REACTIVE ENVIRONMENTS AND AUGMENTED MEDIA SPACES

by

Jeremy R. Cooperstock

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Electrical and Computer Engineering
University of Toronto

REACTIVE ENVIRONMENTS AND AUGMENTED MEDIA SPACES

Jeremy R. Cooperstock, Ph.D. 1996
Department of Electrical and Computer Engineering
University of Toronto

Abstract

With the proliferation of computers and the relentless demand for powerful new systems, our society is quickly reaching a point in which we are overwhelmed by the complexity of the technology around us. The problem is that while computer functionality continues to grow, our mental capacities remain relatively unchanged. As a result, attempts to incorporate computers into our lives often leave us frustrated by our inability to interact with them. To cope with this problem, an alternative paradigm is required, one in which the task, rather than the operation of tools to accomplish it, is central in human-computer interaction.

The paradigm we propose is that of Reactive Environments. Rather than require explicit human communication with the computer, the technology makes use of input from a number of sensors to determine what the user is trying to accomplish. Based on the task and the associated context, the environment carries out useful background processing in order to assist the user. This concept marks a dramatic departure from the primarily foreground, machine-oriented human-computer interaction of the present. Instead, with Reactive Environments, we are offered an opportunity to interact with our surroundings while ignoring the computer.

To develop this concept further and determine what important issues were involved, we designed and constructed a prototype Reactive Environment in a videoconference room setting, ensuring that users could operate all of the equipment without explicitly interacting with a computer. Once this was accomplished, our next step was to make the

room electronically accessible, and augment the environment further so that its potential could be exploited by users who were not physically present.

These efforts led to the emergence of several diverse technologies, including a Smart Light Switch, a laser pointer as a remote control, and an Audio/Video Server Attendant, in addition to the Reactive Environment itself. As a result of this research, we have also been able to realize a number of substantial improvements to the sense of engagement available to users of videoconferencing systems.

Acknowledgments

In 1992, I had the tremendous fortune of meeting K.C. Smith, who was interested in my ideas of distributed cooperative systems, and agreed to supervise my Ph.D. research. The following year, K.C. introduced me to Bill Buxton, who sparked my interest in human-computer interaction and media spaces.

The inspiration for my research grew out of Bill's ideas on ubiquitous computing and a discussion with Kevin McGuire of Object Technologies Inc. during the summer of 1994. From these seeds, and my background in intelligent robotics, grew the concept of Reactive Environments. To help me realize an implementation of this concept, Bill gave me the best research environment one could imagine: a playground filled with toys and no rules to follow, beyond being productive.

A project of this type is by its very nature, multidisciplinary, involving design, implementation, testing, and refinement, of a large number of components. As a result, this led to collaboration with numerous colleagues and built upon the work of Tom Milligan, Tracy Narine, and Dominic Richens of the Ontario Telepresence Project. We were also fortunate to have the generous contributions of Bill Gaver, of the Royal College of Art, UK, who provided us with the Virtual Window concept, an idea that originated from his work with Gerde Smets and Kees Overbeeke at the Technical University of Delft, the Netherlands.

While the ideas of Reactive Environments were refined, I was joined by a number of other researchers who helped in the implementation of many of the systems that made up our prototype effort. In particular, Koichiro Tanikoshi of Hitachi Research Laboratory and Thomas Scheer of the Fachhochschule Ulm, provided countless hours of assistance and stimulating discussion as we built the Reactive Room. Koichiro implemented the first of our Reactive Room components, the document daemon, and wrote the serial interface driver to the motorized cameras. Sidney Fels, a former graduate student of the Neural Networks group in Computer Science, also provided invaluable technical support and helped shape the design philosophy that guided our efforts. In addition to implementing

the signal daemon interface, Sid worked with me on the original Smart Light Switch design, and wirewrapped our first prototype.

