Computer Supported Cooperative Work (CSCW) (2016) 25:279–293
DOI 10.1007/s10606-016-9244-y

Implications of We-Awareness to the Design of Distributed Groupware Tools

Saul Greenberg[1] & Carl Gutwin[2]
[1]*University Of Calgary, Calgary, Alberta, Canada (E-mail: saul.greenberg@ucalgary.ca);* [2]*University of Saskatachewan, Saskataoon, Saskatchewant, Canada (E-mail: gutwin@cs.usask.ca)*

Abstract. We-awareness is the socially recursive inferences that let collaborators know that all are mutually aware of each other's awareness. While we-awareness is easy afforded in face to face collocated collaboration, it is much more difficult to design distributed groupware tools to provide equivalent capabilities: there can be no awareness unless it is programmed in via system features. We identify a series of questions that must be considered if we-awareness is to be supported. What types of awareness information is crucial and should thus be added to the 'blank slate' of a screen sharing system? How can that awareness information be captured through technology, and what information will be lost during this capture process? How should that information be translated, transformed and encoded into a digital form, and—as part of that—what information will be altered as part of that translation process? How will that information be transmitted, and what are the network effects in terms of that information being received in a timely manner? How will that information be represented to other participants in order to enable the rich and subtle interactions that occur in the face-to-face setting? We illustrate the nuances of these questions and why they are difficult to answer by revisiting several prior technical solutions to we-awareness.

Key words: awareness, groupware design, groupware usability, real-time distributed groupware

1. Introduction

When small groups of face to face people work together, they maintain awareness of what others are doing and what they are working on. Amongst other things, awareness is the glue that helps coordinate joint action, that creates opportunities for moving between loosely-coupled and tightly-coupled activities, and that informs mutual understanding of both individual and group actions. Unsurprisingly, CSCW researchers have long argued that some kind of awareness support is essential in groupware systems supporting distributed real-time collaboration (e.g., Dourish and Bellotti 1992). Yet the term 'awareness' is a somewhat nebulous, and many researchers (including ourselves) have tried to define and operationalize it so that it can be readily used by groupware designers.

Within this backdrop, Tenenberg revisits awareness (2016, this issue). In particular, he defines the term 'we-awareness' in tightly-coupled work, i.e., the socially recursive inferences that let collaborators know that all are mutually aware of each other's awareness [ibid]. He contrasts this to 'I-awareness', which focuses only on the actions, communication, and resources that are publicly available and which thus neglects the we-awareness idea of shared intentionality. He does a fine job of framing these notions in not only the philosophical and CSCW literature, but in proving its presence and importance via a fine-grained analysis of co-located pair programming.

Our own research interests in awareness over several decades have been driven by groupware design concerns, where we focused on developing systems supporting a small group of distance-separated collaborators working together over a shared visual work surface. Importantly, we wanted to build these systems on a strong theoretical design foundation rather than mere intuition. In particular, we wanted to understand the importance of awareness to small group collaborative activities, the information that comprises awareness, how that awareness information is gathered, and how that information is used in collaboration (e.g., Gutwin and Greenberg 2002). As with Tenenberg, we also believed that we-awareness would be crucial for many aspects of both loosely and tightly-coupled collaborative work over these workspaces.

We began our building efforts with a fairly good arsenal of theory (including theory that we developed ourselves). Importantly, we wanted to somehow operationalize awareness within our groupware tools intended for remote collaborators working together over a shared visual work surface. Yet this proved non-trivial several decades ago, and is still non-trivial today. The crucial question remains:

How can we develop tools to support we-awareness?

Tenenberg's paper does not answer this question. While he does briefly mention tool support for we-awareness in his Section 6 (2016, this issue), they are mostly pointers to a few techniques developed by others, where there is little in the way of critical reflection of those techniques.

Because we are primarily tool-builders, our commentary will build upon this question and his discussion. We focus mostly on distributed groupware. This is because remote collaborators can only stay aware of each other *through* the tools provide by the system (*cf.* face to face collaboration where people can see and hear each other, see Section 3). In this paper, we will delve into implications for tool support for we-awareness in distributed groupware more deeply. We will touch upon several previously presented ideas that, we believe, are foundational to we-awareness.

We do not offer definitive answers, even though we recognize that several design strategies already exist that provide—or at least hint at—the socially recursive inferences that comprise mutual awareness. As we will see, we-awareness support is non-trivial. There are many constraints and limitations that restrict what can be sensed, transmitted and displayed across the technology held by remote participants, and design trade-offs abound. While Tenenberg identifies we-awareness, we will show that implementing effective tool solutions to we-awareness is a complicated and very interesting challenge, fraught with constraints and tradeoffs. Indeed, we suspect that the philosophical problems raised by Tenenberg arose from the failure of groupware tools to effectively support we-awareness.

