Making Sense of Sensing Systems:
Five Questions for Designers and Researchers

Victoria Bellotti, Maribeth Back, W. Keith Edwards, Rebecca E. Grinter,
Austin Henderson[¥] and Cristina Lopes

Xerox Palo Alto Research Center
3333 Coyote Hill Rd.
Palo Alto, CA 94304 USA
Tel: +1 650 812 4666 Fax: +1 650 812 4471
bellotti@parc.xerox.com

[¥]Rivendel Consulting & Design Inc.
P.O. Box 334
La Honda, CA 94020 USA
Tel: +1 650 747 9201 Fax: +1 650 747 0467
henderson@rivcons.com

ABSTRACT

This paper borrows ideas from social science to inform the design of novel "sensing" user-interfaces for computing technology. Specifically, we present five design challenges inspired by analysis of human-human communication that are mundanely addressed by traditional graphical user interface designs (GUIs). Although classic GUI conventions allow us to finesse these questions, recent research into innovative interaction techniques such as 'Ubiquitous Computing' and 'Tangible Interfaces' has begun to expose the interaction challenges and problems they pose. By making them explicit we open a discourse on how an approach similar to that used by social scientists in studying human-human interaction might inform the design of novel interaction mechanisms that can be used to handle human-computer communication accomplishments.

Keywords

Ubiquitous Computing, sensing input, design framework, social science, human-machine communication.

INTRODUCTION

Designers of user interfaces for standard applications, devices, and systems rarely have to worry about questions of the following sort:

> When I address a system, how does it know I am addressing it?

> When I ask a system to do something how do I know it is attending?

> When I issue a command (such as save, execute or delete), how does the system know what it relates to?

> How do I know the system understands my command and is correctly executing my intended action?

> How do I recover from mistakes?

Familiar GUI mechanisms such as cursors, windows, icons, menus, and drag-and-drop provide pre-packaged answers to

these key concerns. For example, a flashing cursor denotes that system is attending and what its focus is (where typed input will go). Such mechanisms have, by now, become conventions of commonplace and accepted *genres* for interaction. Indeed it is easy to forget that each one had to be carefully designed, before it ever became a convention.

By genre here, we mean *a set of design conventions anticipating particular usage contexts with their own conventions.* Examples of system genres include; games, productivity tools, and appliances and examples of interaction genres include, the GUI, voice activation and the remote control (for home entertainment systems). Genre makes design easier by pre-packaging sets of interaction conventions in a coherent manner that designers can use to leverage user expectations about the purpose and use of a device and to accommodate their existing skills.

By sticking to the GUI genre (and other simpler genres for cell-phones, video-recorders, microwaves and so on), using standardized toolkits, and by copying design ideas from existing solutions, designers now assemble myriad UIs for desktop, laptop, hand-held and other devices from pre-existing components without needing to ponder basic interaction issues. (While our discussion, in the rest of this paper applies to all of these established UI genres equally, we will address our arguments in particular towards comparisons with the GUI.)

However, those working in areas such as Ubiquitous Computing [30], where input is sensed by means other than keys, mouse or stylus (e.g., gesture, voice, or location), have no such well-understood, pre-packaged answers to these questions. Lacking these well-established precedents, designers of sensing systems must constantly confront these basic questions anew. In the rest of this paper we present a framework for addressing the resulting design challenges inherent in sensing systems, drawing on lessons about *human-human interaction* (HHI) in social science.

Our approach is not the same as presenting methods and guidelines for HCI design such as [21] or Apple's well-known Human Interface Guidelines [2]. Such texts are useful for designing systems within GUI-style interaction paradigms. Indeed they provide designers with generalizations relating to the parts, rules and meanings

constituting human-system dialog. However, these approaches tend to deal in specific interaction *mechanisms* rather than the general *accomplishments* they support, they do not fare well when applied to innovative genres of interaction beyond the GUI. Instead, our aim is to revisit and bring together some fundamentals of HCI, borrowing concepts from the social sciences, to provide a systematic *framework for the design of sensing systems*.

