Range: Exploring Implicit Interaction through Electronic Whiteboard Design

Wendy Ju, Brian A Lee, & Scott R Klemmer
Stanford University
353 Serra Mall, Stanford CA 94305
{wendyju, balee, srk} @stanford.edu

ABSTRACT

An important challenge in designing ubiquitous computing experiences is negotiating the transition between explicit and implicit interaction, such as how and when to provide users with notifications. While the paradigm of implicit interaction has important benefits, it is also susceptible to difficulties with hidden modes, unexpected action, and misunderstood intent. To address these issues, this work presents a framework for *implicit interaction* and applies it to the design of an interactive whiteboard application called Range. Range is a public interactive whiteboard designed to support collocated, ad-hoc meetings. It employs proximity sensing capability to proactively transition between display and authoring modes, clear space for writing, and cluster ink strokes. We show how the implicit interaction techniques of *user presentation* (how users implicitly indicate what they are doing), *system presentation* (how systems indicate what they are doing), and *override* (how users can interrupt or stop a proactive system action) can prevent, mitigate, and correct errors in the whiteboard's proactive behaviors. These techniques can be generalized to improve the designs of a wide array of ubiquitous computing experiences.

Author Keywords

Implicit interaction, foreground/background, ambient, proactive

ACM Classification Keywords

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*input devices and strategies, interaction styles*.

INTRODUCTION

One of the defining traits of ubiquitous computing is the pursuit of invisibility. Different camps of interface researchers and designers have taken different tacks towards this elusive goal. This is evidenced by the amazing diversity of ubiquitous computing genres which cite Mark Weiser's "Computer for the 21st Century" [36] as a genesis—ambient displays, tangible user interfaces, context-aware computing, attention-sensitive interfaces, just to name a few. In light of this great variety of approaches towards invisibility, it is useful to keep in mind that invisibility, as championed by Weiser, is not so much about staying beneath notice as enabling seamless accomplishment of task.

In their paper "Making Sense of Sensing Systems: Five Questions for Designers and Researchers," Bellotti, *et al*. point out that ubiquitous computing systems are particularly susceptible to problems of unintended actions, undesirable results, and difficulty detecting or correcting mistakes [1]. This occurs because of the high potential for miscommunication when the interaction between the computing system and the user occurs beneath the user's notice or without the user's initiative. Since invisibility is about enabling seamless accomplishment of desired tasks rather than staying beneath notice, we propose that it is important to understand how to design transitions between explicit and implicit interaction, so that users can make requests, anticipate actions, and make corrections even in situations where they have limited attentional, cognitive, or physical bandwidth for interaction.

The goal of this paper is to explore the range of ways that designers can establish shared understanding between user and system without using keyboard, mouse, or stylus for input, and without using dialog boxes for output. To accomplish this task, we present a framework for implicit interaction, as a well as an implementation of a ubicomp whiteboard application, from which we extrapolate general purpose implicit interaction techniques. It is our hope that this framework and illustration will help to add implicit interaction design to the range and repertoire of ubicomp interaction designers.

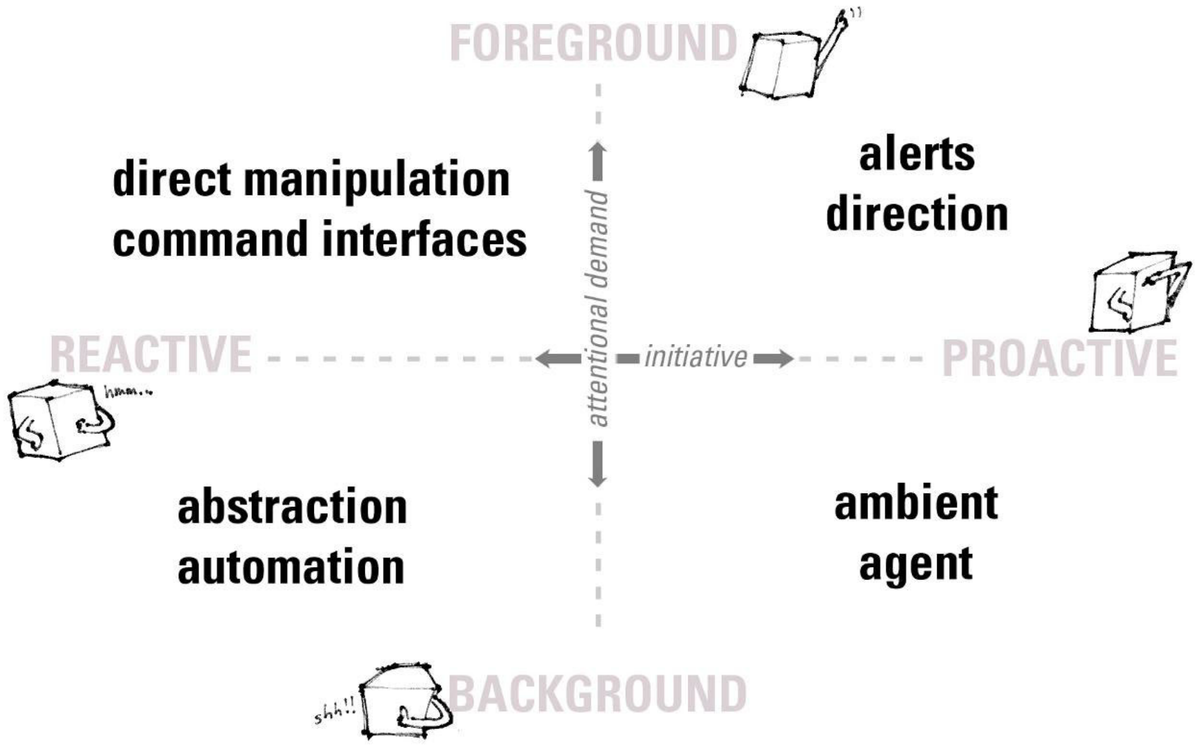

Figure 1. The Implicit Interaction Framework is based on two axes: the level of attentional demand and the balance of initiative between the user and the system. This framework provides a domain-independent characterization of an interaction's implicitness.

