Jeremy R. Cooperstock, Sidney S. Fels, William Buxton, and Kenneth C. Smith

Reactive Environments

Throwing away your keyboard and mouse.

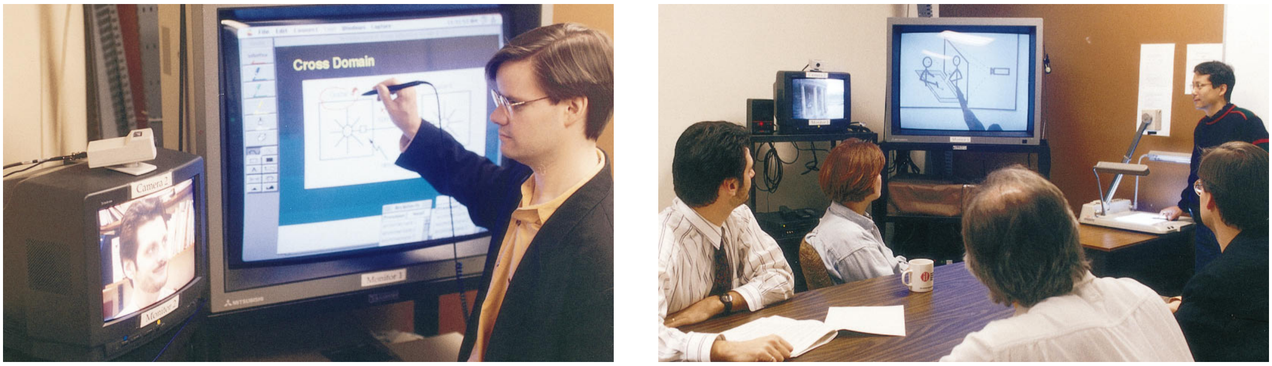


Figure 1. a) The digital whiteboard in use. The design being sketched is visible to people in the room and to the telepresent attendee, whose image appears on the smaller monitor. **b)** The speaker (top-right) is illustrating a diagram in the document camera. The document is displayed on the large video monitor and is also visible to the telepresent attendee.

As information technology becomes increasingly widespread, we are confronted with the burden of controlling a myriad of complex devices in our day-to-day activities. While many people today could hardly imagine living in electronics-free homes or working in offices without computers, few of us have truly mastered full control of our VCRs, microwave ovens, or office photocopiers. Rather than making our lives easier, as technology was intended to do, it has complicated our activities with lengthy instruction manuals and confusing user interfaces.

Designers have been trying to make the computer more "user-friendly" ever since its inception. The last two decades have brought us the notable advances of keyboard terminals, graphics displays, and pointing devices, as well as the graphical user interface, introduced in 1981 by the Xerox Star and popularized by the Apple Macintosh. Most recently, we have seen the emergence of pen-based and portable computers. However, despite this progress of interface improvements, very little has changed in terms of how we work with these machines. The basic rules of interaction are the same as they were in the days of the ENIAC: users must engage in an explicit, machine oriented dialogue with the computer rather than interact with the computer as they do with other people.

In the last few years, computer scientists have begun talking about a new approach to human-computer interaction in which computing would not necessitate sitting in front of a screen and isolating ourselves from the world around us. Instead, in a *computer-augmented environment*, electronic systems could be merged into the physical world to provide computer functionality to everyday objects. This idea is exemplified by Ubiquitous Computing (UbiComp)