2. The novelty of we-awareness?

The theory behind tool support for awareness has a long history, and we begin there. Our opinion is that Tenenberg's notion of we-awareness is not novel, nor have prior works focused exclusively on I-awareness. We believe that we-awareness is included in broad CSCW theories such as common ground (Clark et al. 1991) and distributed cognition (Hutchins 1991) and in narrower theories of awareness including our own framework for workspace awareness (Gutwin and Greenberg 2002). For example, Clark et al. (1991) introduce their discussion on grounding in tightly-coupled communication by emphasizing mutuality, i.e., that each person is trying to understand what the other knows at the moment. In our view, this is we-awareness. They begin with an example of two people playing a duet:

> "They cannot even begin to coordinate on content without assuming a vast amount of shared information or common ground—that is, mutual knowledge, mutual beliefs, and mutual assumptions … And to coordinate on process, they need to update their common ground moment by moment. All collective actions are built on common ground and its accumulation." (p. 127, Clark et al. 1991)

And later, using the example of two people talking:

> "Speech is evanescent, and so Alan must try to speak only when he thinks Barbara is attending to, hearing, and trying to understand what he is saying, and she must guide him by giving him evidence that she is doing just this. Accomplishing this, once again, requires the two of them to keep track of their common ground and its moment-by-moment changes." (p. 128, Clark et al. 1991)

Groupware system designers have also developed various interaction methods that support at least some aspect of we-awareness, albeit with varying levels of success. In the following sections, we will present and critique several of these systems—many from our own early work (as we are most familiar with them) and some from others—to illustrate our subsequent discussion. Our example systems are indicative of particular awareness approaches, and are by no means exhaustive. We use them merely as illustrations to point out why we-awareness support is difficult to do, along with particular issues associated with translating awareness theory to the practice of tool design.

Regardless of whether we-awareness originates with Tenenberg or in earlier CSCW history, Tenenberg adds much value as he explicates what is perhaps tacit (or easily overlooked) in the past literature and in the various groupware tools produced over time.

3. Real world collaborative affordances *vs.* the blank slate of groupware

To set the scene, we contrast how people acquire awareness during face to face interactions *vs.* through distributed groupware.

The real world affords much in the way people can maintain awareness of each other during collaboration, and many of our social practices revolve around those affordances. For example, much of what we do as social beings is dictated by how we perceive and manage our inter-personal space. This is called *proxemics*. We reduce our physical distance during face to face collaboration, which in turn decreases what we perceive as social distance (Hall 1966). As distances lessen, so does our ability to see details of the other person (body language, facial expressions, gaze direction, etc.), the details of the artifacts and tools they are working with, and the actions they are performing with those tools relative to those artifacts (including sounds generated as a side effect), and how those people are mutually monitoring our own actions. People adopt and constantly adjust particular physical arrangements relative to one another as they engage in focused task-dependent conversational encounters. These arrangements are called *F-formations*, typically formed as a roughly circular cluster of people facing a space reserved for the main activity of the group (for example, where the space contains shared artifacts of interest to the group) (Kendon 1990). People can subtly rearrange their *bodily and gaze orientation* to best balance how they can view items and actions in that space *vs.* how they view other people. For example, people working around a table may work face to face, kitty-corner, or side by side, each which favors a different balance. They can rapidly shift their *gaze direction* to acquire a holistic sense of the entire scene, and to attend to particular features within the scene. Through what is called *micro-mobility,* people can also bring artifacts in the shared space, and adjust their orientation towards the group to promote fine-grained viewing and sharing during co-present collaboration (Luff and Heath 1998). Perception is often easy in the face to face environment. People easily hear not only the speech of others, but non-speech sounds caused as a side effect of people moving their bodies and manipulating artifacts (i.e., *consequential communication* (Segal 1995) and *feedthrough* (Dix et al. 1993)). They see motion through peripheral vision, and can easily scan the environment by glancing around. They use *gestures* to augment communication, to bring attention to shared artifacts, and to indicate information about them. All this helps contribute to awareness.

In contrast, computing technology affords nothing, unless it is programmed in. By default, distributed groupware design begins with a blank slate bounded by technical and environmental constraints. No awareness information will be collected, transmitted and displayed unless the programmer codes it into the system. This will be further influenced by how the technology (e.g., devices) are arranged in the environments of its participants, each who may be in a quite different environment compared to the other (e.g., two participant in a high-end meeting room, one in an office, two others tele-commuting from home and from a coffee shop). Design becomes a matter of deciding what to include in the groupware system, which can only be done if designers understand what is required.