IMPLICIT INTERACTION FRAMEWORK

We define implicit interactions to be those based on implied rather than explicit input and output. To understand this better, it is useful to consider what makes explicit interaction explicit. In explicit interaction, the user issues commands—for instance, through a mouse or keyboard command—and receives overt feedback.

One way that interactions can be non-explicit is if the exchange takes place outside the attentional foreground of the user—for instance, when the computer auto-saves your files, or filters your spam. The other way that interactions can be non-explicit is if the exchange is initiated by the system rather than by the user. This occurs in traditional interaction—when the computer alerts to user about new mail, for instance, or when the computer displays a screensaver—as well as ubiquitous computing interaction. While it may seem counter-intuitive that we define these sometimes attention-grabbing interfaces to be implicit, we note that "pushed" information is based on an implied pull.

The implicit interaction framework (see Figure 1) maps interactions against these axes: *attentional demand* of the system on the user, and the *initiative* demonstrated by the system. Our intent in describing the spectrum of interaction is not so much to champion any point on the continuum as a sweet spot, but rather to extend the range and repertoire of interaction designers by calling attention to the different possibilities along the spectra of each axis.

Attentional Demand

Attentional demand is the degree of cognitive and perceptual load imposed on user by the interactive system [26]. *Foreground interactions* require a greater degree of focus, concentration and consciousness, while *background interactions* do not make such demands, and in fact, elude notice [3].

Attentional demand does not correspond easily with any particular metric, in part because attention is very complex [4]. Any comprehensive definition needs to account not only for the *load* on the resource of cognition [22], but also for *spatialization* (when something is in the center versus the periphery of one's notice) [38], *breadth* (when attention is focused on a single stimulus or many), and *gestalt* (whether attention is devoted to the abstracted whole or the individual parts) [34]. The other challenge that researchers have identified is that attention—by its very nature—can be challenging to evaluate directly [26].

Interaction designers commonly manipulate attentional demand by adjusting the perceptual prominence of objects, often implicitly, through visual organization techniques, such as contrast, hierarchy, and weight [40]. Demand may also be choreographed through more dynamic means, such as pointing, (e.g. calling attention to an object through by gesturing at it) or placing (e.g. calling attention to an object through its prominent placement) [7]. Still another way to affect the degree of attention demanded is through abstraction and chunking, wherein small interactions are combined into a larger whole [5].

Initiative

Initiative is an indicator of which party is initiating and driving an interaction. Interactions initiated by the user are *reactive*, whereas interactions initiated by the system are *proactive* [33]. When considering the level of initiative as a design resource, one should take into account both the certainly of the need for action, and the costs involved if the action taken was done so incorrectly. A spell-check feature which checks words as they are being written is more proactive than one that is initiated by a user at the end of writing a letter, because the post-facto spell-check process is started and run by the user, as opposed to started and run without the user's intent; a spell-check that auto-corrects is more proactive still, and a spell-check whose auto-corrections cannot be reverted is most proactive (and most annoying).

Designers can manipulate the proactivity and reactivity of a designed interaction by dictating the order of actions—does the system act first, or wait for the user to act?—as well as by choosing the degree of initiative—does the system act, offer to act, ask if it should act, or merely indicate that it can act? In reactive systems, does the user merely make a high-level request, or does he or she need to perform sustained and detailed actions to accomplish the task? Designers can also control initiative by affecting the certainty of the need for an action or by adjusting the potential cost of error for the action.

Types of Interactions

The following are descriptions of canonical interactions for each quadrant. For illustration, we cite examples of each from the world of traditional desktop computing, but will use instances of each in ubiquitous computing throughout the rest of the paper:

Reactive/foreground

Interactions take place explicitly and at the user's command. Users are given explicit and detailed oversight over actions and feedback on results. Such interactions are appropriate when the interaction is the primary task and is controlled by a knowledgeable user. Normal GUI interaction would fall into this quadrant.

Reactive/background

Interactions occur in response to user actions or external stimuli, but feedback is generalized or hidden from the user (abstraction). Such interactions can spare the user from the nitty-gritty details of a task or help perform routine tasks automatically with little or no user oversight (automation). The "auto-save" on a typical word-processing program exemplifies this type of interaction.

Proactive/foreground

Interaction takes place in the attentional foreground, but involves greater urgency on the part of the object. The object may provide unsolicited information (alerts) or guide the interaction by instructing the user what to do (direction) These interactions are typical in reminder and tutorial scenarios. The "You've got mail" sound and bouncing icon in typical mail program is an example of proactive/foreground interactions.

Proactive/background

The object anticipates what to do and performs with low oversight or input. Usually used for tasks where the cost of error is low: for instance, pre-fetching data, or modeling preferences. It can also enable critical tasks that the user is somehow unable to perform, like alerting the police when someone is intruding into one's home. A common example is the computer screensaver.

While it is possible to speak of the implicitness or explicitness as genres of interaction, it is also important to recognize the potential offered by transitioning between implicit and explicit interactions in response to the dynamics of the interaction. Explicit interaction is bound to have some implicit components, and any implicit interaction is likely to have explicit ones. The techniques explored later in this paper illustrate how, why and when to transition from one type of behavior to another in the course of a larger interaction.