At the start of 1995, Koichiro began the very exciting Audio/Video Server Attendant project along with the assistance of several undergraduate students, and on his return to Japan, passed on the reins to Anuj Gujar, a Master's student who had just joined our group. While the Audio/Video Server Attendant is largely the work of others, Anuj in particular, it serves as an interesting and relevant illustration of the ideas of this thesis. Kimiya Yamaashi took Koichiro's place, and we soon began collaborating on new efforts, including the Extra Eyes system and the World Wide Media Space. Both Koichiro and Kimiya were wonderful scientific collaborators as well as friends, and I have been constantly impressed by their energy and enthusiasm. I am deeply indebted to them for their contributions to my research.

Along the way, I have been offered invaluable suggestions and feedback from many colleagues, including William Hunt, Marilyn Mantei, John Tsotsos, Kim Vicente, and Shumin Zhai of the University of Toronto, Rich Gold, Roy Want, and Mark Weiser of Xerox PARC, Abigail Sellen of Rank Xerox EuroPARC, Wendy Kellogg of the IBM T.J. Watson Research Center, Masayuki Tani of Hitachi Research Laboratory, and Hiroshi Ishii of the MIT Media Lab. I thank these individuals for sharing their ideas with me.

I would also like to express my gratitude to several more people, without whose help this research would have taken twice as long to complete: Victor Ng and Alex Mitchell, System Administrators for the Dynamic Graphics Project, worked many hours of overtime maintaining computers that defied all mortal efforts to keep them alive. Mike Ruicci and Bernie Maillard of the University's CS Lab, always came through with a custom-made cable or an ingenious solution to fit a square peg in a round hole, with a touch of humour on the side. Ben Gamsa, a walking set of UNIX `man` pages, solved so many of our technical problems that `talk ben@atlas.sys` soon became an alias for help. These individuals endured my relentless interruptions of their own work and unfailingly offered their assistance, well beyond the call of duty.

Finally, I wish to extend a special thanks to my advisors and mentors, K.C. Smith and Bill Buxton. K.C. gave me the freedom to pursue my own research directions, and Bill provided the environment and creative stimulation in which this research was able to flourish.

This research was undertaken as part of the Ontario Telepresence Project. Support has come from the Government of Ontario, the Information Technology Research Center of Ontario, the Telecommunications Research Institute of Ontario, the Natural Science and Engineering Research Council of Canada, Hitachi Ltd., Bell Canada, Xerox PARC, British Telecom, Alias|Wavefront, Hewlett Packard, Sun Microsystems, the Arnott Design Group, and Adcom Electronics. Funding for my studies was also provided by scholarships from the Walter C. Sumner Memorial Foundation and the J. Edgar McAllister Foundation. This support is gratefully acknowledged.

For my parents.

Table of Contents

List of Figures

important events, along with a hotkey to switch to the relevant view. 19

List of Tables

Chapter 1 Introduction

... rather than being a tool through which we work,
and thus disappearing from our awareness, the
computer too often remains the focus of attention.
MARK WEISER [82]

As 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 an electronics-free home or working in an office without computers, few of us have truly mastered 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 instruction manuals and confusing user interfaces.

1.1 Background

Designers have been trying to make the computer easier to use or more "user-friendly" ever since its inception. The last two decades have brought us the notable advances of keyboard terminals, graphics displays, and mice, as well as the graphical user interface (GUI), 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 their environment 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) [82] and Augmented Reality [84][10]. 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.

As Weiser laments the status quo [82], we too see how interaction with the computer interferes with our activities rather than enhancing them. Instead, with UbiComp, we are offered an opportunity to interact with computers in human terms. The benefits of this approach are twofold. First, embedding computational power in everyday objects means that people can think about and interact with their environment naturally, without the cognitive burden of figuring out which button to press or which command to type. Second, as a consequence of this natural interaction, systems ranging in complexity from a basic microwave oven to an elaborate videoconference environment, and hopefully even beyond, could become “walk up and use.” As a guiding principle, users should require little or no training in order to take advantage of the technology.