As an example, consider a shared workspace, such as a shared drawing tool that all can see and sketch on. Designers of several early groupware drawing tools thought it sufficient to communicate only the strokes each person made on the drawing surface. For network efficiency, those strokes were displayed to remote viewers only after they were completely transmitted (e.g., Lee 1990). Bodily and gestural actions were not transmitted as they were outside of the workspace. While seemingly reasonable, this approach severely restricted collaborators' interactions in practice. Because gestural actions were not visible, and because speech describing a drawing action was not synchronized with the instantaneous performance of that drawing, it was hard to express ideas. Because gaze and body signals were not visible, it hampered how the group mediated interactions such as turntaking and focusing attention, (Tang 1989, 1991; Greenberg et al. 1992). Because a person could scroll and zoom into a part of the workspace out of view of others (called 'relaxed what you see is what I see, or relaxed-WYSIWIS), collaborators could not easily track what others are doing (Gutwin et al. 1996). Contrast this impoverished "blank slate" system with the tightly-coupled face to face situation as described earlier. For example, when a group works together over a shared surface (such as paper on a tabletop) they can see the drawing of a stroke unfold over time, they can easily perform gestures, they can see and interpret each other's communicative signals, they can understand speech in tandem with actions as they are being performed, and they can glance around to monitor other people's activities.

The difficulties of the "blank slate" can also be seen when thinking about supporting awareness in a distributed version of Tenenberg's pair programming scenario. In his co-located face to face case study, the participants' awareness of the other person's proximity, gaze and orientation direction played a substantial role in supporting the shared work. Let us then consider gaze and orientation awareness by itself as an interesting example of how difficult it can be to identify and translate such face-to-face awareness cues to distributed settings. Distributed pair programming tools typically provide screen sharing, shared editing, and perhaps some capabilities for annotation and pointing. With very few exceptions (e.g., Stotts et al. 2004), screen-sharing tools do not innately support the kinds of interactions seen in Tenenberg's case study, such as seeing where others are looking, or determining a person's orientation towards the work artifacts. Now consider a designer of a distributed groupware tool for pair programming who would like to apply Tenenberg's description of we-awareness to redesign a system for pair programming. Here are some of the questions that designer would have to think deeply about.

- What types of awareness information is crucial and should thus be added to the 'blank slate' of a screen sharing system?
- How can that awareness information be captured through technology, and what information will be lost during this capture process?
- How should that information be translated, transformed and encoded into a digital form, and—as part of that—what information will be altered as part of that translation process?

- How will that information be transmitted, and what are the network effects (e.g., delays, bandwidth requirements) in terms of that information being received in a timely manner?
- How will that information be represented to other participants in order to enable the rich and subtle interactions that occur in the face-to-face setting?

The first question is particularly crucial. The problem of the blank slate is that the designer needs to know beforehand which pieces of information are going to be critical for supporting awareness. For the remaining questions, the designer needs to determine how those pieces of information can be sensed and thus captured, transmitted, and then represented in the interface so that they afford the same kinds of interpretation (and thus awareness) of that information in the distributed context.

One way to approach the first question is to do observational field work to determine what information is required for rich interaction. For example, Tenenberg's case study could be used as the basis for making recommendations about what information is important in pair programming. A difficulty with this approach, however, is that maintaining awareness is often a highly idiosyncratic process. For example, one team of pair programmers might make extensive use of bodily orientation and gaze direction towards the monitors (as in Tenenberg's study), where another pair might make more use of speech cues, handwritten notes, or looking behavior at off-screen objects on the desk. Still others might work primarily from common ground assumptions (based on prior knowledge of how the other person works) or from a structured task organization, rather than from moment-by-moment information updates. The personalities and working styles of participants will also affect this: some may favor a divide and conquer loosely-coupled approach (i.e., largely individual work with occasional cross checks), while others may favor continual tightly coupled interaction (i.e., largely continually collaborative). In some tasks, the awareness needs seen in one instance will in fact be generalizable to other settings and other people. But, people are able to carry out collaboration in many different ways (which we have learned time and again when evaluating our tools). Thus it is difficult to guarantee that a particular methodology for identifying awareness information as used by a particular group (and the corresponding technical solutions that arises from it) will lead to successful support for awareness for another group. Unlike the real world, technical solutions are rarely flexible enough to accommodate a broad range of awareness strategies.

Determining what information is important is only the first part of the blank-slate problem. Next, the designer must somehow translate that information—whether social theory or observational insights—into a system. The challenge begins with the fact that information that is present in the real world may be difficult to sense using technology, or difficult to translate to other representations at the remote end. This can mean that the presentation of awareness information at a remote site (even if it is the right information) is difficult or impossible to use in the way that it was used in a face-to-face context.

