even if they are treated equally. Anything from a different colour border to distinguish people and artefacts, or an icon in the corner to show you know this is a web, this is a person, something like that.
>
> “I don’t think they are the same. They might be the same in terms of how you use them, like, drag them around. But I think In terms of presentation, people are more important than things. Knowing somebody is in this place is more important than knowing this file is in this place. In terms of presentation, I am less comfortable. In terms of how they are used, I’d say go the other way. Dragging them around, having them all take their own sizes. That’s fine. I want to know this is a person and that is a picture of a person.”

One participant somewhat disagreed with this practice of treating people and artefacts equally:

> “I don’t entirely agree. [In terms of the] way people and artefacts are arranged [in a place], I think people and artefacts are different. Knowing there is one person and ten artefacts *vs.* ten people and one artefact makes a huge difference. I understand the concept of making people and artefact equal citizens but I think they are not entirely equal. I am not entirely against or for it. But I think this is a really nice first step.

7.3 Discussion and design implication

The positive reception of CT by study participants indicates that our design generally matches our goals. However, participants’ criticisms and suggestions also point out directions for future improvements.

Light weight locales. Our study verifies that a CT place generally matches people's notion of a light-weight locale. Forming a place via various simple actions proved light-weight and intuitive. Yet participants suggested that CT could have even more fluid actions to ease the task of creating implicit places for various interactions. For example, double-clicking a person may cause the creation of a place with a default chat item and two persons. Participants also suggested that CT can be better integrated with the operating system and other applications. Examples include sharing an artefact by raising its context menu, or by dropping a person onto an artefact (e.g., a web page in Internet Explorer, a file in Windows Explorer and etc.). A more descriptive default name for a place should also be incorporated into future design. As well, a place should have better controls to manage its layout and how media items are arranged.

A place supports a persistent locale so that people can come and go. However, leaving a place is too heavy-weight, and could lead to a clutter of stale places. Alternative designs are needed to make this lightweight, e.g., by allowing one to quit a place by clicking a button and then confirming it via a dialog box, or by providng a recycle bin for deleting places.

As a side effect to the heavy-weight place withdrawal issue, a user may be overwhelmed by a large number of places appearing under the CT console's Places tab. Eventually, this can become unmanageable. The CT Console design was intended to mimic the simplicity of the IM buddy list, where it would manage artefacts and places as well as people. It trades off simplicity rather than power. Yet future designs could bring back some power. One option is to display all places along the centre-periphery continuum, with those at the center having more visual detail and the ones at the periphery being hidden and archived. Another possibility is to allow one to sort the Place list by various criteria — idle time, participation intensity, activity, etc. — with perhaps those at the bottom list deleted. A place can also be assigned with an expiry date, where the system garbage-collect it.

The notion of 'hard walls' needs to be fine-tuned. This includes how non-members are prevented from viewing a place's content, invitations and acceptances, an even the social hierarchy within a place. First, these various open (public) features and the flat structure of a place all depend on the premise that collaborations are formed within an intimate community.

Yet we should not assume that such a community is actually using the system. Sensible defaults should depend upon on the community, such as whether people usually want a place to be open (public) or closed (private). Indeed, our study shows that participants have mixed opinion about these various open/close issues. Come Together should not impose universal defaults. Rather, we should let a mix of site administrators and end users specify their own preferred defaults settings. Regardless of the settings chosen, CT's role should be to provide clear awareness of the current state of a place, e.g., whether the place and its contents is publically visible.

Centre-periphery and individual views. CT uses different presentation forms at different levels for people, artefacts, and places to constitute an individual view set. In general, our participants found these user interface features valuable to support the centre-periphery continuum and individual views of users. However, CT's solutions were not complete or flexible enough to satisfy every user's personal needs. For example, people wanted more than three presentation granularities for places, but less for artefacts. We should explore other designs that present different granularities of the centre-periphery continuum on a feature by feature way.

Integration of people and artefacts. Like the Community Bar (McEwan, 2006), CT wraps both people and artefacts into media items. CT extends this by allowing people and artefacts to exist outside of a place, ore inside one or more places. However, people and artefacts are indeed different in our real life. People commented that we can make media items for people and artefacts more visually distinct. In addition, there is a disparity between the centre-periphery granularity needed for people *vs.* artefacts. The two presentation level design for artefacts in the Things tab can be confusing or over-complicated. Again, this is a trade-off between simplicity and consistency which has to be explored in future designs.

7.4 Conclusion

The study affirms that our design goals for CT are all on the right track. However, the study also suggested many aspects that require fine-tuning. The study also suggested that we have to be careful about the assumptions made of the expected community of users. As a consequence defaults should not be universally applied, but should be suggested by the site and/or end user. Ultimately, we need to experiment with alternative user interfaces and consider their design trade-offs. This is fairly typical of iterative design. Our initial design passes the overall test, where it appears to match our design goals. Yet fine-tuning is needed to make it even more usable and flexible.

Chapter 8. **Conclusion and Future Work**

This research was inspired by the Community Bar to support light-weight formation and light-weight working practice of small, light-weight locales. First of all, I revisited the theoretical foundation underlying the design of the Community Bar—the Locales Framework—and suggested basic design considerations to support light-weight locales. I then reviewed various genres of groupware systems from a Locales perspective and discussed how they support or hinder light-weight formation and working practices of locales. Next, I introduced Come Together, a light-weight groupware system based on those design considerations, incorporating the merits of existing groupware systems, especially instant messengers and the Community Bar. Lastly, I described the technical aspects of CT and discussed people's initial reactions to CT from a preliminary evaluation. This final chapter provides a summary of the contributions and the future direction of this research.

8.1 Contributions

In our everyday life, most of people's gatherings are informal, casual, opportunistic and somewhat ad-hoc. However, such gatherings cannot occur easily between distance separated people. Nevertheless, people use computer and networking technologies to facilitate mutual awareness and interactions via virtual means. The Community Bar was designed to support such meetings but a study of its use revealed that CT's "places" are too heavy-weight to emulate locales for the light-weight working practices of such casual gatherings. This inspired our goal of supporting light-weight group working practices in groupware system design.

I made several significant contributions in the thesis. First is the design considerations, derived from the Locales Framework and existing groupware systems, to support light-weight working practices of locales. Second is our software prototype, Come Together, which demonstrated the viability of these design considerations. The new paradigm of light-weight locale formation in Come Together is the most important feature among all: a place is formed implicitly when people and/or artefacts are drag-and-dropped together and it does not require a name when created. Other lesser but still important contributions include:

Open and public locales. CT also supports other light-weight locale working practices: non-members can view a place's content, no invitation/acceptance/authentication is required for one to join or be added into a place, and there is no imposed social hierarchy within a place.

Integration of people and artefacts. CT introduced the idea of treating both people and artefacts equally, allowing them to exist outside of a place and to be added into places opportunistically.

More customizable individual views. CT provides various presentation forms at different levels for people, artefacts, and places to customize an individual view set.

A preliminary study of CT. One important critique is not to make assumptions of the expected community of users. Defaults should not be universally assumed and applied, but should be suggested by the site and/or end user.