REFRAMING INTERACTION FOR SENSING SYSTEMS

We have, in the last decade, seen a number of innovations in interaction mechanisms best characterized overall as *sensing systems;* including Ubiquitous Computing (Ubicomp) systems [1, 30]; Speech and audio input [18, 27]; Gesture-based input [31] Tangible Interfaces or 'Phicons' (Physical Icons) [17] and Context Aware computing [1, 9, 18]. These sensing mechanisms have expanded what was previously a key-pressing, point-and-click interaction bottleneck, allowing systems to accept a far wider range of input than was previously possible. However, by definition, designers of these systems cannot simply copy existing precedents for handling input and output, unlike standard GUI designers. The *point* of their research is to tackle anew the many challenges that had to be addressed in the GUI and its cousins to make it over Norman's famous gulfs of execution and evaluation [22].

Interaction As Execution and Evaluation

Norman [22] proposes an "approximate model" of seven stages of action with respect to system interaction:

> Forming the goal
> Forming the intention
> Specifying an action
> Executing the action
> Perceiving the state of the world
> Interpreting the state of the world
> Evaluating the outcome

It is important to notice that *Norman's theory of action focuses on user cognition*. Moreover, it implicitly reflects a difference between HHI and HCI. Humans and computers are not equal partners in dialog. Computers are dumb slaves, have limited functionality, and rarely take the initiative. On the other hand, they have capabilities that humans do not. They can output precise information about their state, perform many rapid calculations simultaneously, emulate a vast range of tools and control multiple complex mechanical systems in parallel, and they can be guided and manipulated in many different ways by users.

The clever ploy embodied in the GUI is to exploit the different roles and relative strengths of computer and user and finesse the communication problem by forcing the user (using a display and a pointing and selecting device) to drive interaction, constantly discovering and monitoring which of many possible things the system is capable of and how it is interpreting ongoing action. Norman's account of HCI as an execution-evaluation cycle works well as long as we stick to the GUI genre that pre-packages solutions to the

interaction problem. In this case, the analytic interest then resides mainly in what's going on in the user's head.

Interaction as Communication

In contrast to Norman, our approach highlights *communicative*, rather than *cognitive* aspects of interaction. We agree with the coverage of Norman's model–from human intent to assessment of system action–but focus our attention on the *joint accomplishments* of the user and system that are necessary to complete the interaction, rather than the user's mental model. This stance is driven by a growing appreciation of two developments:

> The potential value of social science to the field of HCI. However, rather than focusing on the findings of sociologists about the use of technology in social settings [e.g., 7, 16] we are *using the kinds of questions addressed by social science in HHI as a model on which to pattern some of the science of HCI*. We understand, as we have said, that HHI and HCI cannot be regarded as identical problem spaces; however, we argue that despite the differences, many of the same communication challenges apply and must be recognized by designers.

> A trend in HCI towards sensing systems that dispense with well-known interaction genres, requiring us to return to the basic communication problems that the pre-packaged GUI interaction solutions so elegantly solved.

Goffman, an interaction analyst who has been particularly influential in social science, has written extensively on interpersonal verbal and non-verbal communication [12, 13, 15]. He provides a perspective on HHI that elucidates *how people manage accomplishments such as addressing, attending to and politely ignoring one another*. For example, signals are used to communicate intention to initiate, availability for communication, or that a listener understands what is being said. Surely *attention to similar mechanisms for HCI could be valuable*.

Further Goffman [14] also developed a notion of *frames* that are social constructs (such as a 'performance,' a 'game,' or a 'consultation') that allow us to make sense of what might otherwise seem to be incoherent human actions. Frames in HHI seem to parallel *genre* in HCI as defined above and may be useful constructs for *informing design*.

From Conversation Analysis, we know that successful conversation demands many basic accomplishments that most humans master. Sacks et al., [25] show how turn taking is managed as conversational participants organize their talk in an orderly fashion. Schegloff et al., [27] demonstrate how mistakes, and misunderstandings are repaired in communication. Button and Casey, [8] examine how people establish a shared topic in conversation. Similarly *humans and systems must manage and repair their communications, and must be able to establish a shared topic* (e.g., some action).

These perspectives provide inspiration for the following five issues that are intended to cover the same ground as

Norman's seven stages of execution, but with the emphasis now being on communication rather than cognition.

Address: Directing communication to a system.

Attention: Establishing that the system is attending.

Action: Defining what is to be done with the system (roughly equivalent to Norman's 'Gulf of Execution').

Alignment: Monitoring system response (roughly equivalent to Norman's 'Gulf of Evaluation').

Accident: Avoiding or recovering from errors or misunderstandings.