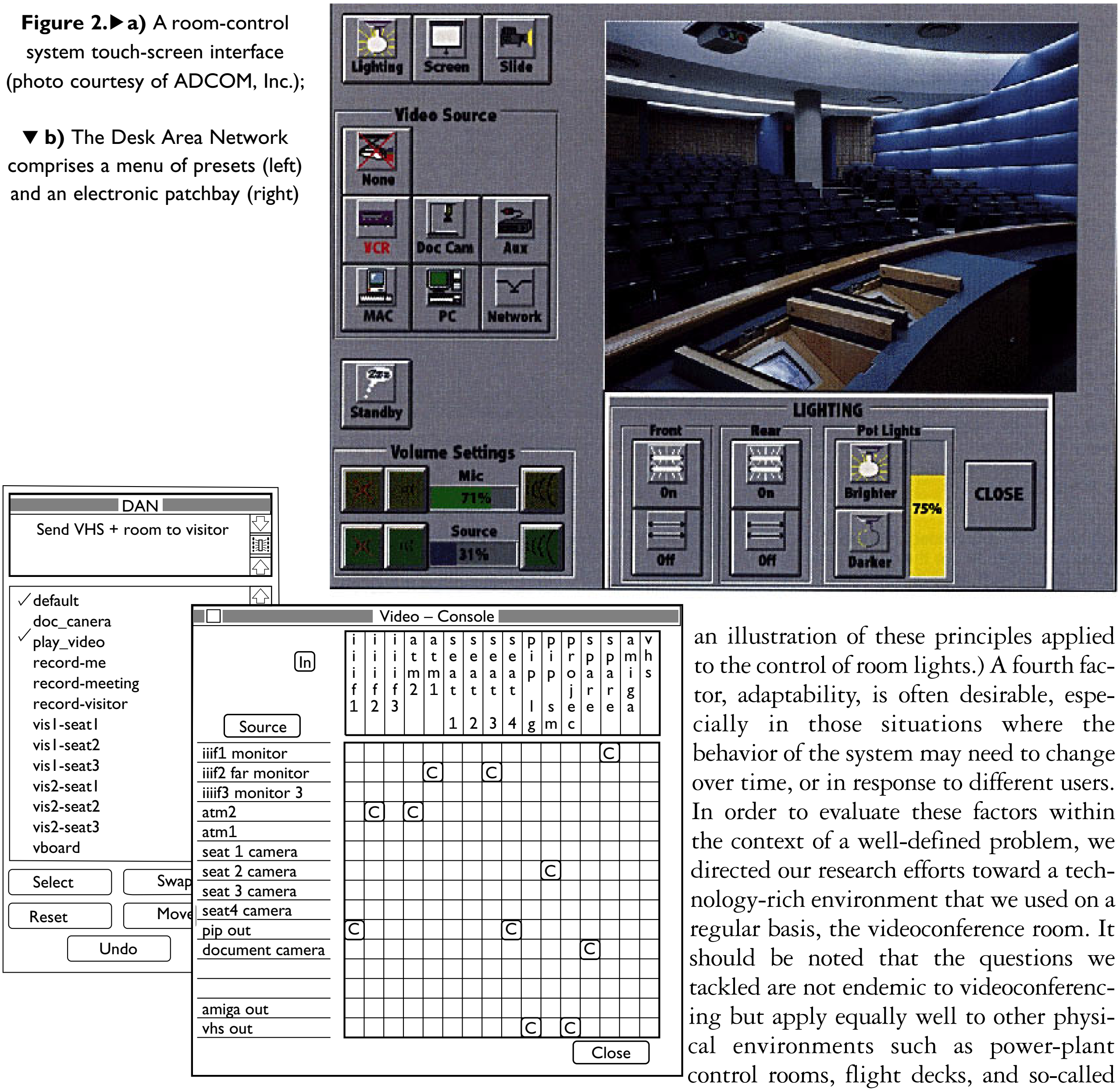


Figure 2.▶ a) A room-control system touch-screen interface (photo courtesy of ADCOM, Inc.);

▼ b) The Desk Area Network comprises a menu of presets (left) and an electronic patchbay (right)

DAN

Send VHS + room to visitor

✓default
doc_canera
✓play_video
record-me
record-meeting
record-visitor
vis1-seat1
vis1-seat2
vis1-seat3
vis2-seat1
vis2-seat2
vis2-seat3
vboard

Video – Console

	iiif1	iiif2	iiif3	atm2	atm1	seat 1	seat 2	seat 3	seat 4	pip lg	pip sm	projec	spare	spare	amiga	vhs
iiif1 monitor														C		
iiif2 far monitor					C			C								
iiiif3 monitor 3																
atm2		C		C												
atm1																
seat 1 camera																
seat 2 camera											C					
seat 3 camera																
seat4 camera																
pip out	C								C							
document camera													C			
amiga out																
vhs out										C		C				

[10] and Augmented Reality [1]. Proponents argue that systems should be embedded in the environment. The technology should be distributed (ubiquitous), yet invisible, or transparent, since the full potential of the computer can only be realized when the machine itself is hidden from the user. This concept marks a dramatic shift from the status quo in which interaction with the computer interferes with our activities rather than enhancing them.

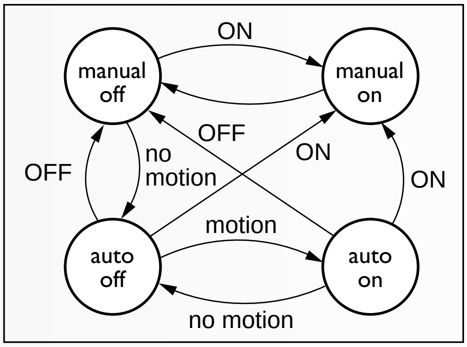
While the promise of technology based on UbiComp is truly exciting, we believe this approach will succeed only if the design of these systems takes into account the human factors governing their use. The factors we consider necessary for usable technology include the invisibility previously described, the affordance of a seamless manual override, and provision of feedback to the user. (See the accompanying sidebar for an illustration of these principles applied to the control of room lights.) A fourth factor, adaptability, is often desirable, especially in those situations where the behavior of the system may need to change over time, or in response to different users. In order to evaluate these factors within the context of a well-defined problem, we directed our research efforts toward a technology-rich environment that we used on a regular basis, the videoconference room. It should be noted that the questions we tackled are not endemic to videoconferencing but apply equally well to other physical environments such as power-plant control rooms, flight decks, and so-called “smart homes,” as well as to software environments such as integrated office suites.