8.2 Future work

Our study of CT has verified the validity of our research directions. In the future, and based on our evaluation, we need to experiment with design alternatives to better match and fine tune CT with the current design goals.

The design of CT took a holistic approach and sought to integrate many concepts in the Locales Framework: people, artefacts, communication, locales, centre-periphery continuum, individual views. However, the actual design seems to partially fragment the framework. From the preliminary study of CT, I recognized a few possible, artificially or arbitrarily separated concepts.

Centre-periphery for people and artefacts in and out of locales. CT supports three presentation levels for people and artefacts in and out of places. However, a certain person or artefact's presentation level in a place is not connected to that outside a place (in CT

Console). This could be related to why participants in our study did not find drag-and-dropping artefacts between two presentation levels under the Things tab useful. A person or an artefact 's movement along the centre-periphery continuum in a place should be somehow reflected in the user's individual view of all people and artefacts in CT Console.

Civic structure and centre-periphery for locales. The support for civic structures is almost left out of CT. We cannot accurately map the centre-periphery continuum of locales on a single dimension without considering civic structures. We have learnt that CT needs more presentation granularities for places to support locales along the centre-periphery continuum. Civic structures can perhaps be incorporated to determine which locales are more relevant to the user.

Trajectory and centre-periphery. Trajectory is anther almost left-out element in CT's design. A person's engagement level to another person and/or an artefact in a locale can be closely linked to the person's trajectories in relation to the other person, the artefact, and the locale she/he belongs to. Perhaps the system could use logged trajectory information to automatically adjust people, artefacts, locales along the centre-periphery continuum for a user's corresponding individual views. For example, future systems may adjust the presentation level of people, artefacts, and places according to how often/recently the user interacts with them.

Centre-periphery of locales and light-weight locale withdrawal. If locales can be mapped on the centre-periphery continuum in fine granularities, a locale on the out-most periphery can be archived without a user's explicit withdrawal from it. This has been suggested by our study participants as one way of solving the heavy-weight place withdrawal issue.

In summary, we need to reconsider how the elements in the Locales Framework are related to each other and how they can be further integrated into a more holistic design. This will be challenging, as, unless one is careful, it can trade off power with complexity in its user interface. Still, CT's design is a promising first step.

Appendix A. References

de Alwis, B., Gutwin, C., & Greenberg, S. (2009). GT/SD: performance and simplicity in a groupware toolkit. In *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems* (pp. 265-274). Pittsburgh, PA, USA: ACM. doi:10.1145/1570433.1570483

Baecker, R. M., Nastos, D., Posner, I. R., & Mawby, K. L. (1993). The user-centered iterative design of collaborative writing software (pp. 399-405). Amsterdam, The Netherlands: ACM. doi:10.1145/169059.169312

Bellini, P., Nesi, P., & Spinu, M. B. (2002). Cooperative visual manipulation of music notation. *ACM Trans. Comput.-Hum. Interact.*, *9*(3), 194-237. doi:10.1145/568513.568515

Bly, S. A., & Minneman, S. L. (1990). Commune: a shared drawing surface. *SIGOIS Bull.*, *11*(2-3), 184-192. doi:10.1145/91478.91514

Bly, S. A., Harrison, S. R., & Irwin, S. (1993). Media spaces: bringing people together in a video, audio, and computing environment. *Commun. ACM*, *36*(1), 28-46. doi:10.1145/151233.151235

Bolstad, Cuevas, Gonzalez, & Schneider. (2005). Modeling shared situation awareness. Presented at the the 14th Conference on Behavior Representation in Modeling and Simulation (BRIMS), Los Angels, CA, USA.

Borning, A., & Travers, M. (1991). Two approaches to casual interaction over computer and video networks (pp. 13-19). New Orleans, Louisiana, United States: ACM. doi:10.1145/108844.108847

Boyle, M., & Greenberg, S. (2005). Rapidly Prototyping Multimedia Groupwar. Presented at the 11th International Conference on Distributed Multimedia Systems, Banff, Canada: Knowledge Systems Institute, Illinois, USA.

Buxton, B. (2007). *Sketching User Experiences: Getting the Design Right and the Right*

Design. Morgan Kaufmann.

Cameron, A. F., & Webster, J. (2005). Unintended consequences of emerging communication technologies: Instant Messaging in the workplace. *Computers in Human Behavior*, *21*(1), 85-103. doi:10.1016/j.chb.2003.12.001

Chang, K. H., Gong, Y., Dollar, T., Gajiwala, S., Lee, B. G., & Wear, A. W. (1995). On computer supported collaborative writing tools for distributed environments (pp. 222-229). Nashville, Tennessee, United States: ACM. doi:10.1145/259526.259559

Curtis, P., Dixon, M., Frederick, R., & Nichols, D. A. (1995). The Jupiter audio/video architecture: secure multimedia in network places (pp. 79-90). San Francisco, California, United States: ACM. doi:10.1145/217279.215128

Dourish, P., & Bellotti, V. (1992). Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work* (pp. 107-114). Toronto, Ontario, Canada: ACM. doi:10.1145/143457.143468

Dourish, P., & Bly, S. (1992). Portholes: supporting awareness in a distributed work group (pp. 541-547). Monterey, California, United States: ACM. doi:10.1145/142750.142982

Eisenberg, R., & Bennage, C. (2008). *Sams Teach Yourself WPF in 24 Hours* (1st ed.). Sams.

Endsley, M. R. (1995). Measurement of Situation Awareness in Dynamic Systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *37*, 65-84. doi:10.1518/001872095779049499

Fahlén, L. E., Brown, C. G., Ståhl, O., & Carlsson, C. (1993). A space based model for user interaction in shared synthetic environments (pp. 43-48). Amsterdam, The Netherlands: ACM. doi:10.1145/169059.169068

Fish, R. S., Kraut, R. E., & Chalfonte, B. L. (1990). The VideoWindow system in informal communication (pp. 1-11). Los Angeles, California, United States: ACM. doi:10.1145/99332.99335

Fish, R. S., Kraut, R. E., & Leland, M. D. P. (1988). Quilt: a collaborative tool for cooperative writing (pp. 30-37). Palo Alto, California, United States: ACM. doi:10.1145/45410.45414

Fitzpatrick, G. (2003). *The locales framework: understanding and designing for wicked problems*. Springer.

Frécon, E., & Nöu, A. A. (1998). Building distributed virtual environments to support collaborative work (pp. 105-113). Taipei, Taiwan: ACM. doi:10.1145/293701.293715

Fussell, S. R., Kiesler, S., Setlock, L. D., & Scupelli, P. (2004). Effects of instant messaging on the management of multiple project trajectories (pp. 191-198). Vienna, Austria: ACM. doi:10.1145/985692.985717

Gaver, W., Moran, T., MacLean, A., Lövstrand, L., Dourish, P., Carter, K., & Buxton, W. (1992). Realizing a video environment: EuroPARC's RAVE system (pp. 27-35). Monterey, California, United States: ACM. doi:10.1145/142750.142754

Gaver, W. W., Smets, G., & Overbeeke, K. (1995). A Virtual Window on media space (pp. 257-264). Denver, Colorado, United States: ACM Press/Addison-Wesley Publishing Co. doi:10.1145/223904.223937

Greenberg, S., & Buxton, B. (2008). Usability evaluation considered harmful (some of the time). In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems* (pp. 111-120). Florence, Italy: ACM. doi:10.1145/1357054.1357074

Greenberg, Saul, Fitzpatrick, G., Gutwin, C., & Kaplan, S. (2000). Adapting the Locales Framework for Heuristic Evaluation of Groupware. *Australian Journal of Information Systems (AJIS)*, 102-108.