These issues may be posed as five questions that a system user must be able to answer to accomplish some action.

Table 1 shows how each question has a familiar GUI answer. Further, each one poses some challenges that are easily solved by sticking to the existing GUI paradigm and its simpler hand-held counterparts. However, for novel sensing systems, the challenges take center-stage as design issues again and we list some of them here, together with some potential problems caused by not addressing them.

ADDRESSING THE CHALLENGES OF INTERACTION

In this section, we review some of the ways each of our five questions is mundanely addressed by conventions in familiar GUI applications. We then consider alternative sensing approaches to interaction drawn from a number of recent research prototypes that expose the related challenges and either succeed or fail in addressing them.

1. Address

The first question we raise is so fundamental that it is often taken for granted in UI design: What mechanisms does the user employ to address the system? Analyses of HHI show that humans make use of a formidable array of verbal and non-verbal mechanisms to accomplish or avoid this activity [12].

The GUI Solution

In GUI applications, the "system" is a very clear concept; it's the box sitting on your desk. Designers know that if the user intends to interact with the system, he or she will use the devices, such as a keyboard or mouse, attached to it. There is little possibility for error, barring cables falling out of sockets, or users accidentally touching input devices.

Basic Question	Familiar GUI Answers	Exposed Challenges	Possible Problems
Address: How do I address one (or more) of many possible devices?	Keyboard Mouse (point-and-click) Social control over physical access	How to disambiguate signal-to-noise How to disambiguate intended target system How to not address the system	No response Unwanted response
Attention: How do I know the system is ready and attending to my actions?	Graphical feedback (e.g., flashing cursor, cursor moves when mouse moved) Assume user is looking at monitor	How to embody appropriate feedback, so that the user can be aware of the system's attention How to direct feedback to zone of user attention	Wasted input effort while system not attending Unintended action Privacy or security problems
Action: How do I effect a meaningful action, control its extent and possibly specify a target or targets for my action?	Click on objects(s) or drag cursor over area around object(s). Select objects from menu (e.g., recent files). Select actions from menu, accelerator keys, etc. Manipulate graphical controls (e.g., sliders).	How to identify and select a possible object for action. How to identify and select an action, and bind it to the object(s) How to avoid unwanted selection. How to handle complex operations (e.g., multiple objects, actions, and more abstract functions that are difficult to represent graphically, such as save).	Limited operations available Failure to execute action Unintended action (wrong response)
Alignment: How do I know the system is doing (has done) the right thing?	GUI presents distinctive graphical elements establishing a context with predictable consequences of action Graphical feedback (e.g., characters appear, rubber-banding) Auditory feedback Detectable new state (e.g., icon in new position)	How to make system state perceivable and persistent or query-able How to direct timely and appropriate feedback How to provide distinctive feedback on results and state (what is the response)	Inability to differentiate more than limited action space Failure to execute action Unintended action Difficulty evaluating new state Inability to detect mistakes Unrecoverable state
Accident: How do I avoid mistakes	Control/guide in direct manipulation Stop/cancel Undo Delete	How to control or cancel system action in progress How to disambiguate what to undo in time How to intervene when user makes obvious error	Unintended action Undesirable result Inability to recover state

Table 1. Five questions posing human–computer communication challenges for interaction design

Exposed Challenges

Such assumptions, however, are invalid when more "ambient" modes of input, such as gesture, are used, as well as when the notion of what precisely constitutes "the system" is a more amorphous concept. In such settings, the following challenges arise:

How to disambiguate signal-to-noise.

How to disambiguate intended target system.

How to not address the system.

Augmented Objects [29] tackle this challenge by the intuitive use of proximity of Augmented Objects to sensors; objects augmented with RFID (radio frequency identity) tags or IR (infrared) emitters can be waved at pickup sensors to initiate action.

Listen Reader [3] is an interactive children's storybook with an evocative soundtrack that the reader "plays" by sweeping hands over the pages. Embedded RFID tags sense what page is open, and capacitive field sensors measure human proximity to the pages. Proximity measurements control volume and other parameters for each page's sounds. Listen Reader, unlike Augmented Objects, allows users to address the system without using RFID tagged objects or IR emitters.

Digital Voices [19] is computer-to-computer interaction (CCI) mechanism that uses audible sound as the communication medium. A user can address a suitably equipped system using another Digital Voices-enabled device, as long as the devices can 'hear' one another. Moreover, the user hears the communication as it occurs.