Why Videoconferencing?

Put simply, the state of the art in videoconference environments provides us with a superb example of technology gone awry. Our conference room equipment includes several cameras and monitors, a VCR, a digital whiteboard, pictured in Figure 1a, and an electronic document camera, shown in Figure 1b, which replaces the standard overhead projector typically found in such environments. The output of these devices can be displayed on any of the monitors in the room and sent to remote, or *telepresent*, participants as well. We are given many wonderful tools, enabling geographically disparate participants to meet, discuss, collaborate, and educate. But control

The Smart Light Switch

Motion detectors, or other occupancy detectors, are often used to activate room lights automatically whenever motion is detected. The simple act of entering, or moving within a room triggers a detector, which completes a circuit, causing the lights to be switched on. Since no explicit interaction takes place between the human and the technology, the mechanism is invisible.



Smart lights diagram. The transition labels ON and OFF denote the pressing of the ON and OFF buttons respectively.

While conventional motion detectors are adequate for automatic control of room lights, their support for manual override is typically quite poor. For example, in order to keep the lights on indefinitely, users must toggle the power off, and then on, in a relatively short time interval. To turn the lights off manually, conventional motion detectors must be disabled at the power supply, or a second switch must be inserted between the motion detector and the lights. In either case, the manual override mechanisms (for on or off) are inconsistent and require special knowledge on the part of the user. Furthermore, manual intervention is required to restore the system to its automatic mode after it has been overridden.

In response to these problems, we developed the Smart Light Switch, which incorporates an obvious manual override mechanism,* a standard soft-touch on/off switch [5], with conventional motion detectors. This system operates in four states or modes, as illustrated by the state diagram appearing in this sidebar. In either *auto-on* or *auto-off* modes, the system acts as an ordinary motion-sensitive light switch. However, when the on or off button is pressed, the system enters the corresponding *manual-on* or *manual-off* mode, thereby causing the switch to behave in a manner consistent with user expectations. A key distinguishing feature of our system is that unlike other motion-sensitive light controllers, the Smart Light Switch does not need to be manually reactivated after being turned off. While motion in the room persists, the switch remains in the *manual-off* mode, but after motion ceases, the switch returns to the *auto-off* mode by itself. This means that when motion is again detected, the switch will enter the *auto on* mode and the lights will be activated.

Although the behavior of our motion-sensitive light controller is quite predictable, we still found it useful to provide state information for feedback purposes. This was accomplished by adding an LED panel that indicates which of the four states the switch is in. As described in the table appearing here, if a new user enters the room and the lights fail to turn on, a quick glance at the LED panel, located beside the light switch, could explain the reason.

An encouraging indication of the success of our design was the fact that most users of the room did not even inquire about the Smart Light Switch. For normal operation, the lights automatically turned on when the room was entered, and turned off after the room was vacated. If the occupants desired the lights to be turned off, or back on again, they could do so manually at any time, exactly as they would have had the Smart Light Switch never been installed. Regardless of whether or not they were aware of the technology, users remained oblivious to it.

LED indicators			**explanation**
auto	**light on**	**motion**	
		✔	manual override in effect
✔			sensitivity too low or no recent notice
✔	✔	✔	light bulb burnt out or bad connection

Probable causes of light not turning on when a user enters the room

*To avoid the problem of the manual switch providing a (potentially incorrect) representation of the state of the system, we use a soft-touch (return-to-center) on/off switch for this purpose.

of these tools is either so limited as to render them ineffective, or so complex that a trained expert is required to operate them.

From our experience in the Ontario Telepresence Project [8] and from our observations of users with various room control systems, we have frequently seen meeting breakdowns occur. Simply giving a presentation is difficult enough, but the additional burden of managing the technology often places too great a cognitive load on the presenter. While control of a single device in isolation is manageable, the complexity increases dramatically when the same device must be operated in conjunction with several others. For example, most of us have little difficulty in turning on room lights or playing a videotape. These are skills that we have mastered through our day-to-day experiences. However, the nature of the problem changes when the same tape must be shown to a remote participant, while both parties continue to see and hear each other. State of the art control system interfaces, such as the touch-screen interface shown in Figure 2a, and the Telepresence Desk Area Network, shown in Figure 2b, attempt to simplify matters by providing a human-computer interface allowing users to select from a number of devices as input sources. The control system then configures the equipment so the audio and video signals are routed appropriately.

Our underlying assumption was that if a human operator is able to infer users' intentions based on their actions, so should an appropriately designed system.