Greenberg, S., & Roseman, M. (1999). Groupware Toolkits for Synchronous Work. *Computer-Supported Cooperative Work (Trends in Software 7)*. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.34.4505

Greenberg, S., Voida, S., Stehr, N., & Tee, K. (2010). Artifacts as Instant Messaging Buddies. In *Hawaii International Conference on System Sciences* (Vol. 0, pp. 1-10). Los Alamitos, CA, USA: IEEE Computer Society. doi:http://doi.ieeecomputersociety.org/10.1109/HICSS.2010.70

Greenhalgh, C., & Benford, S. (1995). MASSIVE: a collaborative virtual environment for teleconferencing. *ACM Trans. Comput.-Hum. Interact.*, *2*(3), 239-261. doi:10.1145/210079.210088

Greif, I., & Sarin, S. (1986). Data sharing in group work (pp. 175-183). Austin, Texas: ACM. doi:10.1145/637069.637092

Grinter, R. E., & Palen, L. (2002). Instant messaging in teen life (pp. 21-30). New Orleans, Louisiana, USA: ACM. doi:10.1145/587078.587082

Gutwin, C., & Greenberg, S. (2002). A Descriptive Framework of Workspace Awareness for Real-Time Groupware. *Comput. Supported Coop. Work*, *11*(3), 411-446.

Haake, J. M., & Wilson, B. (1992). Supporting collaborative writing of hyperdocuments in SEPIA (pp. 138-146). Toronto, Ontario, Canada: ACM. doi:10.1145/143457.143472

Herbsleb, J. D., Atkins, D. L., Boyer, D. G., Handel, M., & Finholt, T. A. (2002). Introducing instant messaging and chat in the workplace (pp. 171-178). Minneapolis, Minnesota, USA: ACM. doi:10.1145/503376.503408

Isaacs, E., Walendowski, A., Whittaker, S., Schiano, D. J., & Kamm, C. (2002). The character, functions, and styles of instant messaging in the workplace (pp. 11-20). New Orleans, Louisiana, USA: ACM. doi:10.1145/587078.587081

Ishii, H. (1990). TeamWorkStation: towards a seamless shared workspace (pp. 13-26). Los Angeles, California, United States: ACM. doi:10.1145/99332.99337

Ishii, H., & Kobayashi, M. (1992). ClearBoard: a seamless medium for shared drawing and conversation with eye contact (pp. 525-532). Monterey, California, United States: ACM. doi:10.1145/142750.142977

Ishii, H., Kobayashi, M., & Grudin, J. (1993). Integration of interpersonal space and shared workspace: ClearBoard design and experiments. *ACM Trans. Inf. Syst.*, *11*(4), 349-375. doi:10.1145/159764.159762

Ishii, H., & Miyake, N. (1991). Toward an open shared workspace: computer and video fusion approach of TeamWorkStation. *Commun. ACM*, *34*(12), 37-50. doi:10.1145/125319.125321

Jancke, G., Venolia, G. D., Grudin, J., Cadiz, J. J., & Gupta, A. (2001). Linking public spaces: technical and social issues (pp. 530-537). Seattle, Washington, United States: ACM. doi:10.1145/365024.365352

Kraut, R., Egido, C., & Galegher, J. (1988). Patterns of contact and communication in scientific research collaboration. In *Proceedings of the 1988 ACM conference on Computer-supported cooperative work* (pp. 1-12). Portland, Oregon, United States: ACM. doi:10.1145/62266.62267

Kuzuoka, H., Kosuge, T., & Tanaka, M. (1994). GestureCam: a video communication system for sympathetic remote collaboration (pp. 35-43). Chapel Hill, North Carolina, United

States: ACM. doi:10.1145/192844.192866

Leland, M. D. P., Fish, R. S., & Kraut, R. E. (1988). Collaborative document production using quilt (pp. 206-215). Portland, Oregon, United States: ACM. doi:10.1145/62266.62282

Mansfield, T., Kaplan, S., Fitzpatrick, G., Phelps, T., Fitzpatrick, M., & Taylor, R. (1997). Evolving Orbit: a process report on building locales. In *Proceedings of the international ACM SIGGROUP conference on Supporting group work: the integration challenge* (pp. 241-250). Phoenix, Arizona, United States: ACM. doi:10.1145/266838.266919

McEwan, G. (2006) Community Bar: Designing for Informal Awareness and Casual Int. (n.d.). .

McEwan, G. (2006, September). *Community Bar: Designing for Informal Awareness and Casual Interaction* (Master's thesis). Calgary, Alberta, Canada: Department of Computer Science, University of Calgary. Retrieved from http://grouplab.cpsc.ucalgary.ca/Publications/2006-McEwan.MScThesis

McEwan, G., & Greenberg, S. (2005). Supporting social worlds with the community bar. In *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work* (pp. 21-30). Sanibel Island, Florida, USA: ACM. doi:10.1145/1099203.1099207

McEwan, G., Greenberg, S., Rounding, M., & Boyle, M. (2006). Groupware Plug-ins: A Case Study of Extending Collaboration Functionality through Media Items. In *Proceedings of 2nd International Conference on Collaboration Technologies* (pp. 42-47). Presented at the CollabTech 2006, Tsukuba, Japan: IPSJ SIG Groupware and Network Services.

Morán, A. L., Favela, J., Enríquez, A. M. M., & Decouchant, D. (2001). Document Presence Notification Services for Collaborative Writing. In *Proceedings of the Seventh International Workshop on Groupware* (pp. 125-135). IEEE Computer Society. Retrieved from http://portal.acm.org/citation.cfm?id=646132.680144

Muller, M. J. (1991). PICTIVE\&mdash;an exploration in participatory design. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology* (pp. 225-231). New Orleans, Louisiana, United States:

ACM. doi:10.1145/108844.108896

Nakanishi, H., Murakami, Y., & Kato, K. (2009). Movable cameras enhance social telepresence in media spaces (pp. 433-442). Boston, MA, USA: ACM. doi:10.1145/1518701.1518771

Nakanishi, H., Yoshida, C., Nishimura, T., & Ishida, T. (1996). FreeWalk: supporting casual meetings in a network (pp. 308-314). Boston, Massachusetts, United States: ACM. doi:10.1145/240080.240314

Nardi, B. A., Whittaker, S., & Bradner, E. (2000). Interaction and outeraction: instant messaging in action (pp. 79-88). Philadelphia, Pennsylvania, United States: ACM. doi:10.1145/358916.358975

Neuwirth, C. M., Kaufer, D. S., Chandhok, R., & Morris, J. H. (1990). Issues in the design of computer support for co-authoring and commenting (pp. 183-195). Los Angeles, California, United States: ACM. doi:10.1145/99332.99354

Newman-Wolfe, R. E., & Pelimuhandiram, H. K. (1991). MACE: a fine grained concurrent editor. *SIGOIS Bull.*, *12*(2-3), 240-254. doi:10.1145/127769.122855

Nielsen, J. (1993). *Usability Engineering* (1st ed.). Morgan Kaufmann.