One problem for sensing input approaches such as these is a risk of failure to communicate with the system if the sensing fails for any reason. The converse problem is avoiding unintended communications with devices that the user does *not* want to interact with. Simply getting too close can lead to accidental address, and so targets must be well spaced, and use limited sensing ranges or durational thresholds. However, auditory feedback from Digital Voices informs the user which devices are responding and helps them to decide whether the response is appropriate.

Accidentally addressing a system could be more than annoying, it could be a serious hazard [4]. Potential danger arises when people talk or gesture normally and a system becomes activated unintentionally. For example, a voice activated car phone triggered accidentally could compete for a driver's attention with serious consequences.

2. Attention

Our second question is related to, but distinct from, the first. The first question focuses only on addressing the system. In addition to this, users must determine whether and when the system is attending to them. Somehow the system must provide cues about attention, analogous to an audience sending signals of their attention (such as gaze and posture) to a human speaker [13].

The GUI Solution

Mechanisms such as flashing cursors and watch icons, are part of the established genre for communicating whether a system is accepting and responding to input. Such mechanisms assume the user is looking at the display.

Exposed Challenges

With sensing systems users may well be looking elsewhere than at a display. The design challenges here are:

How to embody appropriate feedback so that the user can be aware of the system's attention.

How to direct feedback to zone of user attention.

There are inherent problems with sensing UIs. Unobtrusively attached tags and sensors, make it hard for users to distinguish objects that the system is attending to from ones that the system is ignoring (un-augmented objects in the room). Without visible affordances users can unintentionally interact or fail to interact. Further, there could be privacy or security implications from unintended actions such as information being output simply because a user displaces an object and causes a system to become activated. Lack of feedback about system attention is common in many proposed and experimental systems [6].

Conference Assistant [9] is a system that uses sensing technology to identify a user (or rather, a device they carry) and supply information about the context to that user (such as the current speaker and paper being presented). The system also collects information about the user, including location, session arrival and departure times, and supplies this information to other conference attendees.

In this environment, the system is always attending whenever the user is within range. This raises the serious issue of how to keep users aware of what their peers are learning about them. In this design, there is no feedback to users to remind them that their actions are being monitored and recorded; in other words, the system does not provide feedback that it is accepting input from the user. Questions of user privacy have always followed new technologies and will continue to be a tough challenge [5].

In contrast to Conference Assistant, EuroPARC's audio-video ***media space*** [11] used monitors placed next to cameras in public places to tell inhabitants they were on-camera. In this case, if people saw themselves on the monitor, they could tell that the system was, in some sense, attending to them.

3. Action

Even once the user knows how to address the system, and is aware that it is, or is not, attending, more questions remain. The next is about how to effect action: How the user can establish what action she wishes the system to perform, how to control its extent (if it has extent) as well as how to specify (if there are any) targets of that action?

In Conversation Analysis, researchers have addressed somewhat similar issues in relation to establishing and maintaining topic [e.g., 7; 25]. Human understanding of

what Goffman [14] calls a frame, mentioned above, is also relevant, diminishing uncertainty about likely and acceptable actions. We now consider some of the HCI equivalents of these accomplishments.

The GUI Solution

Graphical items, such as menus, icons, images, text and so on, indicate, in Norman's Theory of Action terms, what the system is capable of (bridging his 'Gulf of Execution'). The problem of learning and memorizing how to express a meaningful command to a system (which humans find difficult) is translated into one of choosing from options. Users can explore the UI without changing anything; opening windows, pulling down menus, dragging the scrollbar to inspect contents, and so forth, to get a feel for the range of functionality offered by the application and the objects (such as text or messages) that can be acted on.

In Microsoft Outlook™, for example, a set of menus and toolbars provide access to the functions of the application. These actions can be bound to mail messages and folders, each of which is represented by an item in a list or, alternatively, by an open window. When a message is selected from a list, the user can ascertain which operations are available and which are disallowed for that particular object (disallowed operations are grayed out). In the window view, the set of operations that are allowable for the particular object are grouped together in that window.

In most cases, users perform an action on an object by first selecting the object and then selecting which action to apply to it. The patterns exemplified by Outlook are GUI genre conventions common to many graphical applications.

