

Heuristic Evaluation of Groupware Based on the Mechanics of Collaboration

Kevin Baker¹, Saul Greenberg¹, and Carl Gutwin²

¹ Department of Computer Science
University of Calgary
Calgary, Alberta, Canada T2N 1N4
{bakerkev, saul}@cpsc.ucalgary.ca

² Department of Computer Science
University of Saskatchewan
Saskatoon, Saskatchewan, Canada S7N 5A9
gutwin@cs.usask.ca

Abstract. Despite the increasing availability of groupware, most systems are awkward and not widely used. While there are many reasons for this, a significant problem is that groupware is difficult to evaluate. In particular, there are no discount usability evaluation methodologies that can discover problems specific to teamwork. In this paper, we describe how we adapted Nielsen's heuristic evaluation methodology, designed originally for single user applications, to help inspectors rapidly, cheaply effectively identify usability problems within groupware systems. Specifically, we take the 'mechanics of collaboration' framework and restate it as heuristics for the purposes of discovering problems in shared visual work surfaces for distance-separated groups.

1 Introduction

Commercial real-time distributed groupware is now readily available due to improvements in hardware, increased connectivity of the Internet, and demands of increasingly distributed organizations. Yet with the exception of a few systems, groupware is not widely used. One main reason is that these systems have serious usability problems, a situation caused in part by the lack of practical methodologies for evaluating them. As Grudin points out, most groupware systems are complex and introduce almost insurmountable obstacles to meaningful, generalizable and inexpensive evaluation [14]. Without evaluation methods, groupware developers cannot learn from experience. Consequently, even today's collaborative systems contain usability problems that make them awkward to use.

Even though Grudin made this point over a decade ago, we have not yet developed techniques to make groupware evaluation cost-effective within typical software project constraints. One way to address this dearth is to adapt evaluation techniques developed for single-user software usability. Within the field of human computer interaction (HCI), many low-cost evaluation techniques have moved out of the research arena and into accepted practice. Collectively, these become a toolkit. Each

methodology highlights different usability issues and identifies different types of problems; therefore, evaluators can choose and mix appropriate techniques to fit the situation [23].

While computer supported cooperative work (CSCW) is related to the field of HCI, the standard HCI methodologies have problems when we try to apply them verbatim for the purposes of evaluating groupware, as summarized below.

User observation has an evaluator observe a single person use a system to perform tasks within a semi-controlled setting. When applied to groupware, the experimenter now observes how groups use the system. The problem is that observing groups requires many subjects: the logistics of finding them and scheduling these sessions can be difficult. Groups also vary greatly in composition and in how its members interact, which makes observations difficult to analyze and generalize when compared to single user observations. Observation sessions tend to be longer, as group work requires participants to develop a rapport and interaction style. User observations can also be too sterile for testing groupware as it does not account for social, organizational, political and motivational factors that influence how the group accepts and uses the system.

Field studies have evaluators study people interacting within their work-a-day world. These can provide the context missing from observational techniques. However, they are complex and expensive in terms of time, logistics and analysis. Evaluators require experience and a considerable amount of time to conduct the evaluation. The number of people that must be observed at each site is high, which can make this task overwhelming. Field studies work best at the beginning and end of a design and are not well suited for iterative design.

Inspection methods have evaluators ‘inspect’ an interface for usability bugs according to a set of criteria, usually related to how individuals see and perform a task. As in single-user applications, groupware must effectively support task work. However, groupware must also support *teamwork*, the work of working together. Inspection methods are thus limited when we to use them ‘as-is’, for they do not address the teamwork necessary for groupware assessment. For example, Nielsen lists many heuristics to guide inspectors, yet none address ‘bugs’ particular to groupware usability. Similarly, a cognitive walkthrough used to evaluate groupware gave mixed and somewhat inconclusive results [8]; other researchers are providing a framework for typical groupware scenarios that can form a stronger basis for walkthroughs [4].

In this paper, we adapt Nielsen’s popular heuristic evaluation method to groupware assessment. Given heuristics specific to teamwork, we believe this method can be applied to groupware. In particular, we offer a set of heuristics designed to identify usability problems specific to teamwork between distance-separated groups working over a shared visual work surface.

We begin with a brief overview of Nielsen’s heuristic evaluation method. We then introduce a previously defined framework called the ‘mechanics of collaboration’ and use it to derive eight groupware heuristics that we explain in detail. We close by discussing our initial experiences with applying the heuristics and the steps required to validate them as a discount usability methodology.

2 Heuristic Evaluation

Heuristic evaluation (HE) is a widely accepted discount evaluation method for diagnosing potential usability problems in user interfaces [22,25,26,27,28,29]. This methodology involves a small set of usability experts visually inspecting an interface and judging its compliance with recognized usability principles (the “heuristics”) e.g., ‘Provide Feedback’ or ‘Use the User’s Language’ [22,25,26,27,28]. Heuristics are general rules used to describe common properties of usable interfaces [27]. They help evaluators focus their attention on aspects of an interface that are often sources of trouble, making detection of usability problems easier.

HE is popular with both researchers and industry. It is low cost in terms of time since it can be completed in a relatively short amount of time (i.e. a few hours). End-users are also not required; therefore, resources are inexpensive. Because heuristics are well documented [e.g., 27,28], they are easy to learn and apply. Finally, HE can be used fairly successfully by non-usability experts. An aggregate of 3-5 evaluators will typically identify 75-80% of all usability problems [27]. We look to capitalize on all these factors as we expand this technique to groupware evaluation.