In the following sections, we will discuss our selection of interactive whiteboards for our exploration, review related work on implicit interactions and whiteboards that informed our framework and interaction design, outline the specific design our electronic whiteboard system, Range, and discuss the implicit interaction techniques illustrated by our implementation.

INTERACTIVE WHITEBOARDS AS A TESTING GROUND FOR IMPLICIT INTERACTION

The ephemeral nature of whiteboard ink allows users to share ideas quickly—and just as quickly, to amend those ideas. The improvisational quality of whiteboard use is a good match for the provisional ideas that are generated in informal design meetings, when people are more concerned with entertaining possibilities than communicating fact. The ubiquity of whiteboards in dedicated design spaces (such as war rooms, and project rooms) and informal meeting spaces (such as offices, break rooms, and hallways) is a testimonial to the utility of the whiteboard to designers everywhere.

The utility and ubiquity of whiteboards makes them an appealing platform for computational enhancement. However, the attractive aspects of whiteboards are inextricably linked to the factors that also make them challenging to augment. The shared, public nature of such whiteboards means that the interface must succeed for walk-up use, and the focus on quickly sharing ideas means that any services provided must have a low threshold to entry and minimal attentional overhead.

The issues associated with whiteboards are like those of many ubiquitous computing situations: interactions are often transient and needed on-demand; and the users are often distracted and untrained. We have introduced this whiteboard as an implementation that helps manifest the opportunities for and challenges with implicit design in ubiquitous computing.

RELATED WORK

This paper draws on related work in three areas: the framing of the interaction styles, workplace studies of whiteboard usage, and the design of electronic whiteboards.

Design Frameworks for Implicit Interaction

The framework laid out in this paper builds on Buxton's foreground/background model [1]; in it, Buxton distinguishes the foreground interactions—to paraphrase, intentional activities that take place in the fore of human consciousness—from background interactions, such as a light automatically turning on when you enter a room—which take place in the periphery of consciousness. This model identifies the same attention and initiative used in our framework, but assumes the two are inherently linked. Actions initiated by the user are assumed always to be taken with intent; actions taken by the system are assumed to take place in the periphery. Our framework extends Buxton's framework by decoupling attention and initiative into separate axes. Buxton's foreground corresponds to our reactive/foreground quadrant, and his background corresponds to our proactive/background.

Horvitz *et al.* [17] present a related model for notification displays. It uses an economic model of use attention, and determines the expected utility of presenting users with notifications, based on the level of attentional cost to the user and the expected value of the information. This model traverses the same territory as the right side of our framework, ranging from proactive/foreground to proactive/background. Its use of uncertainty as a measure of proactivity guided our framework's formulation of initiative. This model is ideally suited to help computers make dynamic determinations about the right way to deliver a piece of information. It provides less guidance, however, to the interaction designers developing the different methods the computer might eventually chose from.

Workplace Studies of Whiteboard Usage

The Flatland whiteboard interface [26] was based on informal observations of whiteboard use in office settings. Researchers observed that office use of whiteboards was characterized by thinking and pre-production tasks, everyday content (such as task lists, sketches, and reminders), clusters of content (both persistent and short-lived), and a transitioning between semi-public to personal use. Our design of Range builds on the observations that Flatland is based on. It includes features to supporting range of use from display to whiteboard, freeing up space for drawing, clustering strokes of ink. The major departure in our explorations is the use of distance sensing as input for these features, and the avoidance of meta-strokes or other explicit techniques.

Longitudinal studies of student engineering design teams working on multi-month projects by Ju, *et al.* [19] found that engineers engaged in informal meetings would cycle between phases of drawing and analysis; these changes corresponded with changes in their physical proximity to the whiteboard. Users would stand close to the board when they were writing, further back when discussing written artifacts in detail, or further back still when engaging in meta-discussion. They also found that input was initially free-form, but that meeting participants would often close their meetings by performing post-facto structuring on previously generated sketches, drawing borders, lines, and arrows to explicltly group or relate elements on the board.

Our observations of whiteboards, based on photos taken around campus in several departments, indicate that sketches on the board can generally be categorized as either "read-only" or "write-only." What we called "read-only" were messages that were meant to persist, and changed infrequently: phone numbers of colleagues, lists of upcoming deadlines. Sketches that were "write-only" were usually generated in informal meetings, and were infrequently referenced after their initial creation. Regardless of field, people implicitly placed information that is meant to be static or saved along the edges of the board, saving the center of the board for temporary and speculative work. This finding validates location of information on the board as a crucial context variable.

Design of Electronic and Augmented Whiteboards

Electronic whiteboards emerged out of the ubiquitous computing research at PARC, and their goal of computing by the inch, foot, and yard [36]. PARC's LiveBoard [9] was a rear-projected electronic whiteboard that afforded pen-based input through infrared-emitting styli. Tivoli [28], the LiveBoard's whiteboard application, introduced a set of interaction techniques for creating and manipulating ink-based documents, and supported input from multiple pens simultaneously. Ink strokes were stored as grouped vector objects, and the system introduced gestures for the selection, grouping, and manipulation of ink content.