Lowering the visible complexity to improve the usability of systems is not a new idea. Examples abound of point-and-shoot cameras and anti-lock brakes, in which additional circuitry is added so that an understanding of the full functionality of the underlying layers is not a prerequisite for use. While the notion of correcting the problems of technology with more technology may seem counter-intuitive, it is clear that useful background processing can be performed by correctly deployed systems.

The research described in this thesis explores an extension of this philosophy through the design, implementation, and evaluation of a collection of cooperative systems. While the promise of technology based on UbiComp is truly exciting, we believe that this approach will succeed only if the design of these systems takes into account the human factors governing their use. The factors that we consider important for usable technology include

invisibility as described above, a seamless manual override mechanism, provision of feedback to the user, and adaptability to different requirements and 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. This thesis offers a set of design principles that address the limitations of previous work. It is not our intent to provide experimental data or a detailed analysis of these problems, but rather, to shed light on the problem space and provide qualitative evidence that our approach is plausible, interesting, and worthy of further exploration.

1.2 Why Videoconferencing?

Put simply, the state of the art in videoconference environments provides us with a superb example of technology gone awry. We are given many wonderful tools, enabling geographically disparate participants to meet, discuss, collaborate, and educate. But control 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 [68] and from observations of users with various room control systems, we have seen meeting breakdowns occur again and again. Simply giving a presentation is difficult enough, but the additional burden of managing control of 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 the room lights or playing a tape on the VCR. 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 [9]. It is our contention that such tasks pose difficulties because appliances such as the VCR have been designed without context-sensitive interoperability in mind. Addressing this deficiency is the cornerstone our approach. Until all of the devices in our videoconference room, or appliances in our home, or applications software on our

computers, can be interfaced and perform in coordination, our frustration with the technology is bound to persist.

1.3 Problems with the State of the Art

In discussing the limitations of current approaches to the control of technology-rich environments, it is worthwhile to consider the taxonomy of conscious (attentional) versus subconscious (automatic) levels of human information processing [77]. While each form of processing has its respective strengths and weaknesses, the characteristics of interest for this discussion are that the former is sequential and effortful while the latter is parallel and effortless. Hence, a design goal of complex systems should be to exploit the efficiency of the subconscious processor and minimize our requirements for the conscious processor [77]. However, this is only possible for those tasks in which the human has gained considerable experience with the particular situation.

Returning to the problem of conference room control, we can identify certain tasks such as turning on the lights and playing a video tape as being amenable to subconscious information processing. These are skills that we have mastered through our day-to-day experiences. On the other hand, mixing audio sources and combining video signals so that a presenter and remote participant can communicate while a tape is being played is not an activity with which most of us are familiar. State of the art control system interfaces such as ADCOM's iRoom [1], shown in Figure 1, AMX's AXCESS systems [2], and the Telepresence Desk Area Network, shown in Figure 2, attempt to remedy this problem by providing a human-computer interface that allows the user to select from a number of devices as input sources. The control system then configures the equipment so that the audio and video signals are routed appropriately.

Such interfaces reduce, but do not eliminate, our reliance on the conscious processing system. Overall, they tend to exemplify rather than solve the problems of current technology. Presenters often require a configuration of equipment that the control system does not provide, for instance, 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