3 The Mechanics of Collaboration

We are adapting HE by developing new groupware-specific heuristics. We started this process in Greenberg et al. [11], where we listed five heuristics based on the Locales Framework. This set was tailored for evaluating comprehensive collaborative environments and how they co-existed with the groups’ everyday methods for communicating and collaborating. While likely good for identifying generic groupware problems, these heuristics do not include principles tailored to particular groupware application genres.

In this paper, we specialized our work by developing a more specific but still complementary set of heuristics tailored to the groupware genre of *shared visual workspaces*. These new heuristics should help inspectors evaluate how a shared visual workspace supports (or fails to support) the ability of distance-separated people to communicate and collaborate with artifacts through a visual medium. This application genre is very common e.g., real time systems for sharing views of conventional applications, group-aware versions of generalized and specialized text and graphical editors, and so on.

Specifically, we selected Gutwin’s framework for the *mechanics of collaboration* [17] as the basis for this new set of heuristics. This framework is one of the few dealing with teamwork: it was specifically created with groupware in general and shared workspaces specifically in mind. Gutwin’s mechanics were developed from an extensive analysis of shared workspace usage and theory [e.g. 1,16,38]. It describes the low level actions and interactions that small groups of people do if they are to complete a task effectively. Basic actions include communication, coordination, planning, monitoring, assistance, and protection. The underlying idea of the framework is that while some usability problems in groupware systems are strongly tied to social or organizational issues in which the system has been deployed, others

are a result of poor support for the basic activities of collaborative work in shared spaces. It is these basic activities that the framework articulates.

We believe that the framework can help inspectors identify usability problems of both groupware prototypes and existing systems. While the framework was developed with low-cost evaluation methods in mind, we had to adapt, restructure and rephrase it as heuristics, and augment it with a few other important points that are not included in the framework. Unlike single user heuristics which are somewhat independent—chosen by how well they identified ‘standard’ usability problems [28]—ours have the advantage that they are linked and interdependent as they collectively describe a partial framework of attributes of how people interact with shared visual workspaces.

The resulting eight heuristics are presented below. For each heuristic, we provide an explanation along with how groupware applications typically realize and support its criteria.

Heuristic 1: Provide the Means for Intentional and Appropriate Verbal Communication

In face-to-face settings, the prevalent form of communication in most groups is verbal conversations. The mechanism by which we gather information from verbal exchanges has been coined *intentional communication* [e.g. 1,19] and is typically used to establish a common understanding of the task at hand [1]. Gutwin summarizes three ways in which information is picked up from verbal exchanges [15].

1. People may talk explicitly about what they are doing and where they are working within a shared workspace. These direct discussions occur primarily when someone asks a specific question such as “what are you doing?” or when the group is planning or replanning the division of labour.
2. People can gather information by overhearing others’ conversations. Although a conversation between two people may not explicitly include a third person, it is understood that the exchange is public information that others can pick up.
3. People can listen to the running commentary that others tend to produce alongside their actions. This “verbal shadowing” can be explicit or highly indirect and provides additional information without requiring people to enter into a conversation. This behaviour has also been called “outlouds” [19].

Typical groupware support. Most visual workspace groupware does not support intentional verbal communications directly, as they assume that any communication channel (text, audio, video) is supplied ‘out of band’. Depending on the task, an obvious approach to facilitating verbal exchanges is to provide a digital audio link or text chat facility between participants. These can be implemented in many ways, and each has consequences on how well communication activity is supported. Text chat, for example, can be via parcel-post (e.g. type a line and send) [e.g. 41] or real-time (e.g. character by character). While limited, text can be useful for short or sporadic interactions, or where it is impractical to provide an audio connection. Lengthy or highly interactive meetings require an audio channel, typically supplied by a telephone. Digital audio is available in some systems, but currently suffers problems due to poor bandwidth, latency, and quality.

Video can also support verbal conversations. However, there are questions concerning the benefits of adding video to an audio channel. People have found it difficult to engage in conflict negotiation over distributed media since video can introduce distractors that interfere with the accuracy of interpersonal evaluations [7]. Plus, it has been observed that when conversations shifted to a meta-level, the participants turned away from the monitor showing the computerized work surface and would communicate across the video monitor [35]. While video has appeal and utility, the inspector cannot assume that inclusion of video is the panacea for all intentional communication requirements.

Heuristic 2: Provide the Means for Intentional and Appropriate Gestural Communication

Explicit gestures and other visual actions are also used alongside verbal exchanges to carry out intentional communication. For example, Tang [39] observed that gestures play a prominent role in all work surface activity for design teams collaborating over paper on tabletops and whiteboards (around 35% of all actions). These are *intentional* gestures, where people used them to directly support the conversation and convey task information. Intentional gestural communication takes many forms. *Illustration* occurs when speech is illustrated, acted out, or emphasized. For example, people often illustrate distances by showing a gap between their hands. *Emblems* occur when words are replaced by actions, such as a nod or shake of the head indicating ‘yes’ or ‘no’ [34]. *Deictic reference* or *deixis* happens when people reference objects in the workspace with a combination of intentional gestures and communication, e.g., by pointing to an object and saying “this one” [1]. Whatever the type, groupware must provide ways of conveying and supporting gestural communication by making gestures clearly visible, and by maintaining their relation with both objects within the work surface and voice communications [39].

Typical groupware support. Because people are distance-separated, gestures are invisible unless they are directly supported by the system. In groupware, this is typically done (if at all) via some form of *embodiment*. Techniques include telepointers [37], avatars, and video images.