Consider the example of gaze awareness, where a designer has identified it as an important cue in a face-to-face context (perhaps based on Tenenberg's study or on prior work such as Ishii and Kobayashi 1992) and now wishes to apply that to a distributed pair programming system. One solution is to somehow capture gaze direction algorithmically. Yet sensing gaze direction via technology is a non-trivial problem. It requires technology to track of head position and eye direction (which may be intrusive), and then calibrating the tracked values against the digital content being looked at. But let us assume for the moment that this information can be accurately determined and translated into a digital stream (e.g., coordinates). The final problem—of representing gaze direction in the distributed pair programming tool—is also not easy to solve. One approach could be to place a visual indicator on the shared screens. While this provides information about where the other person is looking, this representation might still not enable the kinds of awareness seen in the case study. In particular, a visual cue on a computer screen is vastly less rich than an actual body in the space beside you (where we see bodies and heads turning), and so even an accurate representation of gaze direction might go unnoticed, or unheeded or be misinterpreted (we discuss this problem in more detail below).

A second solution approach is to just "throw everything you have" at the problem, and try to provide collaborators with many types and varieties of awareness cues in hopes that they can make use of something in the stream to make their decisions about the other person's activities or plans. Unfortunately, this is expensive and time-consuming, is difficult to justify when it is not clear that the information will even be of value, and may still omit critical information. For example, most video-based systems take the "throw everything you have" approach. The idea is that if people can see into the scene, then they should be able to extract the information required. Yet most video-based systems do not work well. Traditional 'talking heads' video chat systems, for example, do not handle gaze awareness correctly: because of camera positioning and parallax, gaze direction is often mis-aligned. These systems also separate the view of the video from the view of the digital workspace, which means that viewers cannot see another person's gaze references to workspace artifacts. This is why video-based chat systems provide questionable value: while they do show that the other person is present and attending the conversation, its other benefits are not clear.

Yet the idea of 'throwing everything you have' at a problem is not as crazy as it might sound. We-awareness means that people who have shared intentionality will attempt to interpret whatever information they have in light of what they know about their shared goal (and their knowledge of what the other person has been doing, etc.), and then adapt their collaborative behaviors accordingly. Therefore, even though the designer may not have been able to determine all of the elements of information that are needed for supporting awareness, people are often able to do a fairly good job of working together in a rich and complex fashion even with not quite the right information. A good example is a groupware system that show telepointers, as these have been found to enable reasonably rich collaboration (and presumably we-awareness) (Greenberg et al. 1996). While telepointers were originally promoted as a way for people to point and gesture around the workspace, they also signal (albeit roughly) where one is focusing their attention (and thus gaze awareness). Movement of the telepointer (especially if tied to speech) further signals presence, activity, intentions and engagement.

Finally, the blank-slate problem raises another difficulty for designers. When they have the choice to allocate resources in the groupware system (e.g., screen space), they have to decide on how to allocate these resources. There are design pressures that often lead away from decisions that support awareness. In particular, as we discuss further below, there is a trade-off between supporting individual power and supporting group awareness, in which increasing the capabilities of the system for the individual may make it more difficult for others to keep track of what is going on. For example, selecting an object with the mouse and clicking "delete" is much easier than reaching across a table to pick up an object and throw it in the wastepaper basket—but the latter action is also much more visible and thus available to others in the group (who may care a lot about what gets deleted).

4. Technical constraints

All design occurs under constraints. As mentioned in the prior section, the real world is a naturally rich place for collaborators, which offers much in the way of resources and flexibility in choosing what to attend. While we may try to reach or replace this richness and flexibility through our technology, it behooves us to understand several essential technical constraints that can restrict what we can do. Ultimately, tradeoffs, work-arounds and quite different designs have to be considered due to these constraints.

The first constraint is imposed by the *networked technology* used by its distributed participants. Input and output devices range from the traditional (e.g., workstations, tablets, digital surfaces and smart phones, with a mix of text, audio, and video capabilities) to the esoteric (tangible devices, eye tracking, gesture and body tracking, etc.). All are affected by the capabilities of the underlying network (bandwidth limitations, lag, jitter). These technological constraints both limit and afford what can be sensed and collected from the various environments, how it can be delivered in a timely fashion, and ultimately how it can be displayed to others in a salient form (Gutwin 2001). The second constraint is *environmental*: the input devices may capture only a small part of the environment (perhaps along with unwanted noise), while the physical characteristics and placement of output devices may limit how information is perceived by its participants. The third constraint is the *disparity* between the desired information the designer may wish to collect and display *vs.* the actual information that can be culled from the various information sources, transformed into a bit stream suitable for transmission, and then somehow displayed on the other side. A fourth constraint are the *output technologies*. If, for example, all work is expected to be performed on a standard workstation, the display's size and its location limit how and where information is presented.