Unfortunately, such interfaces tend to exaggerate rather than solve the problems of current technology. Presenters often require a configuration of equipment that the control system does not provide (e.g., I want to display a document on the large screen and the remote participant on the small screen, but the system only allows me to do the opposite). While the electronic patchbay of the Desk Area Network offers flexibility, there is still the problem of locating the desired selection when a device is activated. This is only exacerbated by the cognitive load of mapping text labels or interface icons to the devices and operations they represent, and vice versa [9]. For example, if I select the "play-video" label in Figure 2b, where will the output go—to all participants or just to a local monitor? Similarly, if I select the "record-meeting" label, will this record only what is taking place locally, or will it also record the remote participants?

Even under the best circumstances, when presenters remember to operate the control system at the right times, meetings involving the videoconference equipment still tend to be awkward. The need to exercise explicit manual control through a user interface is too distracting, both to presenters, who must interrupt their talks, and to the participants, who must endure the interruptions. Many of our conference room users preferred simply to leave the room the way it was rather than deal with the complexities of the interface, even when the configuration was awkward for their particular task. Alternately, some presenters relied on a highly skilled third party to operate the equipment and to ensure that all participants receive the appropriate view.

The Reactive Room

The root of these problems is that we have been stuck in our ways of thinking about computers. All of our interaction with the technology is through the highly limited channel of communication provided by the user interface and takes place purely at the level of the machine. As a result, we cannot "walk up and use" the technology, but must be trained in its operation.

Our research efforts to address these problems led us to develop the concept of a Reactive Environment, a variation of the "skilled operator" theme, in which the technology itself, rather than a human, manages the low-level operation of the room. Some early work in this direction includes the Responsive Office Environments of Elrod et al. [3] and the Augmented Reality kitchen described by Kellogg, Carroll, and Richards [5]. Our underlying assumption was that if a human operator is able to infer users' intentions based on their actions, so should an appropriately designed system. The intent was to reduce the cognitive load of the user by allowing the system to make context-sensitive reactions in response to the user's conscious actions. These efforts culminated in the implementation of such a computer-augmented

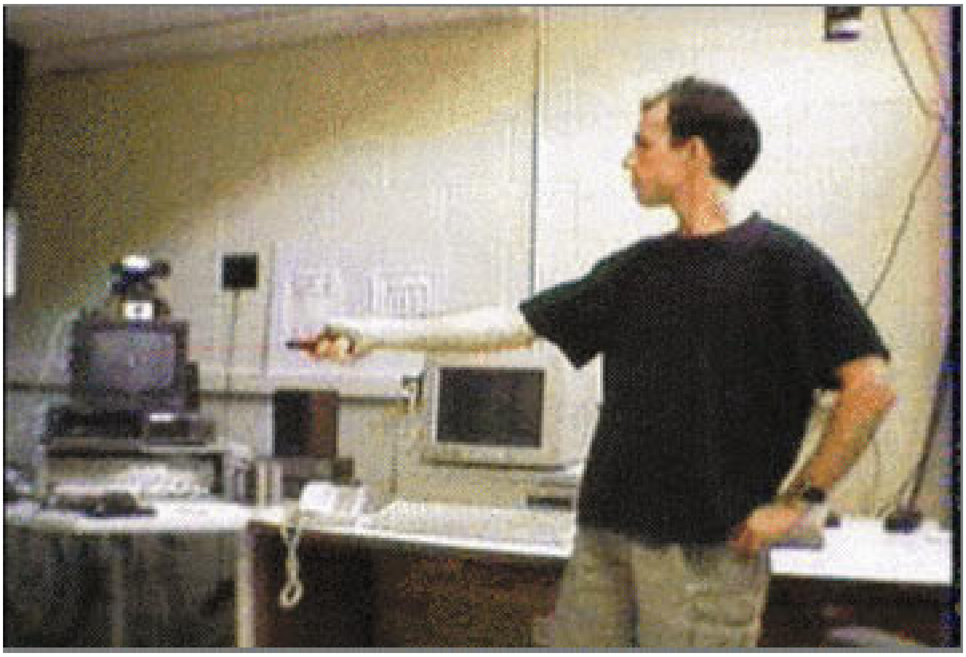


Figure 3. The laser pointer in use.
The speaker is selecting a view for the remote participant by pointing the laser at an electronic seat.

videoconferencing environment, called the Reactive Room [2]. Before elaborating on the details of our prototype, we first illustrate its operation by presenting a possible scenario:

Just before noon, Nicole arrives at the university and enters the lab. The room lights turn on automatically and an audio message greets her. While organizing her presentation for the afternoon, she is distracted by the fluorescent lights, and so turns these off. An hour later, she leaves for a brief meeting and returns just before the presentation is scheduled to begin. When she re-enters the room, the lights turn on again. An electronic calendar that has been awaiting her arrival then activates the presentation equipment and initiates a video connection with the conference room automatically.