Subsequent research [24, 25] explored the use of implicit structure in the user's ink—here the term "implicit" was used to describe structures (such as lists, drawings and tables) whose spatial layout has meaning that were intended and perceived by the user, but not to their system "because it is not defined or declared to the system."[24] In grappling with whether such implicit structures should be exploited by the electronic whiteboard as input, or if input should be wholly freeform, the PARC researchers introduced the first

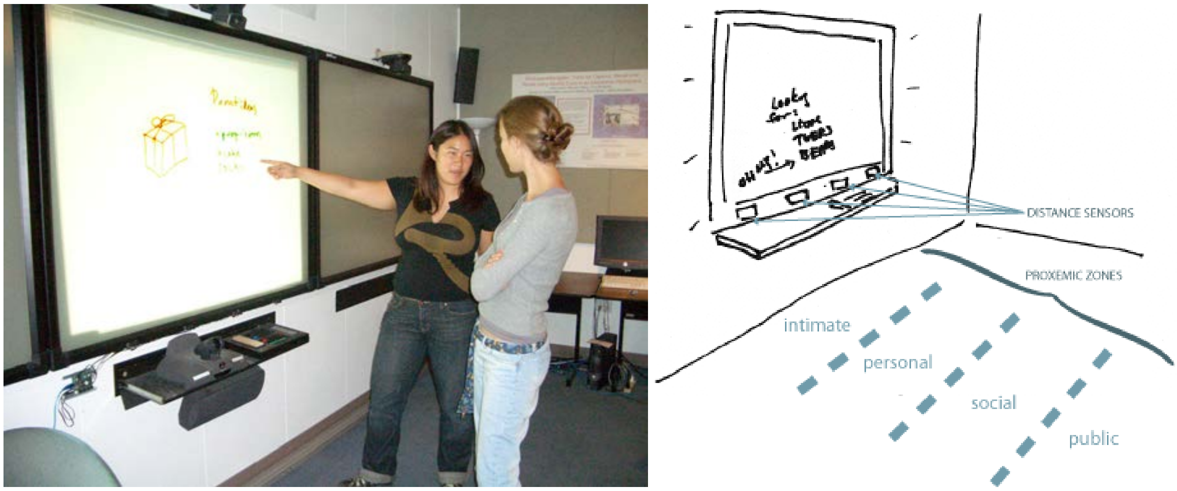

Figure 2. Physical setup of Range (left), with diagram of interaction zones (right).

pen-based interface to decouple *recognition* (having the system create an internal hypothesis of the user's intended structure) from *transformation* (having the system in turn modify the representation of the user's data based on its belief about the structure). These ideas were extended upon in SILK [21] and subsequent informal user interfaces, e.g., [23] [20]. This selective and timed presentation of what the system believes is used in our design of Range.

Recent work on electronic whiteboards has focused on incorporating aspects of the user's physical context in whiteboard use into the interaction. Research on using paper and digital artifacts with an electronic whiteboard [20], on using pen-based command techniques for high-resolution displays [14] and on physical gestures and tokens for specifying behaviors [32] begins to realize Weiser's vision of computation that is embedded into the fabric of everyday life. Current work in ambient interfaces is also exploring the understanding of the user's physical context as an implicit input in the domain of large interactive public displays. Both Prante, *et al.*'s Hello.Wall [29] and Vogel & Balakrishnan's interactive Ambient Public Displays [35] stand out for explicitly noting the proxemic relationship between the physical distance between multiple users and the display, and using it to modify the contents of the display accordingly. Our whiteboard design draws on similar proxemic relationships between users and whiteboards, but the implicit meaning of the being close or far from each board differs because whiteboards are intrinsically meant for writing as well as display.

This paper offers two contributions beyond this work in electronic whiteboard interactions. The first is that it provides a richer framework for describing and designing implicit interactions; the second is that it is oriented towards broadening the range of interactive technique rather than the enriching the pool of whiteboard features.

THE RANGE WHITEBOARD

To illustrate how implicit interaction techniques can be used to prevent, mitigate and correct the problems of proactivity in the area of whiteboard interaction, we designed an interactive whiteboard named Range, which uses infrared distance sensors to subtly and proactively interact with informal meeting participants.

Implementation

Range was implemented using a combination of pre-existing hardware and software tools and technology.

Platform

The Range whiteboard prototype employs a rear-projection SMART Board containing an SXGA+ resolution projector (1400x1050) and a Windows XP PC. Four SHARP GP2Y0A 150 cm analog distance sensors were mounted to the front of the board, and connect to the PC over USB via the d.tools hardware and libraries [16]. The software component of Range was written in C# using the Microsoft Tablet PC SDK and the SMART Board SDK.

Physical Interaction Design

The region in front of the board into four zones, which we called intimate, personal, social, public in reference to proxemics pioneer Edward T. Hall's distance zones [15]. We defined the intimate zone to be the region in which users stand to write at the board, testing with multiple users to increase the robustness of the zone definitions. The personal zone was set further back, at a distance (>15 inches back) where users were not "at" the board, but could easily reach the board for pointing and text manipulation. The social zone (>25 inches back) was out of touching distance from the board but in easy viewing distance of the board. The public zone comprises the distance beyond the social (> 40" back).

The operational zone was based on the user closest to the board; studies [19] indicate that this is usually the person with the pen and thus the person "driving" the interaction at the whiteboard.

Operation

The SMART Board uses a pen tray with four colored styli and an eraser. Strokes made with the styli make ink strokes of the corresponding color on the board, and strokes made with the eraser remove marks intersected by the erase stroke. Input on the capacitive board is presumed to be made by the users' fingers if all the styli are in the tray; such finger input is used to select and move ink strokes and clusters.

Features

We implemented three features in Range that demonstrate implicit interaction techniques: an ambient screensaver, automatic space clearing, and automatic ink stroke clustering.

We modified the SMART Board operation so that inputs issued when the user is in the personal zone are read as select and move operations even if the pen is out of the tray; this seems more natural to users and lessens the instances of erroneous input.

Transition from Display to Drawing Surface

While there is a one-to-one correspondence between the display surface and drawing surface of physical whiteboards, the contents of digital whiteboards may change dynamically, presenting digital material not explicitly drawn or placed on the board.