As an example, let us return to gaze awareness, where a designer decides to capture it using an eye tracker. From an environmental standpoint, the equipment has to be somehow situated (and calibrated) in the space to capture the gaze direction of the viewer. Yet even so, sensing is limited, as the eye tracker only works within a certain field of view and from a certain distance. The designer has to decide whether to capture all saccades (very rapid eye movements), or filter those to eye movements that dwell on a location for at least a certain amount of time. While this information can be transmitted reasonably efficiently as a digital stream of coordinates, small network delays are unavoidable, which means that the received gaze coordinates are always somewhat late (which is especially troubling for rapid saccades). Then there is the question of how one presents these coordinates, perhaps as a dot indicating gaze on the screen, or simulated eyes that appear 'behind' the screen, or even to operate a set of mechanical eyes at the side of the screen. Alternately, the designer may instead decide to capture gaze using video, and just transmit the video stream. From an environmental standpoint, this requires a camera, which is now common place. From as sensing aspect, that camera must somehow be calibrated to align gaze with the artifacts of interest, which is not trivial. Transmission of high fidelity video is reasonable, but often incurs considerable jitter and latency unless dedicated network connections are used. Presentation of that video is also problematic. Most systems place it in a separate window, which means it is disjoint from the workspace. Notably, the early Clearboard (Ishii and Kobayashi 1992) and the later Facetop system (Stotts et al. 2004) layered the video *behind* the workspace, where considerable effort was made to ensure gaze awareness was maintained. Yet this only works for two people, and it is unclear how it could be applied to a larger group. Behind-the-workplace video can also visually interfere with the content on the screen. For example, we would expect the video to work reasonably well alongside high contrast text and line drawings (as typical shown in these systems), but much more poorly with low-contrast photo images (as the video and images would tend to blend together).

5. Reciprocity: balancing awareness

Most of the discussion so far has concerned awareness in general, rather than the specific notion of we-awareness. To this end, we turn to the notion of *reciprocity,* which happens 'for free' in collocated face to face interaction:

> "One of the cardinal assumptions is that if you can see someone else, they can see you and that if you can hear someone else, they can hear you." (Fish et al. 1990, page 7)

More formally, reciprocity can be described as a rule that states that if A can access B via channel C, then B can also access A via channel C (Boyle and Greenberg 2005). In turn, reciprocity provides a natural mechanism for we-awareness support.

If you can see and hear someone else and you know that they can see and hear you, then you know that they are aware of you, and they know that you are aware of them, etc.

Various system designers tried to provide reciprocity to the blank slate of distributed groupware, usually by supplying equal balance to what people saw on either side. We believe this is a reasonable approach. Of course, there are many ways that reciprocity can be operationalized, and reciprocity has many nuances that must be catered to. Consider video-mediated communication, which is usually implemented as a bi-directional stream. A good example is the Video Wall (Fish et al. 1990), which enabled participants across two remote lounges to casually interact through a very large-screen video. While it was explicitly designed to favor reciprocity, technical and environmental constraints meant there were dead spots in the camera's field of view. Fish et al. (1990) mention how, in some cases, people would stand in one of these dead spots and accidentally or covertly observe people on the other side of the wall without their knowledge, which at times introduced confusion or raised privacy concerns. Desktop video chat systems suffer similar imbalances, where it is easy for a participant to stand off-screen while still seeing the other through it.

Most non-video groupware also enable reciprocity to some extent, usually by mirroring exactly the same features on the display of all participants. As one example, graphics drawn on the screen appear identically to all participants. This includes multiple cursors (telepointers), one per person. Similarly, any graphics manipulation done by one person is immediately transmitted and made visible to others as the interaction occurs (e.g., Greenberg et al. 1992). The idea is that all can see one another's cursor as it moves around the workspace, as well as the actions they are performing with it. Again, this reciprocity provides we-awareness. The problem is that various systems allow people to view different parts of a workspace, called relaxed 'what you see is what I see'. Consider two people who have scrolled to two regions of the screen, where they happen to overlap somewhat in the middle. If one person is working on that overlapped area, that person's cursor and actions will be visible to the other person. Yet if the other person is working in the non-overlapped area, that person's cursor and actions will not be visible. Reciprocity, and thus we-awareness, is compromised. There are, of course, various solutions to this problem, typically involve adding a visualization (such as an overview) that shows the entire workspace in miniature. This includes other people's telepointers and their actions (e.g., Gutwin et al. 1996). The catch is that the small size may mean details are lost.