Exposed Challenges

With sensing systems the major challenges are as follows:

How to identify and select a possible object for action.

How to identify and select an action, and bind it to the object(s)

How to avoid unwanted selection.

How to handle complex operations (e.g., multiple objects, actions, and more abstract functions that are difficult to represent graphically, such as save).

The first three challenges become apparent as soon as designers attempt to create "invisible interfaces," in which the UI "disappears" into the environment [30]. In such settings the user is not looking at a computer screen, thus genre and conventions cannot be communicated (the user just has to know what to do). How, then, do sensing systems overcome these challenges?

Want et al.'s Augmented Objects are tagged so that each one can be *permanently bound to a single action* that is elicited by waving the object at a sensor. This provides a simple, *"unidimensional"* input mechanism whereby each object only causes a single action to occur when placed near a particular sensor. The *space of possible actions is limited* to the "actor" objects present in the environment.

The Listen Reader, like the GUI, uses *"matrix"* input; that is, it combines *two* kinds of input streams: four proximity sensors combined with an RFID reader. Unique RFID tags are buried within each page, so that the natural action of turning the page triggers the new set of sounds that will be elicited by gestures. The reader doesn't have to think about selecting new sounds; it's automatic.

In this case, the design is again constrained so that there are no "unwanted selection" or "action binding" issues and the set of possible actions is very small: The range of possible control over the sounds on each page is limited to relative volume, and perhaps pitch shift, but there are no "wrong" responses. This design is aimed at naïve users who will encounter the Listen Reader only once or twice (within what Goffman might call the frame of an exhibition).

As long as these objects are distinctive and suggestive of their action (and can be interpreted in terms of the frames for which they are designed), the range of possible actions may be known. Thus Tangible UIs in general [17] attempt to use physical traits of an object to communicate its virtual affordances. For example, the physical shape of an object may suggest certain uses for it, certain ways it should be held, and so on.

Thus, sensing UIs such as these actually handle the challenges of binding actions to targets, and supporting selection of actions and targets, rather elegantly. By embedding only a limited range of functionality into a set of suggestive physical objects, they provide a natural mechanism for users to bind actions to targets: They simply pick up or gesture at the object(s) of interest.

Our question about action here exposes the inherent challenges associated with binding more than limited system actions to physical objects. At the very heart of the vision for Ubicomp, the notion that *"computers [...] vanish into the background"* [30], lies a serious problem for interaction, which is communicating to the user which objects the potential for possible action is embedded in.

Sensor Chair, [23] is another gesture-based sound control system. The Sensor Chair was designed for the MIT Media Lab's 'Brain Opera.' Unlike Listen Reader, which is constrained for naïve, one-time users, the Sensor Chair is a musical interface with many layers of complexity and control. It does allow "wrong" responses, typically, an inability to discover effective gestures (to elicit system actions) or a miscalculation of spatial requirements. Systems for expert users, like the Sensor Chair, are difficult to use, require training and often rely on multimodal feedback, such as a variable light indicating strength of signal. Of course, they also support much more complex tasks such as a rich and skillful musical performance.

Sensetable [24] is a newer Augmented Objects system that, unlike earlier prototypes, is able to support the *dynamic binding and unbinding of actions to objects*. Sensetable uses augmented 'pucks' that are sensed by a tablet surface. Users can assign semantics to the pucks and manipulate

them on the table to effect computational actions, for example, by the physical binding of a modifier such as a dial on a puck. The puck may represent something like a molecule and turning the dial represents the action of changing its charge. This is a compelling GUI-Phicon hybrid solution to the challenges related to establishing an action and an object to apply the action to. However, it still leaves open the question of how to apply actions to multiple objects simultaneously.

For sensing systems in general a persistent challenge is that abstract operations such as 'copy' or 'find' are likely to be awkward or severely restricted without some means to specify an argument (e.g., where to copy to and what to save the result as). It may be that such systems simply do not lend themselves to operations that may be best suited to keyboard input. Or it may be that researchers have yet to establish new non-GUI ways to do these things.

4. Alignment

Sociologists pay a great deal of attention to the mechanisms that support coordination or alignment of speaker and listener as a conversation progresses [26]. Back-channeling is a term used by linguists to refer to feedback a listener gives as to her ongoing understanding, which is monitored by the speaker. Similarly, systems users must be able to monitor system understanding of their input; in other words to bridge Norman's 'Gulf of Evaluation.'