When users are not engaged with Range, the whiteboard switches to screensaver mode, overlaying the existing whiteboard contents with a transparent blue backdrop and a stream of digital images of interest to the user. Our implementation uses snapshots of previous whiteboard states and other photos of interest from a group Flickr [11] account.

As a user approaches a Range whiteboard in screensaver mode, the backdrop fades and the displayed screensaver content floats off to one side, allowing the user to re-engage the whiteboard contents beneath. If the user touches the departing screensaver content, it stops and becomes selected so that the user may move it to some place on the whiteboard of his or her choosing.

Making space

As the designers of Flatland observed, whiteboards are not merely ephemeral objects: people leave drawings or notes on the board in order to provide shared reference for groups [26]. However, a whiteboard full of writing can present problems. Our observations of whiteboard usage suggest that users are hesitant to erase work, for fear for removing something important. Copying content to another surface takes time, time that may kill a serendipitous, free-flowing conversation.

To address this problem, Range moves board contents out of the center when it senses a user approaching, clearing a space so that the user immediately has a blank space in which to write. Data on the edges of the board are not affected during the board-clearing maneuvers.

Figure 3. Screensaver mode. When not in active use, Range displays photos of interest overlaid on top of any content on the board.

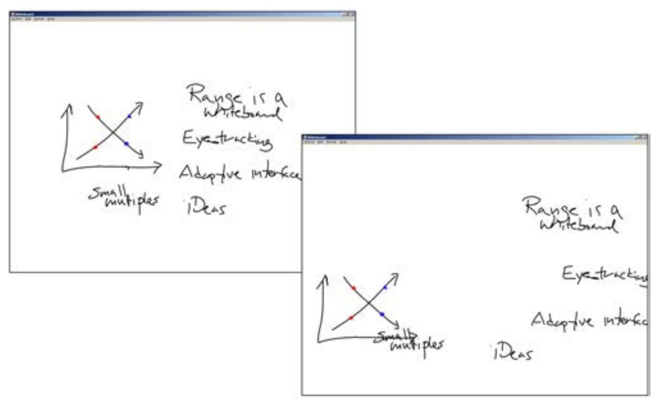

Figure 4. Making space. *Left:* Whiteboard before user approaches board. *Right:* Whiteboard after user walks up to board. Space has been cleared in the center for the user to write by moving existing text off to the periphery.

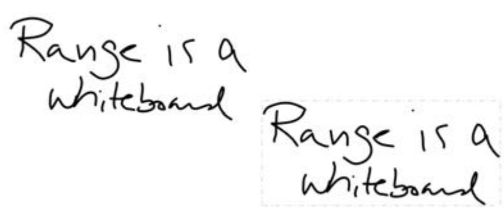

Figure 5. Clustering. While strokes are invisibly clustered in writing mode (left), feedback about clusters is displayed when users are standing in the personal zone (right).

Clustering Ink Strokes

In order to move text and graphics around while maintaining coherency of the sketches, the underlying system needs to have some conception of the semantic units of whiteboard contents. To achieve this, we have implemented a simple form of stroke clustering, using the stroke's timestamp (time of creation) and location on the board (estimated by its bounding box). As strokes are created, the Range system runs a clustering algorithm in the background: strokes that were either created at the same time (temporal locality) or that are close together on the board (spatial locality) are clustered together automatically.

Users are given feedback about the clusters, by way of dotted light-gray bounding boxes, when they are located in the personal zone. Users manipulate clusters as an atomic unit: selecting one stroke in a cluster selects them all by default, and moving a stroke in a cluster moves the whole cluster. Users may override the automatic clustering by lasso selecting one or more strokes, which puts all of the selected strokes into a new cluster.

IMPLICIT INTERACTION TECHNIQUES

The design of the three aforementioned features illustrates the implicit interaction techniques of *user presentation*, *system presentation* and *override*. These features do not necessarily form an exhaustive set of implicit interaction techniques, but they provide characteristic solutions to interaction problems typical of ubiquitous computing [1]. The sociological term "presentation" is used to describe the expression an interactant gives and gives off.[13]

User Presentation

User presentation is how users indicate what they are doing or would like to have done to the system. It differs from traditional conceptions of input in that user presentation is not necessarily intentional, direct, or explicit. The term presentation comes from sociology

Writing related text close together is an example of user presentation in the Range application. The ink strokes are implicitly related to one another by their proximity in space and time. Designating input modes by where users are standing in space or determining the stickiness of specific clusters based on where they are located in space are other instances of user presentation. Users implicitly control the board through placing or marking behaviors that indicate to the whiteboard what action is desired.

Since user presentation is a signal to the system—often not a deliberate one—the system should provide some sort of feedback to validate the inferred input. We discuss mechanisms for error-handling in the subsequent override section. This validation should occur before further action is taken based on this action. In the ink stroke clustering design, for instance, validation occurs when Range outlines the clusters as the user steps back. This moment is opportunistic because it follows the period when the user is actively writing, and should not be interrupted, and usually precedes the period when the clusters of text might be automatically selected and moved.

To the Framework

Looking at user presentation against the implicit interaction framework, we see that the system's recognition of close writing, initially occurs in reactive/background. However, prior to its taking action on anything based on this ambiguous input, the system should verify the inferred signal; this verification is necessarily proactive/foreground, because it is initiated by the system, and of course the user needs to be able to see it. So this trajectory, from the lower left quadrant of the implicit interaction framework to the upper right quadrant, is typical of effective user presentation.