6. Design for individuals, design for groups

Even if perfect reciprocity can be achieved, it is not a blanket solution to we-awareness. This is because reciprocity introduces tradeoffs. Participants in a groupware session act in two roles: as individuals trying to pursue one's own work, and as members of a group pursuing collective work. Designers must try to support both roles. However, the requirements of individuals and groups often conflict, forcing designers to support one at the expense of the other (Gutwin and Greenberg 1998). This tradeoff is particularly evident in the design of interaction techniques for shared workspaces. Individuals demand powerful and flexible means for efficiently interacting with the workspace and its artifacts, while groups require information about each other to maintain awareness. These can produce conflicting requirements. Indeed, the prior literature includes the notion of suppressing we-awareness in various situations. Several examples are listed below.

First, consider the simple action of deleting an object. For the person doing the action, they normally select the item, and hit the delete key. While a highly efficient action for the individual doer, other participants can easily miss that person's intention to delete the object, and even the deletion itself. One design solution is to make small actions larger, in this case, by making the delete action more prominent. This could be done by adding a visual atop the deleted object to emphasise that it has been selected, and then adding an animation to the actual delete event to make it more salient, where it perhaps grows in size for a few seconds before shrinking to nothing (Gutwin and Greenberg 1998).

Next, consider the simple action of raising a menu. The person doing the action needs time to see the menu items, scan for the item desired, and then move the cursor over the item to select it. Yet the same full-sized menu can be highly distracting for the other viewers, especially if it appears atop of where the viewers are working. One design solution is to make large actions smaller, in this case, by showing a semi-transparent menu (or alternately just the menu item being selected attached to the other person's cursor (Gutwin and Greenberg 1998).

A third problem arises from how people want to see their work. One of the strengths of a digital workspace is that people can alter the representation to fit their task. Yet in collaborative work, if two people are doing different things (or perhaps have different roles), one person may want to use a representation that differs from how the other person wants to view it. An example is a hierarchical display of items, where one may want to view it as an indented list and the other as a graphical tree akin to an organizational chart (Greenberg et al. 1992). The challenge is how to show all collaborator's location and activities across both representations.

7. Co-present awareness vs. digital embodiments

A main reason given above as to why it would be difficult to successfully represent gaze awareness in a distributed pair-programming tool is that a marker on the screen is much less noticeable than a real body in the space beside you. The physical body plays an enormous role in awareness and shared behavior. Our own studies of different embodiments on digital tabletops have shown that there are huge differences between physical bodies and graphical representations of bodies.

For example, people are much more aware of a physical arm moving across the shared workspace, because of proxemics conventions about touch avoidance, than they are of a digital photographic representation of the same arm whose movement is controlled by a mouse (Doucette et al. 2015, 2013). People have a clear and detailed understanding of where other people's bodies are in a shared space, what they are doing, and what they are going to do next—all of the main elements of workspace awareness. We have built up this ability because of the need to avoid conflict—we need to know where other people are in order to avoid the social awkwardness of bumping into them, and the impoliteness of causing extra work or interference for them.

In our studies of digital embodiments, we tested whether people would behave the same way around digital representations as they do around real bodies—for example, would they avoid touching the other person (or embodiment), would they avoid occluding the other person's view with their arm (or virtual arm), or would they avoid crossing arms (or embodiments). The results were clear, and striking. Whereas people are extraordinarily aware of real bodies, they take almost no notice of other people's digital embodiments. This result was replicated in several situations, and did not change even when we made the embodiment more obvious, or even caused a buzzer to vibrate whenever the embodiments touched. The effect was also accentuated when participants were distributed—when the other participant was not in the room, people essentially ignored the digital embodiment (a live video arm).

This means that designing a system to support the kinds of subtle awareness that Tenenberg points to in the pair programming example may be extremely difficult. The awareness of another person's gaze direction, which is so easy to accomplish in a co-located setting, may be nearly impossible to support in a distributed tool. Returning to our prior gaze awareness example, a marker on the screen to indicate where another person is looking carries none of the weight of a real person turning their head to look at a different part of the screen. Thus we suspect that these designed awareness cues would go largely unnoticed and unused. This is not to say that new practices and social protocols could not arise around such artificial awareness cues. It is possible that people will become used to eyegaze markers and will start using them to support new kinds of rich awareness. However, this is a much different research question than what Tenenberg proposes.