Okada, K., Maeda, F., Ichikawaa, Y., & Matsushita, Y. (1994). Multiparty videoconferencing at virtual social distance: MAJIC design (pp. 385-393). Chapel Hill, North Carolina, United States: ACM. doi:10.1145/192844.193054

Patterson, D. J., Baker, C., Ding, X., Kaufman, S. J., Liu, K., & Zaldivar, A. (2008). Online everywhere: evolving mobile instant messaging practices (pp. 64-73). Seoul, Korea: ACM. doi:10.1145/1409635.1409645

Quan-Haase, A. (2008). Instant Messaging on Campus: Use and Integration in University Students' Everyday Communication. *The Information Society*, *24*(2), 105-115.

Ranjan, A., Birnholtz, J. P., & Balakrishnan, R. (2006). An exploratory analysis of partner action and camera control in a video-mediated collaborative task (pp. 403-412). Banff, Alberta, Canada: ACM. doi:10.1145/1180875.1180936

Ranjan, A., Birnholtz, J. P., & Balakrishnan, R. (2007). Dynamic shared visual spaces: Experimenting with automatic camera control in a remote repair task. In Conference on Human Factors in Computing Systems - Proceedings (pp. 1177-1186). San Jose, CA, United states: Association for Computing Machinery. Retrieved from http://dx.doi.org/10.1145/1240624.1240802

Romero, N., McEwan, G., & Greenberg, S. (2007). A field study of community bar: (mis)-matches between theory and practice (pp. 89-98). Sanibel Island, Florida, USA: ACM. doi:10.1145/1316624.1316638

Root, R. W. (1988). Design of a multi-media vehicle for social browsing (pp. 25-38). Portland, Oregon, United States: ACM. doi:10.1145/62266.62269

Rounding, M. (2004, April). *Informal Awareness and Casual Interaction with the Notification Collage* (Master's thesis). Calgary, Alberta, Canada: University of Calgary.

Scupelli, P., Fussell, S. R., Kiesler, S., Quinones, P., & Kusbit, G. (2007). Juggling Work Among Multiple Projects and Partner. In *Hawaii International Conference on System Sciences* (Vol. 0, p. 77a). Los Alamitos, CA, USA: IEEE Computer Society. doi:http://doi.ieeecomputersociety.org/10.1109/HICSS.2007.310

Tang, J. C., & Minneman, S. (1991). VideoWhiteboard: video shadows to support remote collaboration (pp. 315-322). New Orleans, Louisiana, United States: ACM. doi:10.1145/108844.108932

Tang, J. C., & Minneman, S. L. (1991). Videodraw: a video interface for collaborative drawing. *ACM Trans. Inf. Syst.*, *9*(2), 170-184. doi:10.1145/123078.128729

Tang, J. C., & Rua, M. (1994). Montage: providing teleproximity for distributed groups (pp. 37-43). Boston, Massachusetts, United States: ACM. doi:10.1145/191666.191688

Tee, K. (2007). *Artifact Awareness for Distributed Groups through Screen Sharing* (Master's thesis). Calgary, Alberta, Canada: Department of Computer Science, University of Calgary.

Tee, K., Greenberg, S., & Gutwin, C. (2009). Artifact awareness through screen sharing for distributed groups. *International Journal of Human-Computer Studies*, *67*(9), 677-702. doi:10.1016/j.ijhcs.2009.04.001

Vos, H. D., ter Hofte, H. T., & de Poot, H. (2004). IM [@Work] Adoption of Instant Messaging in a Knowledge Worker Organisation (p. 10019.1). IEEE Computer Society. Retrieved from http://portal.acm.org/citation.cfm?id=962749.962797&coll=GUIDE&dl=GUIDE&CFID=58132218&CFTOKEN=68061747

Watabe, K., Sakata, S., Maeno, K., Fukuoka, H., & Ohmori, T. (1990). Distributed multiparty desktop conferencing system: MERMAID (pp. 27-38). Los Angeles, California, United States: ACM. doi:10.1145/99332.99338

Whittaker, S., Frohlich, D., & Daly-Jones, O. (1994). Informal workplace communication: what is it like and how might we support it? In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence* (pp. 131-137). Boston, Massachusetts, United States: ACM. doi:10.1145/191666.191726

Appendix B. Media items as plug-ins

As mentioned in Chapter 6, all media item types are DLL (dynamically linked library) files, compiled separately as plugins, and then integrated into the Come Together executable client at runtime. This architecture makes Come Together an open system, where third party developers can add new group interaction capabilities.

When launched, the main program searches for all plug-in DLLs and registers the .Net reflection object for each media item type, an implementation of IPersonItem or IArtefactItem. The main program uses the class reflection object of a media item type to create an instance.

These plug-in DLLs communicate with the main client program (Figure B-1) through a bi-directional contract. In essence, the main program needs to communicate with the view layer of the media item DLLs, while the DLLs need to communicate with the presenter layer of the main program. First, when the main CT client loads a media item DLL plug-in, it provides the DLL with access to the shared dictionary (as represented by an object) and domain logic objects representing people and artefacts (IPerson and IArtefact interfaces, explained shortly). Second, the main CT client needs to retrieve from the DLL the three levels of presentation, which it does via a unique protocol (a common programming interface IItem, explained shortly). Figure 6-5 provides a concrete illustration of the bi-directional communication between the CT client proper and its media item DLLs.

The contract itself is implemented by the DLL, where it must include three program interfaces—IItem (Listing B-1), IPersonItem (Listing B-2), IArtefactItem (Listing B-3). The main client then retrieves the three views (i.e., its view layer) of a media item via these interfaces and two interfaces—IPerson (Listing B-4), IArtefact (Listing B-5)—for a media item to access the domain logic (presenter of a person or an artefact) of the Come Together client proper.