Variations

Numerous variations exist on the general theme of user presentation. Pointing and placing [7], for example are common ways that people use to present information implicitly. A user's pointing at a cluster presents to Range which object to select and possibly move; the highlighting of the selected object provides the verification step discussed above. The placement of objects also presents implicit information about the "read-only" or "write-only" status of the objects on the edge of the screen. The physical location of people before the whiteboard present an implicit indication of what mode they are operating in at the whiteboard. When users deliberately stand in a certain location, for instance, to hide the cluster outlines, this is a form of presentation known as avoidance [12].

Design Notes

When designing user presentation interactions, it is useful to perform fieldwork to understand what meaning exists for different placement, spacing or marks. Alternatively, the designer can invent meanings but then needs to communicate them very clearly. Grocery store shopping counters [7], for instance have been designed to confer

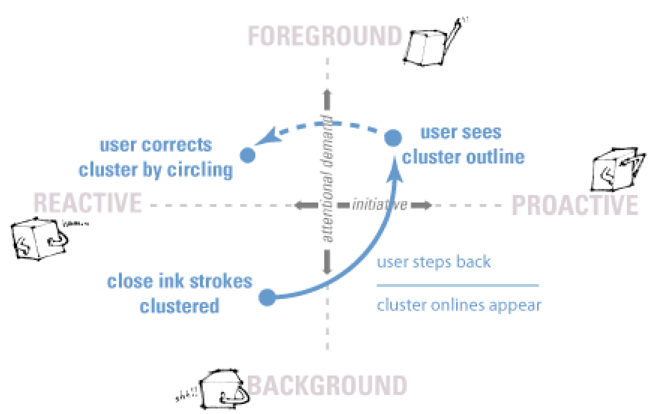

Figure 6. The trajectory of User Presentation (solid line) and Override (dotted) used in clustering, as shown in the Implicit Interaction Framework

special meaning to the objects placed on the counter, but the design is not arbitrary. The counters are located so that the placement of objects is in the foreground of both the shopper and the clerk, and so that the counter helps to obscure those objects that are not part of the financial transaction—the bag from the previous store, or your handbag, for instance.

System Presentation

System presentation is how the system shows the user what it is doing, or what it is going to do. This differs from the traditional conception of output in that it is not necessarily symbolic, overt, or immediate. When the system "presents," it implicitly draws the attention of the user, so that it can make a suggestion or request oversight.

On the Range whiteboard, the outlines for the clusters appear when the user steps back into move/selection mode. This act serves as user presentation, to show the user how the computer grouped his or her ink strokes, and also as system presentation, as the outlines indicate the change in mode to the user by showing it the units that could be selected or moved. System presentation is also evident when the "screensaver" images of the ambient display float off the screen rather than simply disappearing, and in the slow animated movement of the clusters when the whiteboard is clearing space; the motion of the images helps the users understand what the system is doing.

To the Framework

This is a way for the system to let the user know what it's doing. Its trajectory goes from the proactive/background quadrant to the proactive/foreground quadrant.

In system presentation, the system's proactive/foreground notifications are responded to with the user's explicit "undoing" of these actions. This trajectory goes from the proactive/foreground to the reactive/foreground quadrant. Re-clustering or "catching" moving screensaver pics are examples.

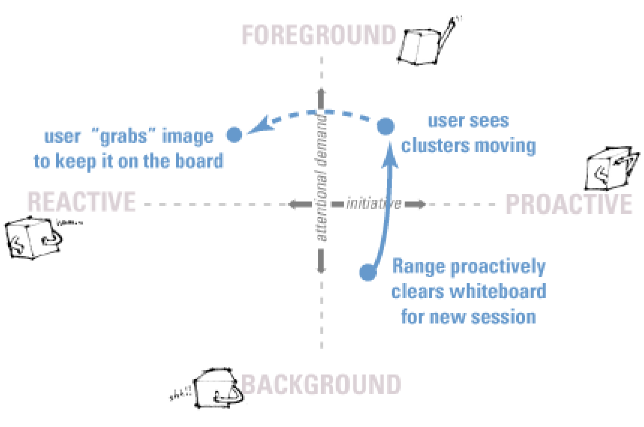

Figure 7. The trajectory of System Presentation (solid line) and Override (dotted) used in making space.

Variations

Specific variations of system presentation techniques include overt preparation and feed-forward. Overt preparation, occurs when the system "shows" that that is preparing to take some action; these cues are generally read as an implicit offer (or threat). A doorman, for example, subtly offers to open the door for you by making a grand show of putting his gloved hand on the handle of the door. Feed-forward signals an impending action by presenting users with the projected outcome of an action it is going to take.

Design Notes

The most challenging aspect of designing presentations is understanding how users will interpret what is presented to them. It is possible to apply some design intuition here, based on what implicit understandings people use to present to one another. For instance, as a rule of thumb, small scale versions of an action (overtly leaning in the direction of the door) are implicitly understood as an offer or request to perform the full scale action (leaving). However, the design of system presentations requires testing with actual users to rule out false interpretations. Designing presentations for new actions also often requires several trials; users don't learn to anticipate an action until they have seen it occur several times.

Override

Override techniques enable users to interrupt or stop the system from engaging in a proactive action. This usually occurs after one of the previous two techniques (user presentation and system presentation) alert the user to some inference or action which is undesirable. Override differs from "undo" because it is targeted at countering the action of the system rather than reverting a command by the user.

The Range whiteboard demonstrates override capabilities in several places. In the transition between display and whiteboard, users are able to "grab" digital content to use it as part of the whiteboard contents. They are also able to stop the motion of objects which are being moved to make space in the center of the board. Users are also able to override the automatic clustering of the Range whiteboard by explicitly selecting and moving a cluster; if the system has erred in clustering the ink strokes, then the cost of manipulation or correction is no more than it would be without the auto-clustering feature.