The GUI Solution

Graphical interfaces display current state, action and results, through *feedback* mechanisms such as echoing input text and formatting, rubber-banding, wire-frame outlines, progress bars, highlighting changes in a document, listing sent messages and so on. In the rare instances where the system takes the initiative (as in Word's 'AutoFormat,' which monitors user actions and deduces automated formatting), the user sees the results in real time as they work (or don't, as the case may be).

Exposed Challenges

The mechanisms above overcome the following challenges:

How to make system *state* perceivable and persistent or query-able.

How to direct *timely* and *appropriate* feedback.

How to provide *distinctive* feedback on results and state (what is the response).

Our first challenge is one of how the user may determine current state. However, by definition, Ubicomp is everywhere, embedded in mundane objects. So the goal of making state perceivable and persistent or query-able seems daunting without something very like a GUI.

With Augmented Objects, gestural UIs, 'sonified' input-output (I/O) systems like Digital Voices, and other novel sensing systems, the risk is that users will not be able to tell whether the system understands or not what the user is trying to do. Without a GUI equivalent, such as the one provided by Sensetable, how does the user know how the system is responding to their gesture? As long as the failure

mode is not problematic, trial and error may be acceptable, but this will certainly restrict the scope of such an interaction style to applications with a more limited space of actions.

Augmented Objects, gestural UIs and sonified I/O do not presuppose any mechanism to display state information in a manner that is consistent with the mode of input.

With respect to the first and third challenges, if a state change is a part of the function of a system, then these issues must somehow be explicitly addressed. We might propose ongoing projection of graphical or audio information into the space of action. Sensetable takes the former approach, displaying both distinctive and persistent information, however this is presently done at the cost of restricting users to working within the projected area. Ideally, Augmented Objects themselves should be capable of displaying the states they have acquired through action.

With respect to the second challenge, Digital Voices has been designed to address the low-end of speed of digital communications, that is, interactions that occur at hundreds of bits per second and that usually take a few seconds to occur. Therefore the timeframe of the machines' interaction is the same as the people's timeframe, and the user can perceive the interaction in real-time as it happens. They can also do so without having to watch the system, for example, they might be attending to other matters, thus the audio channel can be an appropriate alternative to visual displays.

Likewise, Sensor Chair, in addition to playing sounds in response to user proximity to its sensors, gives additional visual cues, in the form of variable intensity lights. Experts can use this timely feedback to further ensure that the system distinctly senses actions around each of its sensors.

As another example of alignment, the **Speakeasy** framework for Ubicomp [10, 20] provides facilities to query and display the state of devices, such as projectors, PCs and printers, and also services in an environment (such as a lecture theater). Users can discover what these entities are doing, if they've failed, if they're available, and so on.

5. Accident

Our final question deals with not only preventing mistakes in the first place, but also informing users about mistakes that have already occurred so they can correct them. Conversation analysts [26, 27] have dealt extensively with breakdowns and repair in HHI, observing that misunderstandings are much more commonplace than one might expect. Likewise, error is an important and to-be-expected part of normal HCI. Thus, as Norman [22] states, "interaction should be treated as a cooperative endeavor between person and machine, one in which misunderstandings can arise on either side."

The GUI Solution

Since systems, unlike humans, often perform actions instantaneously, it is not always possible to provide useful feedback that will allow intervention during system action. However, when a user makes a mistake in Word or Outlook

text editing, they can usually see the result straight away and applications offer 'undo' through a menu item or accelerator key. In Word, certain errors, such as poor speling, can be highlighted or corrected automatically. Many of the actions of the system are also visibly displayed and easily correctable, if the user notices them. These feedback mechanisms occur *after* the action is completed.

Other tasks, such as a long print job or a software compilation, may be long-lived, taking several minutes or even hours to finish. Tools designed to support such work often provide feedback *during* action to allow users to monitor (and cancel) the task. For example, an animated printer icon on the desktop may show that the printer is working, or has stopped working, and provides controls to allow the user to stop a print job.

Some actions, however, are rapid, do not lend themselves well to "preview" feedback, or to easy cancellation, and are inherently undoable. In Outlook it is not possible to retract a message that has been mis-sent (in fact it is very hard to tell this ever happened unless a bounce message arrives). In Word, if the user accidentally saves a document over another document, the application cannot correct the mistake. Experience with such problems means that designers are advised to make risky operations more difficult [4, 22] or to present alert boxes *before* action to protect users from unrecoverable mistakes; however, alert boxes can be irritating and must be used sparingly.