Telepointers are the simplest means for supporting embodiment in a virtual workspace. A person’s cursor, made visible to all, allows one to gesture and point to objects in the workspace. While telepointers are limited 2D caricatures of the rich gestures people do with their hands, they are a huge improvement over nothing at all. Early systems, such as GroupSketch [12], were explicitly designed to facilitate gestural actions: each person had their own large and uniquely identifiable telepointer that they could use simultaneously with the others; telepointers were always visible within the work surface by all participants; they appeared with no apparent delay in order to remain synchronized with verbal exchanges; and they maintained their same relative location to the work surface objects across all displays.

Avatars are synthetic bodies representing the people who populate a 3D landscape. While most avatars are extremely crude, some transmit limited hand and body gestures: the idea is to capture real world gestures and have them appear in the simulated space.

Video can also recreate real bodies within groupware workspaces. Conventional video windows or monitors are not enough, for gestures are detached from workspace objects. A different approach is to mix and fuse the video of the person, their hands, and the worksurface into a single image [e.g. 20,40]. When done correctly, the final image comprises both the artifacts and the person, where gestures relative to the artifacts are maintained.

Heuristic 3: Provide Consequential Communication of an Individual's Embodiment

A person's body interacting with a physical workspace is a complex information source with many degrees of freedom. In these settings, bodily actions such as position, posture, and movements of head, arms, hands, and eyes *unintentionally* "give off" information which is picked up by others [33]. This is a source of information since "watching other people work is a primary mechanism for gathering awareness information about what's going on, who is in the workspace, where they are, and what they are doing" [15]. Similarly, visible activity is an essential part of the flow of information fundamental for creating and sustaining teamwork [33]. This form of consequential bodily communication is not intentional in the same manner as explicit gestures (heuristic 2): the producer of the information does not intentionally undertake actions to inform the other person, and the perceiver merely picks up what is available.

Unintentional body language can be divided into two categories. *Actions coupled with the workspace* include such activities as gaze awareness (i.e. knowing where another person is looking), seeing a participant move towards an object or artifact, and hearing characteristic sounds as people go about their activities. *Actions coupled to conversation* are the subtle cues picked up from our conversational partners that help us continually adjust our verbal behaviour [e.g. 2,24,31]. Some of these cues are visual: facial expressions, body language (e.g. head nods), eye contact, or gestures emphasizing talk. Others are verbal: intonation, pauses, or the use of particular words. These visual and verbal cues provide *conversational awareness* that helps people maintain a sense of what is happening in a conversation. This in turn allows us to mediate turn-taking, focus attention, detect and repair conversational breakdown, and build a common ground of joint knowledge and activities [1]. For example, eye contact helps us determine attention: people will start an utterance, wait until the listener begins to make eye contact, and then start the utterance over again [10].

Typical groupware support. The goal of supporting consequential communication in real-time groupware is to capture and transmit both the explicit and subtle dynamics that occur between collaborating participants. This is no easy task. While the embodiment techniques previously discussed (heuristic 2) are a start, they are very limited. For example, telepointers allow us to see people moving towards an object. They can also change their shape to reflect a natural action such as pointing or writing [12]. Telepointers may hint at where its owner is looking, although there is no guarantee that the person is really doing so. Avatars can go one step further by linking the 'gaze direction' of the avatar to the point of view, thus signaling its owner's approximate field of view in the environment. While these do help, the impoverished

match of these embodiments to a person's actual body movements means that there are many consequential gestures that are not captured and transmitted.

Video systems that mix a person's video embodiment into the workspace are more successful but unfortunately quite limited [e.g. 20]. Special care must be taken with camera placement, otherwise eye contact and gaze awareness will be inaccurate or incorrect—in most desktop systems, we see speakers 'looking' at our navels or hairline simply because cameras are often mounted on top or underneath the monitor. The use of compressed video results in small, jerky and often blurred images that lose many of these subtle body cues. Even with full video, zooming the camera in to capture facial expressions ('talking head' view) means that other body gestures are not visible. Yet zooming out to include the whole body compromises image fidelity and resolution. A notable exception is Ishii's ClearBoard [20], as he goes to great length to keep gaze awareness—whether intentional or consequential—correct.

Audio is also a concern for consequential communication. When the voice channel is of low audio quality, the clarity of a person's speech dynamics are compromised. When the voice is non-directional, people find it difficult to associate a voice with a particular speaker (e.g. multi-point conferencing). With half-duplex channels, people cannot speak at the same time, making it harder for listeners to interrupt or to inject back-channel utterances such as 'ok' and 'ums'. Speaker identification and turn-taking is difficult when a teleconference involves four or more people [7].

Advanced systems mimic the spatial relationships between people in a multi-point collaboration by letting individuals turn their heads and speak to one another just as they do in real life. This is accomplished by positioning monitors and cameras within and across sites so that all people are seen and heard in the same relative position on their video and audio surrogates. People's images and voices are projected onto separate video monitors and speakers. One compelling early example was the MIT Media Labs' 'talking heads' project. They fashioned a transparent physical mask of a participant, mounted it on a motorized platform at the remote site, and then projected the video image of the participant into the mask. Through sensors, the mask would move to reflect the person's actual head movement.