To the Framework

In order to have an override, there needs to be an action to be overridden. When the override is preceded by a user presentation, the override is a correction of how the system interpreted (and reflected) the user's actions. When the override is a response to system presentation, it is a interruption of the presented action. The trajectory for override goes from the proactive/foreground quadrant to the reactive/foreground quadrant.

Variations

Common variations on override are preemption (for instance, when you cover your glass with your hand to indicate that you don't' want more coffee) and retraction (to overtly "cancel" a signal which may have prompted unwanted action.). "Blocking" behaviors—putting your hand in front of an elevator door to stop it, for instance—are a physical version of preemption.

Design Notes

Overrides are often the easiest method to design, because users expect to be able to override things (even though they are frequently disappointed). At the point that users see some unwanted action taking place, they try numerous ways of trying to override the action; it is merely a matter of designing the interaction so that the user's frantic override behaviors are registered as an input. It is possible for the designer to design in affordances for overrides—handles and edges, for example, that the user can grasp, or shields that the user can use to perform blocks.

IMPLICATIONS FOR DESIGN

Implicit interactions enable people to communicate subtly but efficiently. In applying implicit interaction to an interactive whiteboard, we show that implicit interactions are not necessarily the domain of some new genre of interactive objects. Implicit interactions are evident in the design of everything from automatic spell-checkers to interactive robots. By explicitly articulating how, when and why an interaction designer might use implicit interactions, we widen the designer's range in designing for challenging new domains such as ubiquitous computing.

Designers clearly intuit the need and invent solutions that take place within the implicit interaction space. However, the benefit of having the framework allows clever solutions developed for one domain to be generalized and applied analogously to another domain. For instance, let us say that we wanted to apply the ideas from user presentation to the spell-check feature of our word processors. People are often annoyed when the word processor marks properly spelled words as incorrect, or worse yet, auto-corrects them. Currently, most programs enable menu-based override of spell-check. We could build in ways for the user to present to the system, "I'm spelling this word this way deliberately"—a new, non-printing punctuation mark, perhaps, or sophisticated gaze recognition that can spot the dirty glare of a user signaling "don't you dare autocorrect me on this one.

The implicit interaction framework and techniques are ultimately meant to support better design practice by providing a common framework on which to compare analogous designs in disparate contexts. The framework allows designers to better analyze interaction sequences, and communicate about them and thus to more easily generate implicit interactions. The process of implicit interaction design with the framework continues to require originality and innovation on the part of the designer, but greatly enriches the pool of example techniques that he or she can draw on.

CONCLUSION

In this paper, we have explored implicit interaction by applying our implicit interaction framework to the design of an electronic whiteboard application, Range. Range's design is targeted specifically to the needs and practices of informal meeting participants, and yet the framework allows the interaction design techniques used in Range to be generalized to inform the design of implicit interactions in analogous domains.

This work provides a common basis for interaction designers to explore and share the range of implicit interactions and techniques. We provided a framework for better understanding the range of implicit interactions, and illustrated how implicit interaction techniques can be used to prevent, mitigate and correct the problems of proactivity in the area of electronic whiteboard design. The intent of this work is to provide interaction designers working a wide variety of domain- and task-specific ubiquitous computing systems with a framework that allows them to build on each other's of pattern and technique. This in turn can enable designers to better develop more sophisticated ways of implicitly interacting with systems in everyday life.

ACKNOWLEDGEMENTS

This research was supported by an equipment grant from Intel Corporation, a fellowship from the Intel Foundation, as well as a grant from the Wallenberg Global Learning Network. Thanks to Terry Winograd, David Akers and Leila Takayama for their assistance on the paper.

REFERENCES

1. Bellotti, V., Back, M., Edwards, K.W., Grinter, R. Henderson, R., Lopes, C. Making Sense of Sensing Systems: Five Questions for Designers and Researchers. In *Proc. CHI 2002*, ACM Press (2002), 415-422.
2. Bellotti, V., Rogers, Y. From Web press to Web pressure: multimedia representations and multimedia publishing. In *Proc. CHI 1997*, ACM Press (1997), 278-286.
3. Buxton, W. Integrating the Periphery and Context: A New Model of Telematics. In *Proc. Graphics Interface 1995*, (1995), 239-246.
4. Cavanagh, P. Attention routines and the Architecture of Selection. In *Cognitive Neuroscience of Attention*, Posner, Michael I, ed. Guilford Press (2004), 23-24.
5. Chase, W. G. and Simon, H. A. Perception in chess. In *Cognitive Psychology* 4(1973), 55-81.
6. Clark, H.H. and Brennan, S.E. Grounding in Communication. In *Perspectives on Socially Shared Cognition*, Resnick, L.B., Levine, J.M. and Teasley, S.D. eds., American Psychological Association (1991), 127-149.
7. Clark, Herbert H. Pointing and Placing. In *Pointing: Where Language, Culture and Cognition Meet.* Kita Sotaro, ed. Lawrence Erlbaum (2003), 243-268.