Nicole begins her presentation by placing a diagram under the document camera. The remote participants immediately receive a view of this diagram, along with a small "picture-in-picture" of the presenter. When Nicole places a tape in the VCR and presses the play button, the participants see the contents of the tape. From her current position in front of the VCR, Nicole cannot easily see her audience. However, by pressing a button labeled "remote participant" and then a button on a monitor near the VCR, she can move the audience to a more convenient location. An LED over each of these buttons illuminates, and a double beep sounds, indicating that the move has been accomplished. Once the tape stops, the document becomes visible again. Finally, when Nicole removes the diagram, the participants receive a full view of her.

At this point, a new telepresent participant, Alex, joins the meeting. A doorbell sound alerts Nicole to the arrival of the new participant. From his initial position, Alex cannot see Nicole, but by leaning slightly to the left, he causes a motorized camera to slowly pan toward the presenter until she becomes visible.

As seen in this scenario, the Reactive Room satisfies our design principles of invisibility, manual override, and feedback. First, by transferring responsibility for the low-level control of complex technology from the presenter to the Reactive Room, we reduce the cognitive burden and hence, the amount of training required. Instead of relying on a user interface, the technology reacts to the high-level actions that the presenter performs, for example, placing a document under the camera or pressing the play button on the VCR. In other words, the user interface is made invisible. Second, the affordance of a direct manual override mechanism for both the room lights and the presentation devices allows users to override default behavior seamlessly without the confusion of mapping user interface representations to the corresponding devices. Third, the use of audio and visual feedback provides confirmation that various operations have succeeded. In addition, support for remote participants improves their sense of engagement in the meeting and allows them to adjust their views without interrupting the presenter.

Invisibility

To make the user interface invisible, the Reactive Room's presentation technology has been augmented with sensors, computers, and communications. Each device is monitored by one or more daemons, which collect information from sensory input. From this background monitoring and some computation, dae-

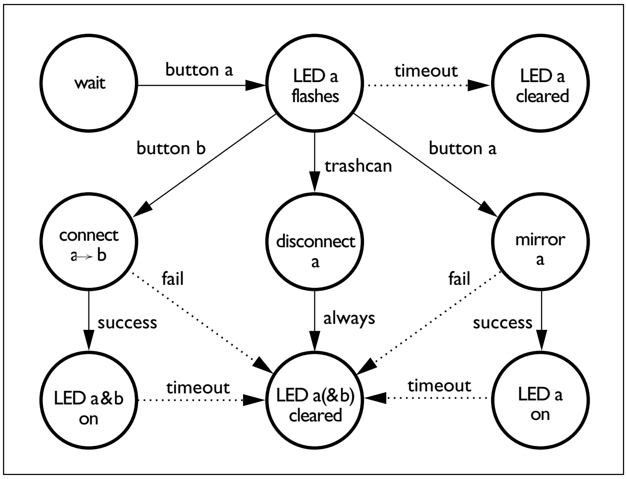


Figure 4. State diagram of the button-and-light modules. The first row of states corresponds to the processing associated with the first button press, while the second row represents the actions taken in attempting to form or drop connections in response to the second button press. The final row depicts the possible states of the system resulting from the operation just performed. Dashed lines are used to indicate state transitions caused by erroneous or incomplete button press sequences.

mons maintain "awareness" of activity relevant to each device and share information with each other when required. For example, the document camera daemon "knows" whether or not a document is on the table by processing the video signal from this device. When the image contains some region of high contrast, the Reactive Room displays the document camera output on an appropriate monitor, and provides the same view to remote participants. If the image becomes uniformly grey, and remains this way for a certain time period, then the daemon assumes that the document has been removed and reacts accordingly. Another daemon monitors the status of a microswitch, installed in the pen holster of the digital whiteboard. When the pen is picked up, the switch opens and the whiteboard is considered to be in use, causing its output to be displayed automatically. Similarly, the VCR daemon polls the status of the VCR and reacts to various operations.

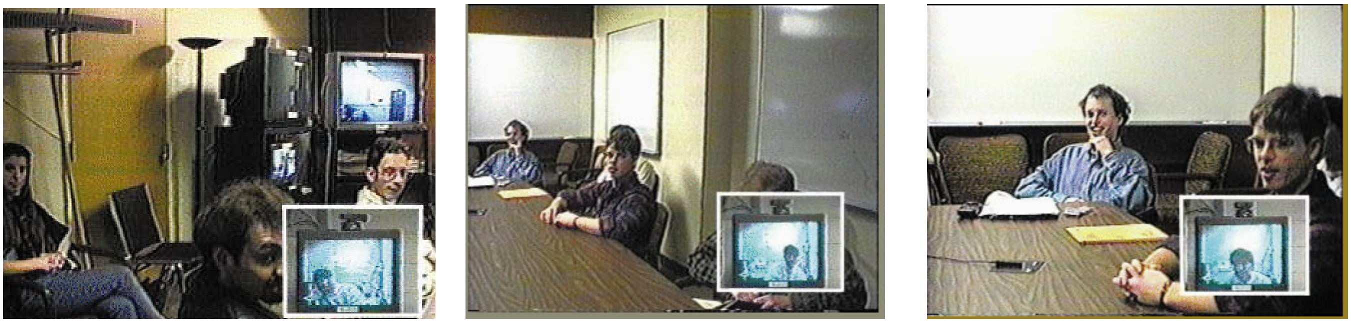


Figure 5. The Virtual Window camera control system in operation. The larger images represent the view received by the remote participant, while the small inset images represent the appearance of the participant on a local video monitor. The motorized camera appears at the top of the video monitor.