Heuristic 4: Provide Consequential Communication of Shared Artifacts (i.e. Artifact Feedthrough)

In face-to-face settings, consequential communication also involves information *unintentionally* given off by artifacts as they are manipulated by individuals [e.g. 5,9]. This information is called feedback when it informs the person who is manipulating the artifact, and *feedthrough* when it informs others who are watching [5]. Physical artifacts naturally provide visual and acoustic feedback and feedthrough. Visually, artifacts are physical objects that show their state in their physical representation, and that form spatial relationships with one another. In addition, an artifact's appearance sometimes shows traces of its history of how it came to be what it is (e.g., object wear). Acoustically, physical artifacts make characteristic sounds as they are manipulated (e.g. scratch of a pencil on paper) [9]. By seeing and hearing an artifact as it is manipulated, people can easily determine what others are doing to it.

Another resource available in face-to-face interactions is the ability to identify the person manipulating an artifact. Knowing who produced the action provides context for making sense of this action, and helps collaborators mediate their interactions.

Actions within a shared workspace are often used to bid for turn-taking in a conversation; therefore, being able to associate the action with the initiator helps others yield their turn [38].

Due to the spatial separation between artifact and actor, feedthrough tends to be the only vehicle for sharing artifact information amongst groupware participants. However, groupware complicates feedthrough since it limits the expressivity of artifacts. For instance, direct manipulation of artifacts in a virtual workspace (e.g. dragging and dropping a small object) can easily go unnoticed since these actions are not as visible when compared to face-to-face equivalents. They can also happen instantaneously (e.g., a click of a button), leaving little warning of their occurrence and little time to see and interpret them. Similarly, indirect manipulation of a virtual artifact (e.g. menu selections) is difficult to connect to the person controlling the action even when we can determine the action's meaning. Unless feedthrough is properly supported by the system, collaboration will be cumbersome.

Typical groupware support At the lowest level, the shared virtual workspace must display the local user's feedback to all remote users. In the event of a direct manipulation of an artifact, the designer must show not only the final position of a moved object but also the selection of the object and the intermediate steps of its move. In groupware, this can be accomplished via *action feedthrough* whereby each individual sees the initial, intermittent, and final state of an artifact as it is manipulated [15]. Early groupware systems imposed "what you see is what I see" (WYSIWIS) view sharing where all participants saw the exact same actions as they occurred in the workspace [e.g. 36]. Similarly, feedthrough must be supported during the indirect manipulation of an artifact. *Process feedthrough* ensures that local feedback of a person selecting an operation or command is also transmitted to all others to help them determine what is about to happen [15]. Intermediate states of indirect manipulation can also be presented via visual techniques such as action indicators and animation. With both types of events, all collaborators must be able to identify the producer of the action. Presenting the interim feedback of an artifact to all participants during an operation ensures changes do not happen instantaneously and that information other people can gather about the activity while it is happening is not reduced. Identifying the producer of the action helps to provide context to it.

Physical objects typically display information regarding how they were created and how they have been manipulated. In contrast, the bland appearance of virtual artifacts reduces what can be learned about their past actions. Objects have a consistent appearance and manipulating them does not automatically leave traces [18]. Their monotony means that people have fewer signs for determining what has happened and what another person has been doing. Techniques for displaying the history of virtual artifacts include 'edit wear' and 'read wear' [18].

In contrast to physical artifacts, virtual ones do not have natural sounds; therefore, many groupware workspaces are silent. If sounds are to be heard, designers must create and add synthetic replacements to groupware [e.g. 9]. Currently, it is difficult to reproduce the subtlety and range of natural workspace sounds. In addition, the directional and proximal components of sounds tend to be weak since workspaces are 2D with limited audio output devices.

Heuristic 5: Provide Protection

In face-to-face settings, physical constraints typically prevent participants from concurrently interacting within a shared workspace. Conversely, groupware enables collaborators to act in parallel within the workspace and simultaneously manipulate shared objects. Concurrent access to the shared space is beneficial since collaborators can work in parallel, and because it helps negotiate the use of the space. In addition, it reduces the competition for conversational turn taking since one person can work in the shared space while another is talking and holding the audio floor [38]. On the other hand, concurrent access to objects can introduce the potential for conflict. People can inadvertently interfere with work that others are doing now, or alter or destroy work that others have done. People should be protected from these situations.

Anticipation plays an important role in providing protection. People learn to anticipate each other's actions and take action based on their expectations or predictions of what others will do in the future. Amongst other things, participants can in turn avoid conflicting actions. Therefore, collaborators must be able to keep an eye on their own work, noticing what effects others' actions could have and taking actions to prevent certain kinds of activity.

Social scientists have found that people naturally follow social protocols for mediating their interactions, such as turn-taking in conversations, and the ways shared physical objects are managed [2,38]. Therefore, concurrency conflicts may be rare in many groupware sessions since people mediate themselves—but this can only happen if people have a good sense of what is going on. People are quite capable of repairing the negative effects of conflicts and consider it part of the natural dialog [13]. Of course, there are situations where conflict can occur, such as accidental interference due to one person not noticing what another is doing. In some (but not all) cases, slight inconsistencies resulting from conflicts may not be problematic.

Typical groupware support. Many groupware systems give all collaborators equal rights to all objects. To provide protection, they rely on people's natural abilities to anticipate actions, mediate events and resolve conflicting interactions. The system's role is limited to providing awareness of others' actions and feedback of shared objects. For example, remote handles can graphically warn users that someone else is already using an item. Although social protocols will generally work, this approach may not be acceptable under certain situations. For instance, by allowing conflicts to occur systems force the users to resolve these events after they have been detected. This is undesirable if the result is lost work. Users may prefer 'prevention' to 'cure'. The quality of awareness will also not function well with high-latency communications where there is a delay in delivering one user's actions to others.