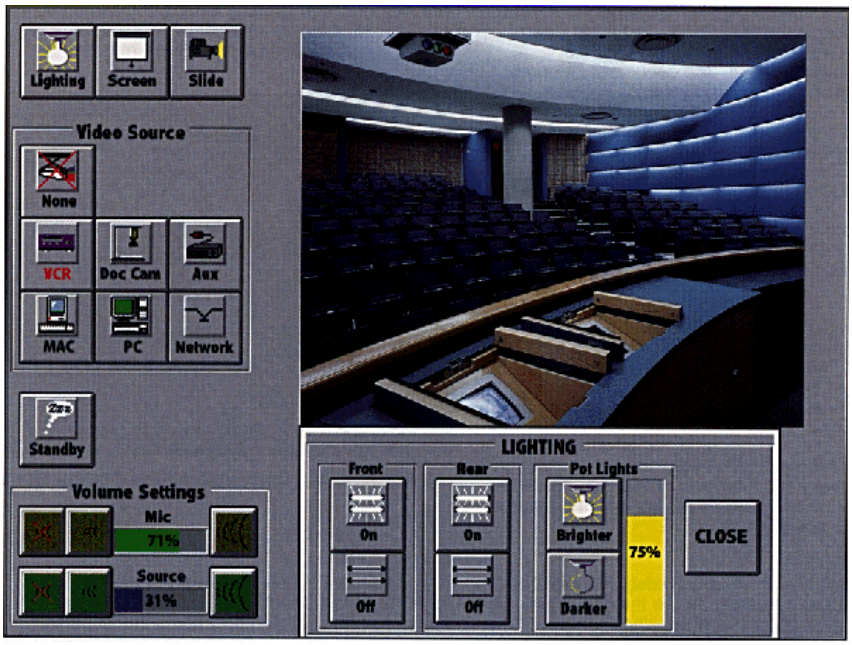

FIGURE 1 The ADCOM iRoom™ touch-screen interface. Photo courtesy of ADCOM Inc.

and operations they represent, and vice versa [78]. For example, the “play-video” label in Figure 2 is described by “Send VHS + room to visitor” but what view of the room will be provided and will the visitor continue to hear me while I explain the video clip? 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 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

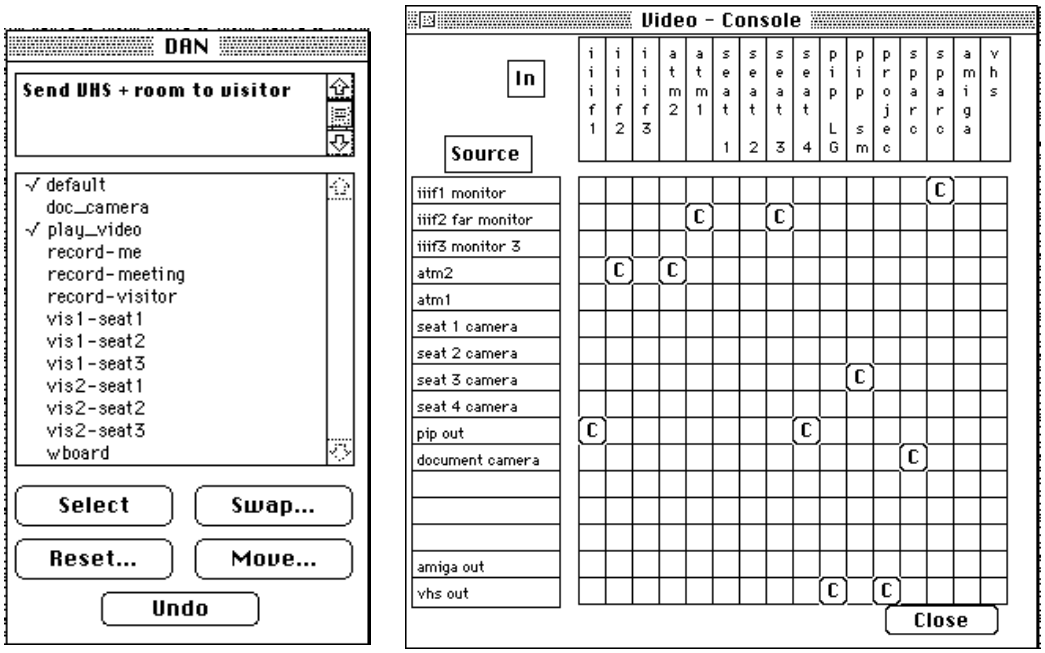

FIGURE 2 The Desk Area Network (DAN) is comprised of a menu of presets (left) and an electronic patchbay (right).

machine. As a result, we cannot “walk up and use” the computer, but must be trained in its operation. Rather than limit ourselves to this restrictive user interface paradigm, we believe that users should be able to interact with their environment using whatever tools and techniques with which they are comfortable. By hiding the interface, UbiComp offers the possibility of intuitive, or subconscious, interaction, and thus, a significant reduction both in the need for user training and in the cognitive demands the technology places on the presenter.