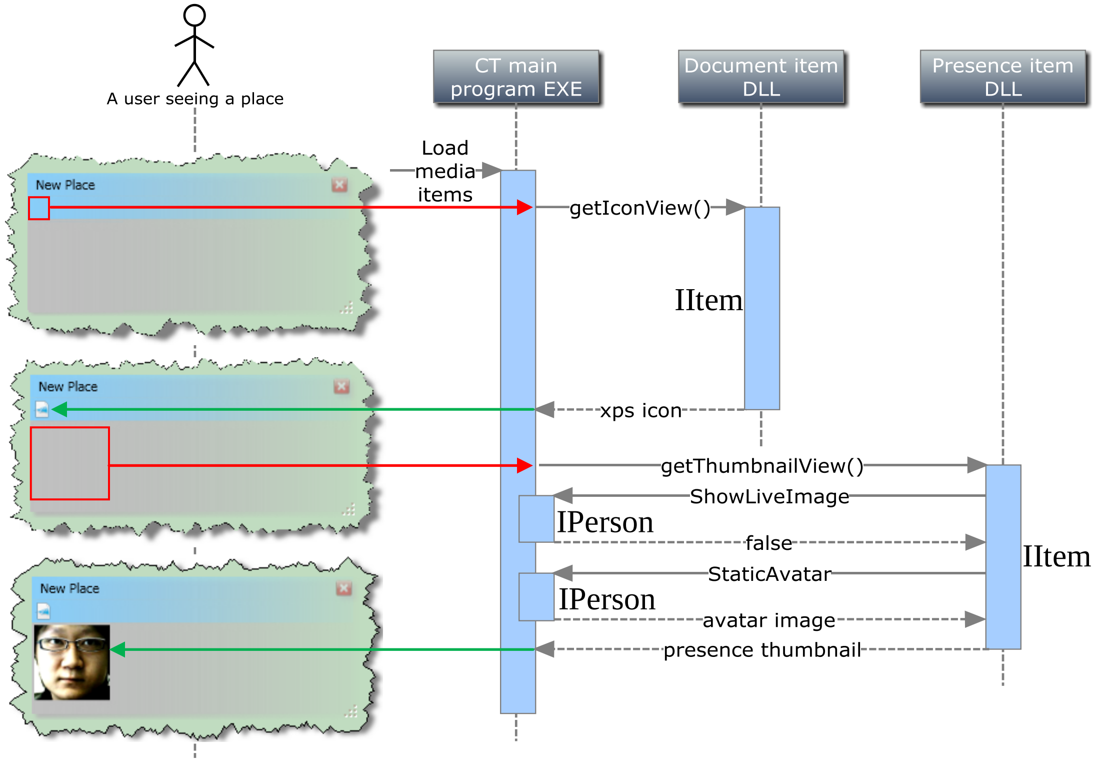


Figure B-1 Sequence Diagram: CT plug-in DLLs communications

```
/*The interface must be implemented to provide the three
views     of a media item as WPF user controls.
*/
public interface IItem : ICloneable
{
    UserControl GetIconView();
    UserControl GetThumbnailView();
    UserControl GetFullView();
}
```

Listing B-1 The IItem interface

```
public interface IPersonItem: IItem
{
    //To initialize a media item of a person
    void StartItem(SharedDictionary sd, string path,
                    IPerson personHelper);
    //To  release  resources  when  the  media  item  is
destructed
    void StopItem();
}
```

Listing B-2 The IPersonItem interface

```
public interface IArtefactItem: IItem
{
    //To initialize an artefact media item
    void StartItem(SharedDictionary sd, string path,
                    object obj, IArtifact artifactHelper,
                    IPerson personHelper);
    //To handle dropped object(s)
    void DropData(IDataObject data);
    //To  release  resources  when  the  media  item  is
destructed
    void StopItem();
}
```

Listing B-3 The IArtefactItem interface

```
/*The presenter interface for a person in the main
program
*/
public interface IPerson
{
    string DisplayName { get; }
    string PersonGuid { get; }
    string Email { get; }
    Image StaticAvatar { get; }
    Image LiveImage { get; }
    bool IsOnline { get; }
    bool ShowLiveImage { get; }

    String Path { get; }
    SharedDictionary sd { get; }
}
```

Listing B-4 The IPerson interface

```
/*The presenter interface for an artifact in the main
program
*/
public interface IArtefact
{
    string Name { get; set; }
    string ArtefactGuid { get; set; }
    string OwnerGuid { get; set; }

    event EventHandler ArtefactUpdated;
    void SignalArtefactUpdate();

    String Path { get; }
    SharedDictionary sd { get; }
}
```

Listing B-5 The IArtefact interface

Both IPersonItem and IArtefactItem extend the IItem interface. As a result, all media items, in fact, implement IItem indirectly. IItem (Listing B-1) simply has three methods used by the main client program to retrieve the three views (as WPF User Controls) of a media item whenever they need to be displayed.

Come Together is designed to treat people and artefacts equally, but the underlying implementations of media items for people and artefacts are different. Media items for people must implement the IPersonItem interface (Listing B-2) while those for artefacts must implement the IArtefactItem interface (Listing B-3). Media items for people also behave slightly different from those for artefacts, where it updates its content if a drag-and-droppable object of a matching type is dropped onto it. Therefore, IArtefactItem (Listing B-3) contains a DropData method, called by the main client, which allows the media item to handle a dropped object.

Once created, a media item's StartItem method, in either IPersonItem (Listing B-2) or IArtefactItem (Listing B-3), is called for the main program to pass important arguments. The difference of StartItem methods in IPersonItem and IArtefactItem include:

- IPersonItem requires an IPerson object passed into the media item.
- IArtefactItem requires an IArtefact object and an IPerson object passed in the media item because some artefact media item types need to know who the current user is.
- IArtefactItem's StartItem (Listing B-3) has one more parameter: a dropped object which is passed in when a media item is created with a drag-and-drop.

IPersonItem and IArtefactItem serve one way of the contract for the main program to retrieve the three views of a media item DLL. IPerson and IArtefact serve the other way of the contract for a media item DLL to retrieve domain data of a person or an artefact. The singleton shared dictionary instance is still passed from the main program to media items. A path, /people/aPersonGUID (string path in StartItem, Listing B-2) or /artefact/anArtefactGUID (string path in StartItem, Listing B-2), in the shared dictionary is also passed in for the media item to store extra, type-specific, custom data about a person or an artefact in the information hierarchy.

Appendix C. Design Process

The design of Come Together is an exploratory process for a possible groupware solution to support light-weight group working practices, during which I gained general interaction design experience via four major phases: direct coding of the first design's prototype, iterative sketching of the multiple subsequent designs, defining and refining the final design using a variation of the Pictive method (Muller, 1991), and coding and refining the final software prototype, Come Together.

I began with a bottom-up approach, where I started coding the software. This initial design was prototyped with considerable details implemented. I developed a complex tabular window system (Figure C-1) to integrate people, artefacts, and places. The system could have been deployed as a real distributed groupware with actual networking capability. I coded different versions of the prototype with minor variations. The process was time-consuming and expensive. Still, it was a valuable experience where I learnt about the basic technology I was using, and uncovered various issues in my design. I also learnt that I had mistakenly misbelieved that I knew the correct design at the beginning; as Buxton (2007) states, coding prototypes should be the last thing to do in the process of designing user interfaces.

Next, I started a few iterations of sketching different user interface designs as storyboards (interaction sequences) on paper (Buxton, 2007). Since sketches are very cheap and totally disposable, I explored a number of distinct and quite different designs in a shorter period of time. For example, one issue concerned layout: desktop widgets vs. windowed designs; places as containers vs. container-less designs; implicit place formation vs. explicit place formation designs. Using storyboarding, I was able to develop and then communicate my ideas effectively enough to my supervisor and others in a more efficient way. This process also inspired the theoretical part of my thesis: analysis of the Locales Framework and existing systems for the groupware design considerations to support lightweight group working practices.