To assist with social protocols, technical measures such as access control, concurrency control, undo, version control, and turn-taking have been implemented. For example, concurrency control could manage conflicting actions and thus guard against inconsistencies. However, concurrency control in groupware must be handled differently than traditional database methods since the user is an active part of the process. People performing highly interactive activities will not tolerate delays introduced by conservative locking and serialization schemes. Access control can also be used to determine who can access a groupware object and when. Access control may be desirable when people wish to have their own private objects that only they can manipulate and/or view. Within groupware access control must be managed in a

light-weight, fine-grained fashion. If not, it will be intrusive: people will fight with the system as they move between available and protected objects.

Heuristic 6: Management of Tightly and Loosely-Coupled Collaboration

Coupling is the degree to which people are working together [32]. In general terms, coupling is the amount of work that one person can do before they require discussion, instruction, information, or consultation with another person. People continually shift back and forth between *loosely- and tightly-coupled collaboration* where they move fluidly between individual and group work. To manage these transitions, people should be able to focus their attention on different parts of the workspace when they are doing individual work in order to maintain awareness of others. Knowing what others are doing allows people to recognize when tighter coupling could be appropriate. This typically occurs when people see an opportunity to collaborate, need to plan their next activity, or have reached a stage in their task that requires another's involvement. For example, assisting others with their task is an integral part of collaboration whereby individuals move from loose to tight coupling. Assistance may be opportunistic and informal, where the situation makes it easy for one person to help another without a prior request. Awareness of others in these situations helps people determine what assistance is required and what is appropriate. Assistance may also be explicitly requested. Gutwin observed one participant making an indirect statement indicating that they wanted assistance, and their partner left their tasks to help out, and then returned to what they were doing [15]. In either case, to assist someone with their tasks, you need to know what they are doing, what their goals are, what stage they are in their task, and the state of their work area.

Typical groupware support. The traditional WYSIWIS approach ensures that people stay aware of one another's activities, but is often too restrictive when people regularly move back and forth between individual and shared work [6,9]. More recent systems allow people to move and change their viewports independently, allowing them to view the objects that interest them. This is called relaxed-WYSIWIS view sharing [37]. Unfortunately, when people can look at different areas of the workspace, they are blinded to the actions that go on outside their viewport unless the designer accounts for this. This difficulty of maintaining awareness of others when we are not working in the same area of the workspace is exacerbated because display areas are small and of very low resolution when compared with the normal human field of view. The reduction in size forces people to work through a small viewport, thus only a small part of a large workspace is visible at a time. The reduction in resolution makes artifacts harder to see and differentiate from one another; therefore, visual events can be more difficult to perceive. Groupware must address these issues by providing visual techniques that situate awareness information in the workspace. However, techniques encounter the same visibility problem: the relevant part of the workspace has to be visible for the techniques to be of any use. When the workspace is too large to fit into a single window, the entire area outside the local user's viewport cannot be seen unless special techniques are included that make the relevant parts of the workspace visible. Examples are included here.

Overviews provide a birds-eye view of the entire workspace in a small secondary window. A properly designed overview makes embodiments, actions and feedthrough

visible, regardless of where they occur in the workspace. They also show where people are working in the workspace and the general structure of their activities. An overview showing additional awareness information like *view rectangles* (an outline showing what another can see) and telepointers is called a *radar view*. With radar views, people can easily pursue individual work by moving their view rectangle in the radar view to a different part of the workspace. Conversely, if they want to work closely together, they can quickly align their view rectangles atop one another. *Detail views* duplicate a selected part of the workspace. This secondary viewport provides a closer look at another person's work area: they show less of the workspace than an overview does, but what they do show is larger and in greater resolution. *Focus+context views* provides both local detail and global context within the same display, usually through information visualization techniques such as fisheye views or special lenses.

Heuristic 7: Allow People to Coordinate Their Actions

An integral part of face-to-face collaboration is how group members mediate their interactions by taking turns and negotiating the sharing of the common workspace [38]. People organize their actions in a shared workspace to help avoid conflict with others and efficiently complete the task at hand. Coordinating actions involves making some tasks happen in the right order, at the right time while meeting the task's constraints [15]. Symptoms of poor coordination include people bumping into one another, duplication of actions, or multiple individuals attempting to concurrently access shared resources.

Coordination of action is a higher order activity built upon many mechanisms listed in the previous heuristics. Within a shared workspace, coordination can be accomplished via two mechanisms: "one is by explicit communication about how the work is to be performed ... another is less explicit, mediated by the shared material used in the work process" [30]. The first mechanism implies the need to support intentional and appropriate verbal communication (heuristic 1). The second uses workspace awareness to inform participants about the temporal and spatial boundaries of others' actions. In face-to-face interactions, the close physical proximity among the collaborators allows them to mediate actions since they are peripherally aware of others and all actions. Thus collaborators can fit the next action into the correct sequence.

Both mechanisms are beneficial for different levels of coordination. At the fine-grained level, awareness is evident in continuous actions where people are working with shared objects. One example is the way that people manage to avoid making contact with one another while collaborating within a confined space. On a larger scale, groups regularly reorganize the division of labour i.e., what each person will do next as the task progresses. These decisions depend in part on what the other participants are doing and have done, what they are still going to do, and what is left to do in the task. Knowing activities and locations can help determine who should do what task next.