8. Privacy, plausible deniability and mediating distraction

Another issue in implementing we-awareness concerns the loss of privacy. For example, the Video Wall system mentioned earlier supplies reciprocity across two lounges, but comes at the cost—and risk—of privacy incursions (especially because reciprocity is not guaranteed in that system). This means that participants must be willing to give up some of their privacy in order for others to acquire awareness information. This is not always the case. For example, following Fish et al. (1990)'s video wall work, Jancke et al. (2001) used video to connect several kitchens across departments in a research institute. Yet various users of those kitchens were not receptive to this, where some strongly felt that their privacy was violated (e.g., by distant viewers observing and interrupting conversations in process). This was in spite of the presence of controls to turn the system on or off.

The loss of *plausible deniability* is also tied into we-awareness. In essence, there are many times when people do not want others to know about their presence or activity. For example, Nardi et al. (2000) describe the case of instant messengers, where a person receiving a message may not want others to know that they are actually present (e.g., that they may not want to respond). They can plausibly deny their presence even if the system includes on-line indicators, because these are just estimates of their presence. In phone conferencing systems, some participants mute their microphones so that others don't know that they are actually reading email. Two-way video systems, in contrast, do not provide the opportunity for plausible deniability, as all know who is present and attending the conversation (unless one positions their body off-screen). Similarly, a shared digital workspace (including overviews) show the activity status of all, which means that a person who is not contributing will be noticed. They cannot plausibly deny their lack of activity or involvement.

Finally, a further limitation on we-awareness is that it can become distracting. Consider the real world case, where a person wants to talk to a person in a nearby office. She walks towards his office door but hears that he is talking. A quick peek towards the door confirms that he is busy on the phone, so she decides not to disturb them. Thus we see an example where privacy (and awareness) is self-regulated (Boyle and Greenberg 2005). Yet implementing these subtleties is difficult to do in groupware. For example, most video-based groupware systems are either fully on or off. They would open the audio channel at full volume and display the face of the person peeking into the room, thus making that peeking action highly salient and ultimately defeat the purpose of checking on one's availability without disturbing them.

In summary, designing for we-awareness is non-trivial both for technical reasons and because people may not want others to "know that you know that I know" etc. The problem is that we-awareness makes the assumption that remote collaborates all want mutual awareness. Yet this assumption is often incorrect: unlike face to face encounters, remote collaborators are in different environments and contexts, where participants can have quite different (and unbalanced) desires for we-awareness.

9. Conclusions

Tenenberg's article raises some valuable possibilities for the design of distributed awareness systems. We and other researchers have been thinking about supporting these kinds of shared behavior (although not calling it we-awareness) for many years. Here we have argued that recognizing the concept of we-awareness, and showing instances of how it differs from individual awareness, is an important but limited contribution. Knowing that we want to support the subtle and interdependent forms of shared work and we-awareness that we see in the real world does not, unfortunately, tell us how to build systems that can allow fully equivalent kinds of work and we-awareness over distance. We identified several potential roadblocks in doing this.

However, we are far from discouraged. We saw how simple techniques, such as telepointers and video, can partially supply the needed information, and that other richer and exciting techniques have already been developed. There is much to be done in studying the way that collaboration evolves in distributed contexts, and we are certain that the knowledge gained will inspire new techniques. Indeed, our article is perhaps best view as a cautionary tale: while we stress that we-awareness cannot be completely solved through simplistic solutions, we believe that the very complexity of this problem can lead to thrilling research and breakthroughs.

Acknowledgments

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada.

References

Boyle, M. and S. Greenberg (2005). The Language of Privacy: Learning from Video Media Space Analysis and Design. *ACM Transactions on Computer-Human Interaction - ACM TOCHI*, vol. 12, no. 2, pp.328–370.

Clark, Herbert H., and Susan E. Brennan (1991). Grounding in communication. In L. B. Resnick, J. M. Levine, and S. D. Teasley (Eds.): *Perspectives on socially shared cognition*, vol. 13, pp. 127–149. American Psychological Association.

Dix, A., J. Finlay, G. Abowd, and R. Beale (1993). *Human-Computer Interaction*, Prentice Hall.

Doucette, A., C. Gutwin, and R. Mandryk (2015). Effects of Arm Embodiment on Implicit Coordination, Co-Presence, and Awareness in Mixed-Focus Distributed Tabletop Tasks, *GI'15: Proceedings of the 41st Graphics Interface Conference*pp. 131–138. Canadian Information Processing Society.

Doucette, A., C. Gutwin, R. Mandryk, M. Nacenta, and S. Sharma (2013). Sometimes when we touch: how arm embodiments change reaching and collaboration on digital tables. *CSCW '13: Proceedings of the 2013 ACM Conference on Computer-Supported Cooperative Work*, pp. 193–202. New York: ACM Press.