In general, context plays an important role in determining the intended behavior of each device. For example, if the record button is pressed during a meeting with remote participants, the VCR should record both the local and remote views, possibly by routing video signals through a mixer or "picture-in-picture" device. However, if there are no remote participants, then the VCR need only record the local view. In our reactive environment, knowledge of whether a remote participant is attending the meeting is obtained through communication with another daemon responsible for controlling connections to our media space. All of these interactions follow the principle of invisibility. The simple act of placing a document under the camera, picking up a pen to draw on the whiteboard, or pressing the play or record button on the VCR is sufficient to trigger the appropriate sequence of events

Manual Override

Relegating control of technology to a collection of background processes introduces the risk of preventing the user from superseding the default automatic behavior of the system when appropriate or desired. For instance, the fact the VCR output is automatically routed to an appropriate monitor is of little benefit if the presenter cannot mute the volume when necessary. There may also be situations in which the user explicitly wishes to disable the computer from initiating any activity, in other words, make use of the "master off switch."

The question then becomes how one provides users with a simple and seamless *manual override* mecha-

Reactive Environments break through the barriers of traditional keyboard-and-mouse computing and offer an intuitive way of interacting with our surroundings.

nism, to deal with those occasions where the default behavior of the technology differs from their intentions. If I walk into a room and the lights turn on automatically, I still want the ability to turn them off at any time, without resorting to a computer interface or an instruction manual. Similarly, users must be able to override the Reactive Room by establishing connections between various devices without directly handling the computer. Resorting to the GUI patchbay of Figure 2b for manual override is simply not acceptable. If use of the manual override is as complicated as the original GUI, users are unlikely to become familiar with its operation.

To permit the functionality of a manual override, we provided a set of button-and-light modules, consisting of a single push button and an LED, attached to each device[1] in the room. By physically locating the button with its corresponding device, we need not concern ourselves with the problems of abstract representations inherent to graphical user interfaces. Manual connections can be made simply by pressing the buttons corresponding to the appropriate source and destination. To avoid ambiguity, the order of source and destination button presses is normally important only in cases where both devices are input and output-capable. However, for consistency, we require that all manual connections be made in the order of source first, destination second.

Breaking connections is handled by connecting a destination device to a special module known as the *trashcan*. While the semantics of this operation are, strictly speaking, in contradiction with our "source-first" design, the operation seems to be more sensible to users in this manner, as it corresponds to the physical analogy of dropping an object into the trash. Pressing the same module button twice in succession causes a mirror connection[2] to be made if such a connection is possible.

To illustrate by example, suppose we wish to see and hear a remote participant on *monitor2*, and provide this person with the output of our document camera. If this were not the default behavior of the Reactive Room, then pressing the button associated with the remote participant and the button associated with *monitor2* would establish the first connection. The second connection would be formed by pressing the document camera button and the remote participant button.

The button-and-light modules were originally designed as prototype tools. We considered the need for the presenter to walk around the room in order to establish non-default connections between various devices too awkward for general use. Our solution was to use a calibrated laser detector to provide the required functionality. As shown in Figure 3, users can simply point to a source and destination device with a laser pointer to establish a connection between the two devices. This approach allows control of the room from any location, without compromising the benefits of a physical device representation, as provided by the button-and-light modules.

Feedback

One of the major concerns we faced in automating the control of a conference room was what would happen when the technology broke down. Without automation, there were already a large number of failure points, most of which left the user helpless and frustrated, with no idea of what had gone wrong. While an attempt to correct the existing condition in this respect would be beyond the scope of this research, it was our intent to avoid the introduction of potential sources of failure that would not offer explanation. Although such information is often insufficient to allow casual users to correct the problem, it at least reassures them that they are not the cause. Aside from the issue of what to do when things go wrong, we were also concerned with the more typical case of the system functioning correctly and conveying relevant state information without a graphical user interface. For example, in a videoconference, how do I know that the videotape I am playing is visible to the remote participants?

The need for such feedback was addressed in part by various audio cues, indicating events such as someone entering the room (either physically or electronically) or potential problems such as a daemon not functioning. A great deal of useful feedback was also

[1] For simplicity, each electronic seat, comprised of a camera, monitor, microphone, and speaker, is assigned a single button-and-light module. A special module without a corresponding physical device is required to represent remote participants.