The coordination of activities at both levels is also assisted by anticipation. People take action based on their expectations or predictions of what others will do in the future. Anticipation is integral to fine-grained coordination whereby people predict events by extrapolating forward from the immediate past. If you see someone

reaching for a pen, you might predict that they are going to grab it. In turn, you can take action based on this prediction (e.g. pick up the pen and hand it to the other person or alter your own movements to avoid a collision). In this case, anticipation is supported by the up-to-the-moment knowledge of the activity (i.e. where the other person's hand is moving) and the location (i.e. the location of the hand in relation to the pen). In addition, your prediction could have taken into account other knowledge, such as the other person's current activities and if they required a pen. When prediction happens at a larger scale, people learn which elements of situations and tasks are repeated and invariant. People are experts at recognizing patterns in events, and quickly begin to predict what will come next in situations that they have been in before [15].

Typical groupware support. People are generally skilled at coordinating their communication and interaction with each other. Consequently, tools used to support collaboration should not impose a structure that attempts to manage the interactions for them. Instead, tools should facilitate the participants' own abilities to coordinate their communication and collaboration. Workspace awareness provides people with information they need to determine whether others' behaviour or current workspace events match the patterns that they have learned. Therefore, groupware must allow individuals to remain aware of others within a shared workspace and the nature of their actions. The visual techniques presented in heuristics 2 through 5 will help to establish this awareness. In addition, collaborators must be able to see all actions within the context of the entire workspace even when people are working in different parts of it. Implementing relaxed WYSIWIS helps to ensure that this is capable within groupware. Finally, collaborators must have the ability to communicate verbally (e.g. via an audio link). It is the inclusion of all this support within groupware systems that enables collaborators to effectively coordinate their activities at both a fine-grain level or on a larger scale.

Heuristic 8: Facilitate Finding Collaborators and Establishing Contact

One problem with groupware is that it is not clear how people actually begin their groupware meetings. In everyday life, relatively few meetings are *formal* i.e., scheduled in advance with pre-arranged participants. These are usually arranged via e-mail, telephone, formal meeting requests, etc. In reality most meetings are *informal* encounters: unscheduled, spontaneous or one-person initiated meetings. These are facilitated by physical proximity since co-located individuals can maintain awareness of who is around. Under these circumstances, people frequently come in contact with one another through casual interactions (e.g. people bump into each other in hallways) and are able to initiate and conduct conversations with little effort. While conversations may not be lengthy, much can occur: people coordinate actions, exchange information, or offer opportunities. Successful teams rely on regular, informal, and unplanned contact between their members [3,21]. It is more difficult to support informal groupware encounters since the bottleneck to rich spontaneous interactions is distance [21].

In electronic communities, people are distributed. Therefore we need to support how the group determines who is around and their availability if they are to initiate contact in a real-time groupware session. Even when potential collaborators have

been identified, many mundane factors now interfere with making contact over computers. People must know electronic addresses and select from many communication channels and applications that are available to the group. People must ready software, equipment, and each other well in advance for real-time remote conferencing. From a technical perspective, sites may not have the same software; workstations may not support the necessary media (e.g., digital audio); specialized equipment may not be available (e.g., video cameras); poor networks may limit interactions; and applications must run across platforms.

Typical groupware support. Groupware applications must overcome the distance barrier that inhibits informal encounters in order to be successful. Information on potential collaborators must be provided so that they can be easily found and their availability for group work can be determined. If collaborators are able and willing to engage in a groupware session, you must be able to initiate contact with minimal effort. Instant messaging provides a simple but limited mechanism for seeing who is around. Another more comprehensive approach is used in TeamWave [12]: a room metaphor helps people know who is around and makes it easy to move into conversation and work. Specifically it does the following:

- *Being available.* People can pursue single user activities in a room. Analogous to a physical room used for both individual and group activities, people will be around more often and thus available for real time encounters.
- *Knowing who is around and available for interaction.* Within a spatial setting, we sense who else is around as we walk down hallways, glance into offices, and see others in public spaces. We judge their availability by a variety of cues such as if their door is open and how busy they look. TeamWave's room metaphor provides a similar sense of presence and awareness by displaying the inhabitants of each room within the electronic community (via a user list) as well as status information. To judge availability, four 'door states' indicate a person's desire to be interrupted. As in real life, a wide and partially open door icon indicates a willingness to accept interruptions, while a barred door suggests that the room and its inhabitants are inaccessible.
- *Establishing contact.* There are several ways of establishing contact with individuals. One can just enter a populated room, and they are immediately connected to others: room occupants also see the person enter because their picture appears. A room-specific chat facility allows conversations to occur. If a person would rather initiate a conversation before entering a room, they can page somebody. Phone calls can be quickly established since phone numbers are available in each person's business card. To establish contact with others not currently logged on to TeamWave, a person can leave a note in a room suggesting a meeting time and place.
- *Working together.* The power of the room metaphor is that, once in a room, the working context is immediately available. All tools and room artifacts are at hand and new tools can be easily added.

4 Summary and Future Work

Our primary goal in this paper was to describe our groupware heuristics. We are especially concerned about keeping them simple enough to help an inspector look for many of the actions that are crucial to smooth and effortless collaboration within a shared workspace.

Our next step, which we have only just begun, is to validate how inspectors actually use these heuristics for uncovering problems. To gain some preliminary insight into the practicality of these groupware heuristics as a discount usability methodology, we performed an inspection of an object-oriented groupware drawing application. We uncovered many usability problems that we believe we would have otherwise overlooked in a casual inspection of the interface. The heuristics helped us to focus our attention on the critical issues pertaining to effective communication and collaboration among groupware participants. In addition, we were able to perform the inspection within a couple of hours. Consequently, we believe our initial development and application of these mechanics of collaboration heuristics appears promising.