Dourish, P., and V. Bellotti (1992). Awareness and Coordination in Shared Workspaces, *CSCW'92: Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work*, pp. 107–114. New York: ACM Press.

Fish, R., T. Kraut, and B. Chalfonte (1990). The VideoWindow system in informal communication. In *CSCW'90: Proceedings of the 1990 ACM conference on Computer-Supported Cooperative Work*. New York: ACM Press, pp. 1–11.

Greenberg, S., C. Gutwin, and M. Roseman (1996). Semantic Telepointers for Groupware. In *OZCHI'96: Proceedings of the Sixth Australian Conference on Computer-Human Interaction Hamilton, New Zealand, November 24–27.* IEEE Computer Society Press, pp. 54–61.

Greenberg, S., M. Roseman, D. Webster, and R. Bohnet (1992). Human and technical factors of distributed group drawing tools. *Interacting with Computers*, vol. 4, no. 1, pp. 364–392.

Gutwin, C. (2001). The effects of network delays on group work in real-time groupware. In *ECSCW'01: Proceedings of the European Conference on Computer Supported Cooperative Work*, pp. 299–318. Dordrecht: Kluwer Academic Publishers.

Gutwin, C. and S. Greenberg (2002). A Descriptive Framework of Workspace Awareness for Real-Time Groupware. *Computer Supported Cooperative Work: The Journal of Collaborative Computing*, vol. 11, no. 3–4, pp. 411–446, Special Issue on Awareness in CSCW.

Gutwin, C. and S. Greenberg (1998). Design for Individuals, Design for Groups: Tradeoffs between power and workspace awareness. *CSCW'98: Proceedings of the 1998 ACM Conference on Computer-Supported Cooperative Work*, New York: ACM Press.

Gutwin, C., S. Greenberg, and M. Roseman (1996). Workspace Awareness in Real-Time Distributed Groupware: Framework, Widgets, and Evaluation. *People and Computers XI (Proceedings of HCI '96)*, Springer-Verlag, pp. 281–298.

Hall, E.T. (1966). *The Hidden Dimension*, 1st ed. Garden City, N.Y.: Doubleday.

Hutchins, Edwin. (1991). *Cognition in the Wild*. Cambridge, USA: MIT Press.

Ishii, H., and M. Kobayashi (1992). ClearBoard: A Seamless Medium for Shared Drawing and Conversation with Eye Contact, *CHI'92: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 525–532. New York: ACM Press.

Jancke, G., G. Venolia, J. Grudin, J. J. Cadiz, and A. Gupta (2001). Linking public spaces: technical and social issues. In *CHI'01: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 530–537. New York: ACM Press.

Kendon, A. (1990). *Conducting Interaction: Patterns of Behavior in Focused Encounters*. Cambridge: Cambridge University Press.

Lee, J.J. (1990). Xsketch: A multi-user sketching tool for X11, in *Proc. Conf. Office Information Systems*, (Boston, April 25–27), pp. 169–173.

Luff, P. and C. Heath (1998). Mobility in collaboration, in: *CSCW'98: Proceedings of the ACM Conference on Computer Supported Cooperative Work*. New York: ACM Press, pp. 305–314.

Nardi, B., S. Whittaker, and E. Bradner. (2000). Interaction and outeraction: instant messaging in action. In *CSCW"00: Proceedings of the 2000 ACM Conference on Computer-Supported Cooperative Work*, pp. 79–88. New York: ACM Press.

Segal, L. (1995). Designing Team Workstations: The Choreography of Teamwork, In P. Hancock, J. Flach, J. Caird and K. Vicente (eds) *Local Applications of the Ecological Approach to Human-Machine Systems*, pp. 392–415, Hillsdale, NJ.: Lawrence Erlbaum.

Stotts, David, Jason McC. Smith, and Karl Gyllstrom (2004). Support for distributed pair programming in the transparent video facetop. In *Proceedings of the Fourth Conference on Extreme Programming and Agile Methods— XP/Agile Universe*, pp. 92–104. Berlin Heidelberg: Springer.

Tang, J. (1989). *Listing, Drawing, and Gesturing in Design: A Study of the Use of Shared Workspaces by Design Teams*, Ph.D. thesis, Stanford University, Stanford, CA, 1989.

Tang, J. (1991). Findings from Observational Studies of Collaborative Work, *International Journal of Man–machine Studies*, vol. 34, no. 2, pp. 143–160.

Tenenberg, J., W.-M. Roth, and D. Socha (2016). From I-awareness to we-awareness in CSCW. *Computer Supported Cooperative Work (CSCW),* vol. 25, no. 4–5. [Special issue: *Reconsidering 'Awareness' in CSCW*].