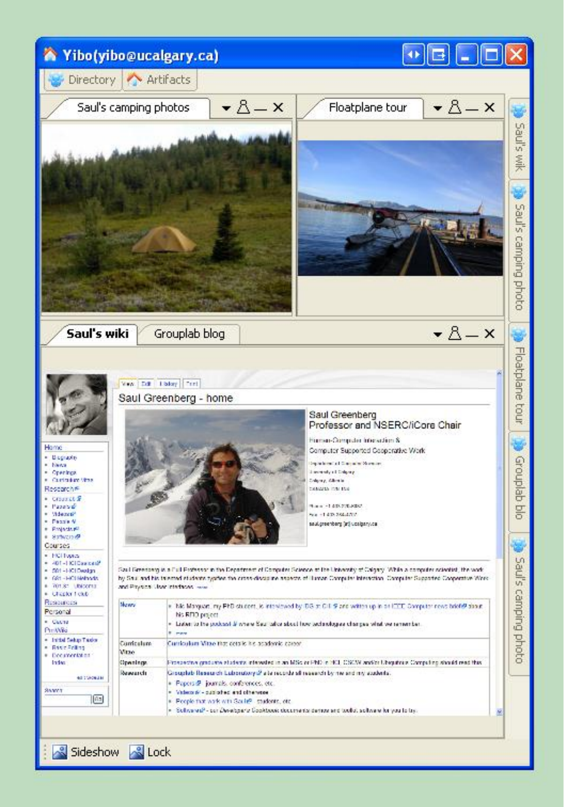


Figure C-1 Early Prototype

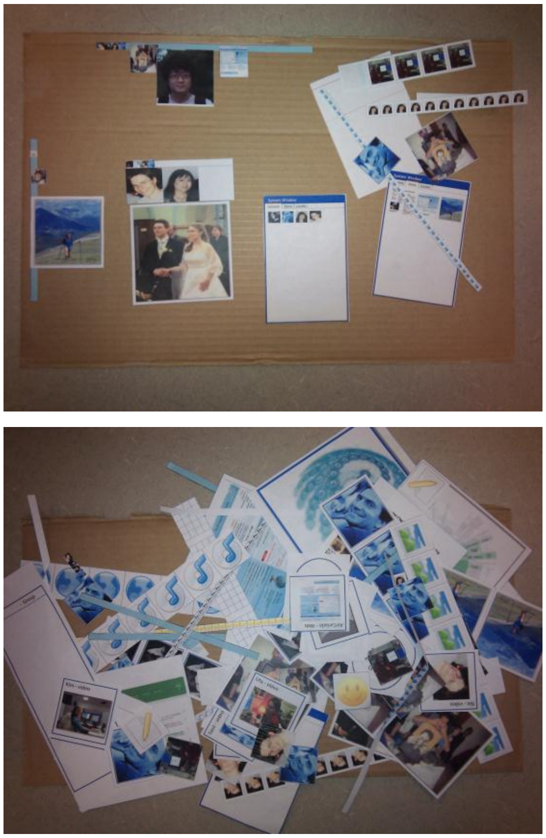
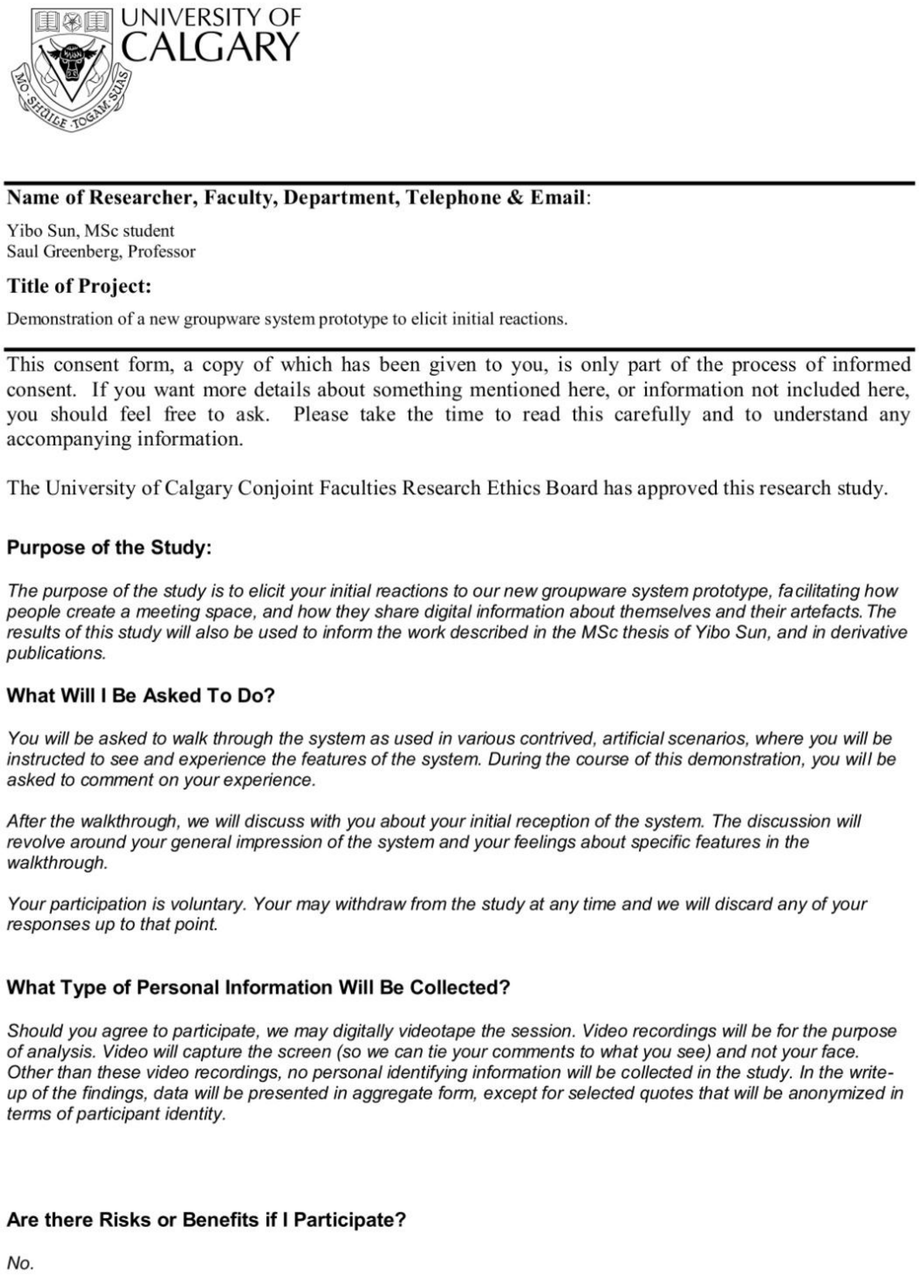


Figure C-2 paper board prototyping

Then, I narrowed down onto one particular design using desktop floating widgets and docking stations on the periphery of the screen. To quickly experience alternative designs with minor changes, I used a variation of the Pictive process (Muller, 1991), where I created different sized paper-based media items, and then laid them out on cardboard (the 'screen') to examine different arrangements of places and other aspects (Figure C-2). My approach differed from Pictive in that Pictive was meant to encourage non-developers to get involved with the design, whereas I used this method primarily for my own development of the design and to communicate and get feedback of my design from other HCI experts.