Of course, the above evaluation is not enough to validate our groupware heuristics as a discount usability methodology. The number of usability problems uncovered during our inspection is not truly indicative of a typical practitioner since we (as evaluators) are intimately familiar with the groupware heuristics. Consequently, we are now planning a formal evaluation. Outside evaluators will inspect several real-time shared workspace applications using the groupware heuristics. The aim is to assess the ability of usability specialists to learn and apply the groupware heuristics by analyzing the quantity of problems detected by each evaluator. Ideally, each evaluator will uncover a large relative proportion of all problems found by all evaluators; however, even Nielsen found that the average performance of individual evaluators to be modest [25]. Thus we look to define the average proportion of usability problems found as a function of the number of evaluators performing the heuristic evaluation. As with conventional HE, we hope that only a small number of evaluators (about 3) are required to find a reasonably high proportion of the problems. This would allow the technique to remain low cost in terms of resources.

References

1. Clark, H. (1996). *Using Language*. Cambridge University Press, Cambridge.
2. Clark, H. and Brennan, S. (1991). Grounding in Communication, in *Readings in Groupware and Computer Supported Cooperative Work*, R. M. Baecker ed., Morgan-Kaufman Publishers, 222-233.
3. Cockburn, A. and Greenberg, S. (1993). Making contact: Getting the group communicating with groupware. *Proc ACM COCS'93 Conference on Organizational Computing Systems*, 31-41.
4. Cugini, J., Damianos, L., Hirschman, L., Kozierok, R., Kurtz, J., Laskowski, S. and Scholtz, J. (1997). Methodology for Evaluation of Collaboration Systems. *The evaluation working group of the DARPA intelligent collaboration and visualization program, Rev. 3.0*. <http://zing.ncsl.nist.gov/nist-icv/documents/method.html>.
5. Dix, A., Finlay, J., Abowd, G., and Beale, R. (1993). *Human-Computer Interaction*, Prentice Hall.

6. Dourish, P., and Bellotti, V. (1992). Awareness and Coordination in Shared Workspaces. *Proc. ACM CSCW'92*, 107-114.
7. Egado, C. (1990). Teleconferencing as a Technology to Support Cooperative Work: Its Possibilities and Limitations, in *Intellectual Teamwork: Social and Technological Foundations of Cooperative Work*, J. Galegher, R. Kraut, and C. Egado ed., Lawrence Erlbaum, 351-372.
8. Eriback A.L. and Hook, K. (1994). Using Cognitive Walkthrough for Evaluating a CSCW Application. *Proc ACM CHI '94*, 91-92.
9. Gaver, W. (1991). Sound Support for Collaboration, *Proc. 2nd ECSCW'91*, 293-308.
10. Goodwin, C. (1981). *Conversational Organization: Interaction Between Speakers and Hearers*, Academic Press.
11. Greenberg, S., Fitzpatrick, G., Gutwin, C. and Kaplan, S. (1999). Adapting the Locales Framework for Heuristic Evaluation of Groupware. *Proc. OZCHI'99 Australian Conference on Computer Human Interaction*, 28-30.
12. Greenberg S. and Roseman, M. (1998). Using a Room Metaphor to Ease Transitions in Groupware. Research report 98/611/02, Dept of Computer Science, University of Calgary, Canada.
13. Greenberg, S., Roseman, M., Webster, D. and Bohnet, R. (1992). Human and technical factors of distributed group drawing tools. *Interacting with Computers*, 4(1), 364-392.
14. Grudin, J. (1988) Why CSCW Applications Fail: Problems in the Design and Evaluation of Organizational Interfaces. *Proc ACM CSCW'88*, 85-93.
15. Gutwin, C. (1997). Workspace Awareness in Real-Time Distributed Groupware. Ph.D Thesis, Dept of Computer Science, University of Calgary, Canada.
16. Gutwin, C. and Greenberg, S. (1999). The Effects of Workspace Awareness Support on the Usability of Real-Time Distributed Groupware. *ACM Transactions on Computer-Human Interaction*, 6(3), 243-281.
17. Gutwin, C. and Greenberg, S. (2000). The Mechanics of Collaboration: Developing Low Cost Usability Evaluation Methods for Shared Workspaces. *IEEE 9th Int'l Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET-ICE'00)*.
18. Hill, W., Hollan, J., Wroblewski, D., and McCandless, T. (1991) Edit Wear and Read Wear. *Proc CHI '91*, 3-9.
19. Heath, C., Jirotko, M., Luff, P., and Hindmarsh, J. (1995). Unpacking Collaboration: The Interactional Organisation of Trading in a City Dealing Room. *Computer Supported Cooperative Work*, 3(2), 147-165.
20. Ishii, H., Kobayshi, M., and Grudin, J. (1992). Integration of interpersonal space and shared workspace: ClearBoard design and experiments. *Proc ACM CSCW'92*, 33-42.
21. Kraut, R., Egado, C., and Galegher, J. (1988). Patterns of Contact and Communication in Scientific Research Collaboration Remote Communications. *Proc ACM CSCW'88*, 1-12
22. Mack, R. and Nielsen, J. (1994). Executive Summary. In *Usability Inspection Methods*, Nielsen, J. and Mack, R., Eds. John Wiley and Sons, New York, 1-23.
23. McGrath, J. (1996). Methodology matters: Doing research in the behavioural and social sciences. In R. Baecker, J. Grudin, W. Buxton and S. Greenberg (eds) *Readings in Human Computer Interaction: Towards the Year 2000*, Morgan-Kaufmann, 152-169.
24. McLaughlin, M., (1984). *Conversation: How Talk is Organized*, Sage, Beverly Hills.
25. Nielsen, J. (1992). Finding usability problems through heuristic evaluation. *Proc ACM CHI'92*, 372-380.
26. Nielsen, J. (1993). *Usability Engineering*. Academic Press.
27. Nielsen, J. (1994). Chapter 2: Heuristic Evaluation. In *Usability Inspection Methods*, Nielsen, J. and Mack, R., Eds. John Wiley and Sons, New York, 25-62.
28. Nielsen, J. (1994). Enhancing the explanatory power of usability heuristics. *Proc ACM CHI '94*, 152-158.
29. Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces. *Proc ACM CHI '90*, 249-256.