[2] A mirror connection for an electronic seat is established by routing the camera output directly to the monitor.

obtained simply by offering presenters a video monitor that reflects the view being provided to remote participants.

The button-and-light modules used for manual override operations were designed to provide direct feedback through the use of audio and different light states. A single beep sounds when the user presses the first button, and the associated LED begins flashing, indicating further input is required. When the second button is pressed, the computer makes a connection between the corresponding two devices, so long as the connection is possible and does not violate any system-imposed constraints.[3] At this point, a double beep sounds, and both modules' LEDs turn on, indicating the desired connection has been established. The LEDs remain illuminated until a set period expires. If, however, the connection fails, the LEDs are immediately switched off. The same feedback is provided when the laser pointer, instead of physical button, is used to select devices. The module operations are summarized in the state diagram of Figure 4.

The importance of audio and visual feedback for these operations cannot be overstated. Prior to our introduction of these feedback mechanisms, we often observed the "pedestrian crossing button-press syndrome," in which users would repeatedly press the same button until something happened. Furthermore, in an environment of such complexity, even experienced users require explicit indication of the success or failure of their operations. The combination of audio and visual feedback provided by the button-and-light modules fulfills this need with a minimum of additional equipment.

Adaptability

With invisibility, manual override, and feedback addressed, our design principles offer an approach to technology that may result in systems that are truly "walk up and use." However, we have so far ignored the differing user requirements and expectations of the technology that critically influence its desired behavior. Rather than being treated identically, users may require different default configurations and reactions to user-initiated operations. This brings us to the fourth principle, namely, how do we make the system learn the characteristics of different users, and adapt to suit their requirements? If two people have different expectations as to how a system should behave, then ideally, the system will respond differently to each of them.

To provide an adaptive mechanism to the room, we drew on the ideas of programming by example [6, 7]. When a user establishes a non-default connection or destroys a default connection through the use of the button-and-light modules or the laser pointer, the Reactive Room records this action. If the same action is repeated a certain threshold number of times, the room issues an audio alert and changes its default behavior so that this action will be taken under similar circumstances in the future. A calendar daemon that handles room bookings keeps track of who is using the equipment, so that the new actions can be saved in the preference file corresponding to the appropriate user. In this manner, different users can tailor the behavior of the room as appropriate to their requirements.

Augmenting Rooms for Telepresent Users

Our continuing research explores how a similar application of design principles can benefit telepresent users, thereby empowering them with more effective interaction with both the people and technology in our environment.

By virtue of their location, telepresent participants are ordinarily limited to the view provided by a stationary video camera. In essence, their vision is controlled by a second party, typically the conference presenter, who determines which camera will provide output to the remote site. To overcome this limitation, we adopted a solution that allows the remote participant far more control over the received view, yet requires no additional equipment beyond what is already required for videoconferencing, namely a video camera and monitor. This approach involves the Virtual Window concept [4], which uses the video image of the remote participant's head to navigate a motorized camera locally. The remote monitor is treated as a window through which the local room can be viewed. By applying a head-tracking algorithm to the remote video signal, we can determine the position of the user's face in relation to his or her monitor. This position is then used to drive a motorized video camera locally. When the user leans to the left or right, or moves up or down, the camera pans or tilts accordingly (see Figure 5). The zoom factor is similarly determined by the user's proximity to the monitor. As the participant moves closer to the monitor, the camera zooms to provide the sensation of moving toward objects in the conference room.[4]

While our experience with this system revealed significantly improved user sense of engagement in meetings, it also reinforced the shortcomings of the audio-visual channel. When the camera was focused on a small area, the loss of global context or peripheral

[3] For example, it is possible to connect the document camera to the VCR, but not vice versa, since the document camera cannot be a video destination.

[4] The analogy to a window may be somewhat inappropriate in this respect, since moving closer to a physical window results in the field of view increasing rather than decreasing.

awareness often made the user unaware of important activity taking place out of view. To provide users with such context in addition to a detailed, high-resolution view, we investigated the simultaneous display of a detailed view, linked to its position within a wide-angle (global) view [11].

Conclusions and Lessons Learned

Since the system was in flux during development, only ad hoc observational evaluation has thus far been performed, but we anticipate objective user studies in the near future. Early feedback encourages our belief that the Reactive Environment approach offers great promise. Both experienced and novice users have successfully used the Reactive Room to give videoconference presentations, after only a few minutes of explanation. A representative comment made by a remote participant after a presentation mediated by the Reactive Room, is worthy of note: "I want to congratulate whoever was operating the equipment during that meeting. Everything seemed to switch at just the right time."