Subsequently, I developed the software prototype of Come Together. The development underwent an incremental testing process, e.g., media items' presentation levels, and placement within places and the CT Console. The layout of content and various controls under each tab in the CT Console also went through a few iterations. I continually made design decisions based on trade-offs between consistency, simplicity, and power. Eventually, I tested the prototype with the preliminary study as described in Chapter 7, and uncovered the matches and mismatches between the final design and the goals I had originally set out to acheive.

Appendix D. Consent Form



UNIVERSITY OF CALGARY

Name of Researcher, Faculty, Department, Telephone & Email:

Yibo Sun, MSc student
Saul Greenberg, Professor

Title of Project:

Demonstration of a new groupware system prototype to elicit initial reactions.

This consent form, a copy of which has been given to you, is only part of the process of informed consent. If you want more details about something mentioned here, or information not included here, you should feel free to ask. Please take the time to read this carefully and to understand any accompanying information.

The University of Calgary Conjoint Faculties Research Ethics Board has approved this research study.

Purpose of the Study:

The purpose of the study is to elicit your initial reactions to our new groupware system prototype, facilitating how people create a meeting space, and how they share digital information about themselves and their artefacts. The results of this study will also be used to inform the work described in the MSc thesis of Yibo Sun, and in derivative publications.

What Will I Be Asked To Do?

You will be asked to walk through the system as used in various contrived, artificial scenarios, where you will be instructed to see and experience the features of the system. During the course of this demonstration, you will be asked to comment on your experience.

After the walkthrough, we will discuss with you about your initial reception of the system. The discussion will revolve around your general impression of the system and your feelings about specific features in the walkthrough.

Your participation is voluntary. Your may withdraw from the study at any time and we will discard any of your responses up to that point.

What Type of Personal Information Will Be Collected?

Should you agree to participate, we may digitally videotape the session. Video recordings will be for the purpose of analysis. Video will capture the screen (so we can tie your comments to what you see) and not your face. Other than these video recordings, no personal identifying information will be collected in the study. In the write-up of the findings, data will be presented in aggregate form, except for selected quotes that will be anonymized in terms of participant identity.

Are there Risks or Benefits if I Participate?

No.

What Happens to the Information I Provide?

Only the researchers will have access to the notes taken and video-records from either the demonstration or the discussion. The researchers may publish your reactions and comments about the system without revealing your identity.

You are free to withdraw from the study at any time and the researchers will discard any of your responses you have provided up to the point.

Data will be locked in office cabinets in paper form or stored on secure computer systems in digital form. All data will be kept for a maximum of one year, after which time it will be permanently deleted.

Signatures (written consent)

Your signature on this form indicates that you 1) understand to your satisfaction the information provided to you about your participation in this research project, and 2) agree to participate as a research subject.

In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. You are free to withdraw from this research project at any time. You should feel free to ask for clarification or new information throughout your participation.

Participant's Name: (please print) ____

Participant's Signature ____Date: _______________

Researcher's Name: (please print) ____

Researcher's Signature: _________________________________Date: ________________

Questions/Concerns

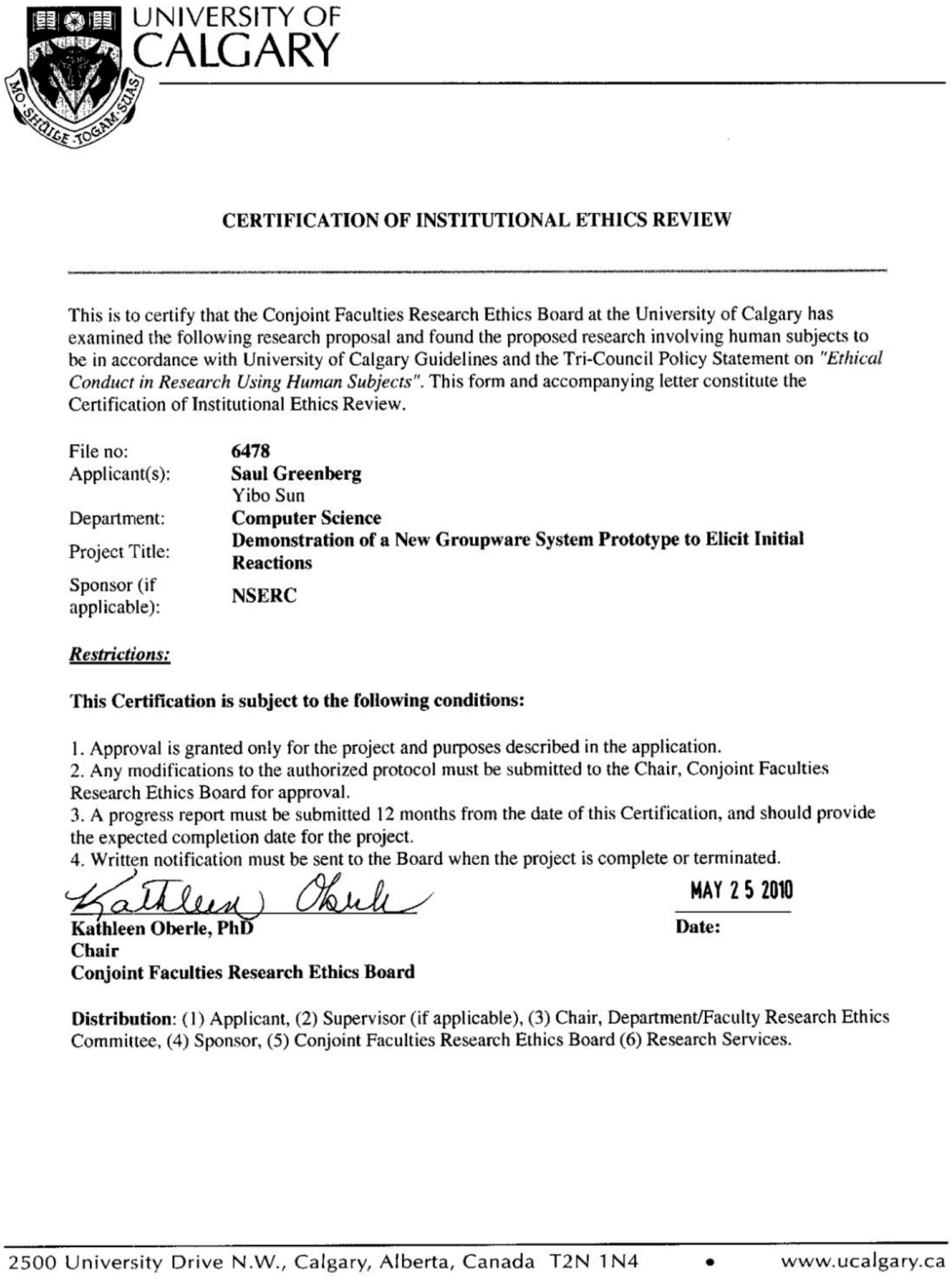
If you have any further questions or want clarification regarding this research and/or your participation, please contact:

Mr. Yibo Sun, Department of Computer Science, (403)210-9499, yibo.sun@ucalgary.ca and/or
Dr. Saul Greenberg, Department of Computer Science, (403)210-9499, saul.greenberg@ucalgary.ca

If you have any concerns about the way you've been treated as a participant, please contact the Senior Ethics Resource Officer, Research Services Office, University of Calgary at (403) 220-3782; email rburrows@ucalgary.ca.