Exposed Challenges

How to control or cancel system action in progress

How to disambiguate what to undo in time

How to intervene when user makes obvious error

In order to correct mistakes, they have to be visible *in time* to take action before it is too late; perhaps during or immediately after a system response, and sometimes even before. Feedback needs to be instantaneous, but without a GUI, ambiguity is a serious problem. Both the action and its object need to be represented in a manner such that they can both be identified and specified as targets for undo.

There has been little discussion in the Ubicomp literature so far concerning failure modes and errors. For example, the designers of Augmented Objects [30] and Sensetable [24] do not even mention the possibility of error! In Listen Reader, a heavily constrained museum piece, error is designed out of the system; the user cannot do anything "wrong." This is one possible route to go with sensing systems, but it works only in simple interaction situations (e.g., for naive users). More complex systems must allow for error as a trade-off against greater achievement.

Digital Voices applications, are appealing in that constant feedback is provided, which should allow the user to cancel an error in progress (for example a transmission that is unwanted). However, it is not clear how users could differentiate system communications that contain erroneous content from correct ones.

As things stand in sensing systems, our accident-avoidance challenges, though serious, are largely unaddressed. We believe this is because the field of sensing system research is in its infancy and the existing prototypes have so far been restricted to areas where erroneous behavior has limited consequences. Future, more ambitious systems will most likely need to provide a wide range of mechanisms for dealing with the common problem of error.

DISCUSSION

We have presented the five questions of our framework for designing interaction with sensing systems. One final and important question is: *How do we intend this framework to be used?*

Informing Design

We believe that the issues we have raised provide the beginnings for a systematic approach to the design of interactive systems without tried and tested precedents. In particular, we have addressed our arguments to a novel class of systems that obtain user-input through sensing user action, rather than through standard input devices such as the keyboard, mouse or stylus. By considering each of our questions and ensuring they have dealt with the corresponding challenges, designers should be able to avoid a number of potential hazards or pitfalls.

As just one example, automobiles are gradually acquiring a growing number of on-board systems such as hands-free phones, navigation and security systems, etc. Looking to the future, we might anticipate a number of problems as the voice channel in the car becomes increasingly overloaded and displays proliferate. Our framework is a starting point for those wishing to find innovative solutions without making hazardous design mistakes [4].

Framing Further Research

This paper also represents an invitation to social scientists, in particular, interaction and conversation analysts, to develop and improve on our analysis and to apply their understandings of human-human interaction to help designers develop systems that can communicate more naturally and effectively with people.

We are working in a time when systems are rapidly taking on many new forms and functions, faster even than people can find uses for them. With so much design innovation ongoing, there is a wide range of opportunities for social scientists to team up with innovators in academic and commercial research settings to define and refine new mechanisms that will become the conventions of future interaction genres.

Our aim here is to open a new discussion on innovative design research for human-machine communication and we look forward to further efforts in this area.

ACKNOWLEDGMENTS

REFERENCES

1. Abowd, G. & Mynatt, E. Charting past, present, and future research in ubiquitous computing. ACM Trans. Computer-Human. Interaction. 7, 1 (Mar. 2000), 29-58.

2. Apple Computer, Inc. Macintosh Human Interface Guidelines. Reading, MA: Addison-Wesley Publishing Co., 1992. ISBN 0-201-62216-5.

3. Back, M., Cohen, J., Gold, R., Harrison, S., & Minneman, S. Listen Reader: An electronically augmented paper-based book. *Proceedings of CHI 2001* (Seattle WA, April 2000) ACM Press, 23-29.

4. Barfield, L. The Real World: Powerful Functions. *SIGCHI Bulletin, 29, 2,* April, 1997.

5. Bellotti, V. Design for Privacy in Multimedia Computing and Communications Environments. In P. Agre, & M. Rotenberg (eds.). *Technology and Privacy: The New Landscape*. MIT Press: Cambridge (1998).

6. Bellotti, V & Rodden T. The Perils of Physical Interaction. In Proceedings of W. Mackay (ed.) *DARE 2000, Designing Augmented Reality Environments*, (Elsinore, Denmark, April 2000) ACM Press.