Of course, there were several occasions when the reactive technology did not behave ideally. Perhaps the most obvious example of this was when presenters forgot to remove a page from the document camera. In this case, participants continued to receive a view of the document, long after it had ceased to be appropriate. However, most of the concerns voiced by users tended to be related to other factors. Two important issues were image quality (participants receiving a blurred view when the presenter was pointing at a document) and inappropriate positioning of cameras (participants watching an empty seat in the picture-in-picture view when the presenter was standing beside the VCR, playing a video clip). Hopefully, these problems can be addressed by increased bandwidth and the use of automatic presenter-tracking cameras.

Extending the UbiComp paradigm, Reactive Environments break through the barriers of traditional keyboard-and-mouse computing and offer an intuitive way of interacting with our surroundings. We no longer have to be baffled by technology and frustrated by confusing user interfaces. We have seen that useful background processing can be carried out by context-sensitive reactive systems, thereby hiding the user interface and facilitating the control of complex technology. Our hope is that this work will stimulate and influence further research, helping to promote alternatives to unnecessarily complex interfaces in the design of future technology.

Additional information on the Reactive Room, including video demos and a live video feed is available from www.dgp.toronto.edu/~rroom

References

1. Buxton, W. Living in augmented reality: Ubiquitous media and reactive environments. In K. Finn, A. Sellen and S. Wilber, Eds. *Video Mediated Communication*, Erlbaum, NJ. To be published.
2. Cooperstock, J.R., Tanikoshi, K., Beirne, G., Narine, T., and Buxton, W. Evolution of a reactive environment. In *Proceedings of Human Factors in Computing Systems CHI '95*, (Denver, CO, May 7–11), ACM Press, NY, 1995, pp. 170–177.
3. Elrod, S., Hall, G., Costanza, R., Dixon, M. and Des Rivieres, J. Responsive office environments. *Commun. ACM 36*, 7 (July 1993), 84–85.
4. Gaver, W., Smets, G., and Overbeeke, C. A virtual window on media space. In *Proceedings of Human Factors in Computing Systems CHI '95*, (Denver, CO, May 7–11), ACM Press, NY, 1995, pp. 257–264.
5. Kellogg, W., Carroll, J., and Richards, J. Making reality a cyberspace. In M. Benedikt, Ed., *Cyberspace: First Steps*, MIT Press, 1991.
6. Lieberman, H. An example-based environment for beginning programmers. In *Artificial Intelligence and Education*, R. Lawler and M. Yazdani, Eds. Ablex, NJ, 1987.
7. Myers, B. *Creating User Interfaces by Demonstration*. Academic Press, San Diego, 1988.
8. Riesenbach, R. The Ontario telepresence project. In *Conference Companion of Human Factors in Computing Systems CHI '94* (Boston, MA, April 24–28), ACM Press, NY, 1994, pp. 173–174.
9. Vicente, K. and Rasmussen J. The ecology of human-machine systems II: Mediating "direct perception" in complex work domains. *Ecological Psychology 2*, 3 (1990), 207–249.
10. Weiser, M. Some computer science issues in ubiquitous computing. *Commun. ACM 36*, 7 (July 1993), 75–83.
11. Yamaashi, K., Cooperstock, J.R., Narine, T. and Buxton, W. Beating the limitations of camera-monitor mediated telepresence with extra eyes. In *Proceedings of Human Factors in Computing Systems CHI '96* (Vancouver, BC, April 13–18), ACM Press, NY, 1996, pp. 50–57.

Jeremy R. Cooperstock is a visiting researcher at the Sony Computer Science Laboratory, Tokyo, and an assistant professor in the department of electrical engineering at McGill University, Montreal, Canada.

Sidney S. Fels (fels@mic.atr.co.jp) is a visiting researcher in the Media Integration and Communications Research Laboratories of Advanced Telecommunication Research (ATR), Kyoto, Japan.

William Buxton (buxton@aw.sgi.com) is Principal Scientist-User Interface Research, at Alias|Wavefront and an associate professor in the department of computer science at the University of Toronto.

Kenneth C. Smith (eesmith@ee.ust.hk) is a professor of electrical and computer engineering, mechanical engineering, computer science, and library and information science at the University of Toronto and the director of computer engineering at the Hong Kong University of Science and Technology.

Many people have contributed to this project, most notably, Kimiya Yamaashi and Koichiro Tanikoshi of Hitachi Research Laboratory and Thomas Scheer of the Fachhochschule Ulm. We would also like to thank Hiroshi Ishii of the MIT Media Lab for his many helpful suggestions and Bill Gaver of the Royal College of Art, UK, for providing us with the idea of using the Virtual Window, a concept that originated at the Delft University of Technology, The Netherlands.

This research has been undertaken as part of the Ontario Telepresence Project, with support from the Ontario Government Centres of Excellence, Xerox PARC, Bell Canada, Bell Northern Research, Alias|Wavefront Research, Object Technology International, Sun Microsystems, Hewlett Packard, Hitachi Corp., NTT, the Arnott Design Group, Adcom Electronics, IBM Canada, and the Natural Sciences and Engineering Research Council of Canada. This support is gratefully acknowledged.