A copy of this consent form has been given to you to keep for your records and reference. The investigator has kept a copy of the consent form.

Appendix E. Ethics Approval



UNIVERSITY OF
CALGARY

CERTIFICATION OF INSTITUTIONAL ETHICS REVIEW

This is to certify that the Conjoint Faculties Research Ethics Board at the University of Calgary has examined the following research proposal and found the proposed research involving human subjects to be in accordance with University of Calgary Guidelines and the Tri-Council Policy Statement on *"Ethical Conduct in Research Using Human Subjects"*. This form and accompanying letter constitute the Certification of Institutional Ethics Review.

File no:	**6478**
Applicant(s):	**Saul Greenberg** Yibo Sun
Department:	**Computer Science**
Project Title:	**Demonstration of a New Groupware System Prototype to Elicit Initial Reactions**
Sponsor (if applicable):	**NSERC**

Restrictions:

This Certification is subject to the following conditions:

1. Approval is granted only for the project and purposes described in the application.
2. Any modifications to the authorized protocol must be submitted to the Chair, Conjoint Faculties Research Ethics Board for approval.
3. A progress report must be submitted 12 months from the date of this Certification, and should provide the expected completion date for the project.
4. Written notification must be sent to the Board when the project is complete or terminated.

Kathleen Oberle

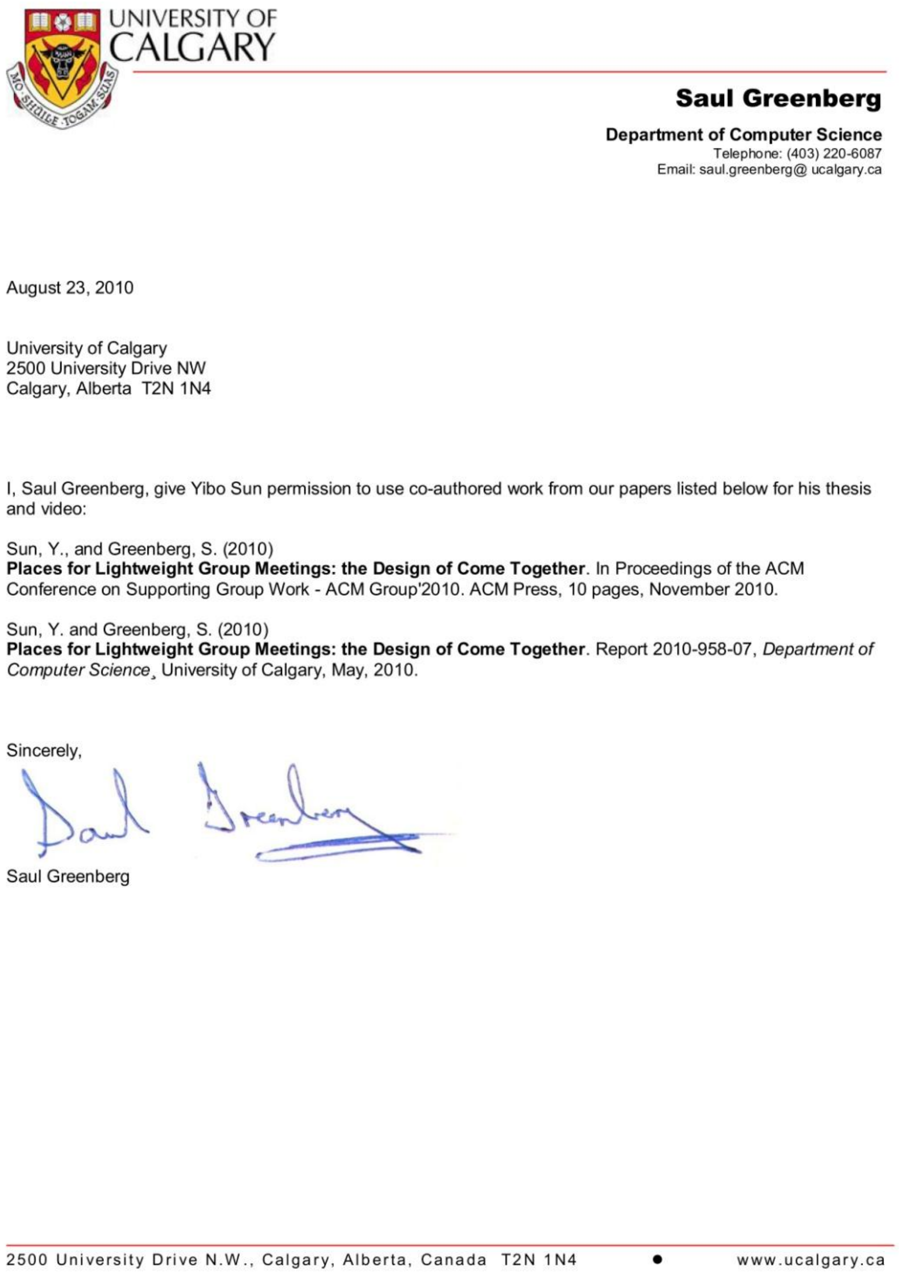
Kathleen Oberle, PhD
Chair
Conjoint Faculties Research Ethics Board

MAY 2 5 2010
Date:

Distribution: (1) Applicant, (2) Supervisor (if applicable), (3) Chair, Department/Faculty Research Ethics Committee, (4) Sponsor, (5) Conjoint Faculties Research Ethics Board (6) Research Services.

2500 University Drive N.W., Calgary, Alberta, Canada T2N 1N4 • www.ucalgary.ca

Appendix F. Co-Author Permission



UNIVERSITY OF CALGARY

Saul Greenberg

Department of Computer Science
Telephone: (403) 220-6087
Email: saul.greenberg@ ucalgary.ca

August 23, 2010

University of Calgary
2500 University Drive NW
Calgary, Alberta T2N 1N4

I, Saul Greenberg, give Yibo Sun permission to use co-authored work from our papers listed below for his thesis and video:

Sun, Y., and Greenberg, S. (2010)
Places for Lightweight Group Meetings: the Design of Come Together. In Proceedings of the ACM Conference on Supporting Group Work - ACM Group'2010. ACM Press, 10 pages, November 2010.

Sun, Y. and Greenberg, S. (2010)
Places for Lightweight Group Meetings: the Design of Come Together. Report 2010-958-07, *Department of Computer Science*¸ University of Calgary, May, 2010.

Sincerely,

Saul Greenberg

Saul Greenberg

2500 University Drive N.W., Calgary, Alberta, Canada T2N 1N4 ● www.ucalgary.ca