8. Cooperstock, J.R., Fels, S., Buxton, W., Smith, K.C. Reactive Environments. In *Communications of the ACM*, 40(9), ACM Press(Sept 1997), 65-73.
9. Elrod, S., Bruce, R., Gold, R., Goldberg, D., Halasz,F., Janssen,W., Lee, D., McCall, K., Pedersen,D., Pier, K., Tang, J., and Welch, B. Liveboard: A large interactive display supporting group meetings, presentations and remote collaboration. In *Proc. CHI 1992*, ACM Press (1992), 599-607.
10. Fass, A.M., Forlizzi, J., Pausch, R.. MessyDesk and MessyBoard: Two Designs Inspired by the Goal of Improving Human Memory. In *Proc. DIS 2002*, 303-311.
11. Flickr. *http://flickr.com.*
12. Goffman, E. *Interaction Ritual: Essays on Face-to-Face Behaviour.* Harmondsworth, Penguin Books, 1972.
13. Goffman E. *Presentationa of Self in Everyday Life.* New York, Anchor Books, 1959.
14. Guimbretiere, F., Stone, M. and Winograd, T. Fluid Interaction with High-Resolution Wall-Size Displays, In *Proc UIST 2001*, ACM Press (2001), 21-30.
15. Hall, E. *The Hidden Dimension.* Garden City, Doubleday, 1966.
16. Hartmann, B., Klemmer, S.R., Bernstein, M., and Mehta, N. d.tools: Visually Prototyping Physical UIs through Statecharts. In *Extended Abstracts of UIST 2005*, ACM Press (2005).
17. Horvitz, E., Kadie, C., Paek, T., and Hovel, D. 2003. Models of attention in computing and communication: from principles to applications. *In Communications of ACM* 46(3) ACM Press(March 2003), 52-59.
18. Ju, W., Ionescu, A., Neeley, L., and Winograd, T. Where the Wild Things Work: Capturing Physical Design Workspaces. In *Proc. of Conference on Computer Supported Cooperative Work 2004*, ACM Press (2004), 533-541.
19. Ju, W., Neeley, L., Winograd, T., and Leifer, L. Thinking with Erasable Ink: Ad-hoc Whiteboard Use in Collaborative Design. Technical Report #20060928, Center for Design Research, Stanford University, CA, 2006.
20. Klemmer, S.R., Newman, M.W., Farrell, R., Bilezikjian, M., and Landay, J.A. The Designers' Outpost: A Tangible Interface for Collaborative Web Site Design. In *Proc UIST 2001*, ACM Press (2001), 1-10.
21. Landay, J.A and Myers, B.A. Interactive Sketching for the Early Stages of User Interface Design, In Proc. CHI 1995, ACM Press (1995), 43-50.
22. LaVie, N. Perceptual Load as a Necessary Condition for Selective Attention. *Journal of Experimental Psychology: Human Perception and Performance*, 21:3 (1995), 451-468.
23. Lin, J., Newman, M.W., Hong, J.I. and Landay, J.A, "DENIM: Finding a Tighter Fit Between Tools and Practice for Web Site Design." In *Extended Abstracts of CHI 2000*, ACM Press (2000), 510-517.
24. Moran, T. P., Chiu, P, van Melle, W. & Kurtenbach, G., Implicit Structures for Pen-Based Systems within a Freeform Interaction Paradigm. In *Proc. CHI 1995*, ACM Press (1995), 487-494.
25. Moran, T. P.; Chiu, P.; Van Melle, B. Finding and Using Implicit Structure in Human-organized Spatial Layouts of Information. In *Proc. CHI 1996*. ACM Press(1996) 346-353.
26. Mynatt, E. D., Igarashi, T., Edwards, W. K., and LaMarca, A. Flatland: New Dimensions in Office whiteboards. In *Proc CHI 1999*, ACM Press (1999), 346-353.
27. Pashler, H. *The Psychology of Attention.* MIT Press, 1998.
28. Pedersen, E.R., McCall, K., Moran, T. and Halasz, F. Tivoli: An Electronic Whiteboard for Informal Workgroup Meetings. In *Proc. CHI 1993*, ACM Press (1993), 391-398.
29. Prante, T., Röcker, C., Streitz, N. A., Stenzel, R. , Magerkurth, C. , van Alphen, D. and Plewe, D. A. Hello.Wall –Beyond Ambient Displays. In *Adjunct Proceedings of Ubicomp 2003*. 277-278.
30. Reeves, B. and Nass, C. *The Media Equation*. New York, Cambridge University Press, 1996.
31. Russell, D.M., Trimble, J P., and Dieberger, A. The Use Patterns of Large, Interactive Display Surfaces: Case Studies of Media Design and Use for BlueBoard and MERBoard. In *Proc HICSS 2004*. IEEE (2004).
32. Streitz, N. A., Geißler, J., Holmer, T., Konomi, S. Müller-Tomfelde, C., Reischl, W. Rexroth, P., Seitz, P., and Steinmetz, R. i-LAND: An Interactive Landscape for Creativity and Innovation. In *Proc. CHI 1999*, ACM Press (1999), 120-121.
33. Tennenhouse, D. Proactive Computing. In *Communications of the ACM* 43:5, ACM Press (May 2000), 43-50.
34. Treisman, A. and Gelade, G. A Feature-Integration Theory of Attention. In *Cognitive Psychology*, 12 (1980), 97-136.
35. Vogel, D., and Balakrishnan, R. Interactive Public Ambient Displays: Transitioning from Implicit to Explicit, Public to Personal, Interaction with Multiple Users, In *Proc. UIST 2004*, ACM Press (2004), 137-146.
36. Weiser, M. The Computer for the 21st Century. *Scientific American*, 265:3 (September 1991), 94-104.
37. Weiser, M. Ubiquitous computing. In *IEEE Computer* 26, IEEE(October 1993), 71-72.
38. Weiser, M. and Brown, J.S. Center and periphery: Balancing the Bias of Digital Technology. In *Blueprint for the Digital Economy*, Tapscott, D. ed. McGraw-Hill (1998), 317-335.
39. Wisneski, C., Ishii, H. and Dahley, A. Ambient Displays: Turning Architectural Space into an Interface between People and Digital Information. In *CoBuild 1998*, (1998)
40. Wroblewki, L. Visible Narratives: Understanding Visual Organization. On *Boxes and Arrows*, AIGA, January 20th, 2003.