30. Robinson, M. (1991) Computer-Supported Cooperative Work: Cases and Concepts. *Proc Groupware'91*, 59-75.
31. Sacks, H., Schegloff, E., and Jefferson, G. (1974). A Simplest Semantics for the Organization of Turn-Taking for Conversation, *Language*, 50, 696-735.
32. Salvador, T., Scholtz, J., and Larson, J. The Denver Model of Groupware Design, *SIGCHI Bulletin*, 28(1), 52-58
33. Segal, L. (1994). Effects of Checklist Interface on Non-Verbal Crew Communications, NASA Ames Research Center, Contractor Report 177639.
34. Short, J., Williams, E., and Christie, B. (1976). Communication Modes and Task Performance, in *Readings in Groupware and Computer Supported Cooperative Work*, R. M. Baecker Eds., Morgan-Kaufman Publishers, Mountain View, CA, 169-176.
35. Smith, R., O'Shea, T., O'Malley, C., Scanlon, E., and Taylor, J. (1989). Preliminary experiences with a distributed, multi-media, problem environment. *Proc EC-CSCW'89*.
36. Stefik, M., Foster, G., Bobrow, D., Kahn, K., Lanning, S., and Suchman, L. (1987). Beyond the Chalkboard: Computer Supported for Collaboration and Problem Solving in Meetings. *Comm. ACM*, 30(1), 32-47.
37. Stefik, M., Bobrow, D., Foster, G., Lanning, S., and Tatar, D. (1987) WYSIWIS Revised: Early Experiences with Multiuser Interfaces. *ACM Transactions on Office Information Systems*, 5(2), 147-167.
38. Tang, J. (1991). Findings from Observational Studies of Collaborative Work. *Intl J Man-Machine Studies*, 34(2), 143-160.
39. Tang, J. and Leifer, L. (1988). A Framework for Understanding the Workspace Activity of Design Teams. *Proc ACM CSCW'88*, 244-249.
40. Tang, J. and Minneman, S. (1990). VideowDraw : A Video Interface for Collaborative Drawing. *Proc ACM CHI'90*, 313-320.
41. Tatar, D., Foster, G., and Bobrow, D. (1991). Design for Conversation: Lessons from Cognoter. *International Journal of Man-Machine Studies*, 34(2), 185-210.

Discussion

D. Salber: Even though your focus is on shared workspaces your heuristics might apply to email or instant messaging. Did you look at heuristics for a broader range of groupware systems.

K. Baker: Not really. Some of these heuristics apply to groupware in general eg finding collaborators, establishing sessions. These are based on a theoretical framework. They describe how people actually work together. They are therefore important in these applications. We don't have (or have not found) a framework for more general groupware such as email yet.

P. Curzon: One of the heuristics was about the importance of informal collaboration. However, most real world informal collaboration occurs when you are taking a break from work and the computer. If informal collaboration is some important isn't this always going to be a problem with groupware.

K. Baker: The important thing is that these informal conversations are quick to initiate. The heuristic "facilitate finding collaborators and establishing contact" is important here. Setting up communication in groupware has to be lightweight and simple. For most existing groupware systems this is not the case. There are important differences between planned and informal collaboration but this heuristic is important to both.

P. Van Roy: More a comment: The book "Understanding comics" by Scott McCloud gives interesting insights on the interaction between words, images, sequence, levels of abstraction for example the concept of closure. Given the two images - the human automatically fills in the gap between them.

K. Baker: Interesting, please send us a reference.

P. Smith: What do your heuristics have to say about baton passing as a mechanism in collaborative environments.

K. Baker: Multiple heuristics may be applicable. For instance, "allow people to coordinate" and "provide protection" are both related to this problem. It is important to apply heuristics with respect to the particular circumstances. For instance, in formal meeting, we need significantly more control/protection. Like Nielsen's heuristics these are anchor points. To take the knowledge of the inspector and make sure that they look for all of the necessary problems. There is training list material that goes with these points. In groupware, training material and experience are more vital as novice evaluators may not have much experience with groupware systems. Even some expert evaluators who looked at early versions of these heuristics were still left hanging, and some asked for checklists. As an example one commercial groupware tool that we looked at initially looked really nice. However, when the authors applied these heuristics they found lots of serious problems. The students didn't report that many problems. However, this was because the system didn't even support fundamental activities. Modern systems can miss really basic things. So such heuristics are necessary. One thing that these heuristics do is help novice evaluators to articulate the problems.