7. Button, G. (ed.). *Technology in Working Order.* Routledge, London, UK (1993).

8. Button, G., & Casey, N. Generating topic: The use of topic initial elicitors. In J. Atkinson & J. Heritage (eds.) *Structures of Social Action. Studies in Conversation Analysis*. Cambridge University Press, (1984) 167-189.

9. Dey, A., Salber, D, & Abowd, G. A Conceptual Framework and a Toolkit for Supporting the Design of Context Aware Applications. *HCI Journal, Special Issue on Context Aware Computing*, (in press).

10. Edwards, W.K., Newman, M.W., Sedivy, J.Z. The Case for Recombinant Computing. *Xerox PARC Technical Report CSL-01-1*. April 2001.

11. Gaver, W., Moran, T., MacLean, A., Lovstrand, L., Dourish, P., Carter, K., & Buxton, W. Realizing a Video Environment: EuroPARC's RAVE System. *Proceedings of CHI'92* (Monterey, CA, May 1992). ACM Press 27-35.

12. Goffman, E. *The Presentation of Self in Everyday Life*. Anchor, Doubleday, New York, (1959).

13. Goffman, E. *Behavior in Public Places*. Free Press, Macmillan, Boston, (1963).

14. Goffman, E. *Frame Analysis: An essay on the organization of experience*. Northeastern University Press, New York, (1974).

15. Goffman, E. *Forms of Talk*. Oxford: Basil Blackwell, (1981).

16. Hughes, J., King, V., Rodden, T. & Andersen H. The Role of Ethnography in Interactive Systems Design, *Interactions, 2,2*. April, 1995, 56-65.

17. Ishii, H. & Ullmer, B. Tangible Bits: Towards seamless interfaces between people, bits and atoms. *Proceedings of CHI'97* (Atlanta, GA, March 1997) ACM Press, 234-241.

18. Long, S., Aust, D., Abowd, G. & Atkeson, C. Cyberguide: prototyping context-aware mobile applications. *Proceedings of CHI'96* (Vancouver, Canada, April, 1996) ACM Press, 293-294.

19. Lopes, C. & Aguiar, P. Aerial Acoustic Communications. *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (Mohonk Moutain Resort, New York. October 2001).

20. Newman, M., Sedivy, J., Edwards, W.K., Hong, J., Izadi, S., Marcelo, K., Neuwirth, C., & Smith, T. Designing for Radical Interoperability: User Experience Issues in Recombinant Computing. (Submitted for publication).

21. Newman, W. & Lamming, M. *Interactive System Design*. Addison Wesley, (1995).

22. Norman, D. A. *The Design of Everyday Things*. Doubleday: New York, New York, (1990).

23. Paradiso, Joseph. The Brain Opera Technology: New Instruments and Gestural Sensors for Musical Interaction and Performance. *Journal of New Music Research, 28, 2,* (1999), 130-149.

24. Patten, J., Ishii, H., Hines, J. & Pangaro, G. Sensetable: A Wireless Object Tracking Platform for Tangible User Interfaces, *Proceedings of CHI'01* (Seattle, WA, April 2001), ACM Press, 253-260.

25. Sacks, H., Schegloff, E. & Jefferson G. A simplest systematics for the organisation of turn-taking for conversation", *Language 50,* (1974) 696-735.

26. Sacks, H., Schegloff, E. & Jefferson G. *Lectures on conversation*. Oxford: Basil Blackwell (1992).

27. Schegloff, E., Jefferson, G., & Sacks, H. The preference for self-correction in the organization of repair in conversation. *Language, 53,* (1977), 361-382.

28. Schmandt, C. *Conversational Computing Systems*. New York, Van Nostrand Reinhold, 1993.

29. Want, R., Fishkin, K., Gujar A, & Harrison, B. Bridging physical and virtual worlds with electronic tags. *Proceeding of CHI'99* (Pittsburgh, PA, May 1999). ACM Press, 370-377.

30. Weiser, M. The Computer for the Twenty-First Century. *Scientific American, 265, 3* (September 1991), 94-104.

31. Zimmerman, T., Smith, J.R., Paradiso, J.A., Allport, D. and Gershenfeld, N. (1995). Applying Electric Field Sensing to Human-Computer Interfaces. *Proceedings of CHI'95,* (Denver, CO, May 1995) ACM Press. 280-287.