1.4 Thesis Roadmap

The first part of this thesis, contained in Chapters 2-4, provides an in-depth study of our research efforts in this direction, culminating in the implementation of a computer-augmented environment called the *Reactive Room* [20]. To provide an interesting testbed of technologies, the Reactive Room is a fully equipped and operational videoconferencing environment. The room offers users an appropriate, automatic (invisible) configuration of equipment in response to their actions, as well as the capabilities of full manual override,

feedback, and adaptability. Beyond the general scope of exploring these important issues, the immediate intent of the Reactive Room was to increase the effectiveness of human-human communication in a context where the complexities of technology ordinarily impede natural interaction.

However, in constructing a powerful, yet easy to use computer-augmented environment, we had inadvertently widened the gap between physically present and electronically present, or "telepresent" users. The Reactive Room freed local users from the burden of controlling complex videoconference equipment, but offered little in terms of empowering remote attendees with the ability to select their own views, or to interact with devices that were beyond their physical reach. Furthermore, our initial research had ignored the limitations that a narrow channel of communication imposes on human interaction. Due to the nature of the electronic medium, telepresent users are deprived of the wide field, rapidly directable stereoscopic vision and highly sensitive binaural audio that we take for granted in face to face discussion. The critical role that these abilities play in our social interaction prompted the next phase of research, which was concerned with engineering solutions to the tasks of empowering remote users and improving their sense of presence over an audio-visual channel. This research is discussed in Chapter 5 of the thesis.

Finally, in Chapter 6, the contributions of this thesis are summarized and future research directions are explored.

1.5 Literature Review

The foundations of this research come from a number of diverse disciplines, including Human Computer Interaction (HCI), Computer Supported Cooperative Work (CSCW), intelligent robotics, cognitive psychology, and biological vision.

A prominent theme in this work, in which most of our design efforts were focussed, is that of videoconference environments and media spaces. Early work in this domain includes Colab [72], and research at Rank Xerox EuroPARC [11], including Gaver's RAVE system [28]. These initial projects shaped the Media Spaces project at Xerox PARC [4], the Cavecat Project [50], and the Ontario Telepresence Project [68], which provided the architectural backbone for this thesis. Our efforts to improve the sense of engagement of

participants in a media space built on the pioneering work of the MTV studies [29][35] as well as the Virtual Windows concept [60][30].

The starting point for this thesis can be found in Ecological Interface Design (EID) [77]. This framework attempts to improve human-machine interaction by allowing the user to control and observe a system at multiple levels, either directly or indirectly, as required. Building on Gibson's theory of affordances and the ecological approach to perception [31], Vicente and Rasmussen develop the idea that properties of the environment that allow actions, or *affordances*[1] necessary for effective system control should be built into the interface. The motivation for this design comes from observations of operators' natural tendency to utilize subconscious processing and the realization that greater performance efficiency and flexibility may arise when a process can be controlled directly [66]. Providing this capability through the interface, while allowing the user to exert higher levels of control when necessary, are the keys to this approach.

One technique advocated by Rasmussen for coping with complexity is the means-end hierarchy of affordances [67]. This model allows goal-directed organisms to deal effectively with the *degrees of freedom problem* for complex systems, by reducing the dimensionality of their description of the system [78]. Thelen summarizes the degrees of freedom problem, originally posed by Bernstein [5], as follows:

> *"How can an organism with thousands of muscles, billions of nerves, tens of billions of cells, and nearly infinite possible combinations of body segments and positions ever figure out how to get them all working toward a single smooth and efficient movement with invoking some clever 'homunculus' who has the directions already stored?"* [73]

The means-end hierarchy essentially provides a representation of all possible means to achieve all possible goals, taking any constraints into consideration. As one moves up the

1. Norman defines the term *affordance* as the perceived and actual properties of an object that determine how the object could possibly be used [59].

hierarchy, the detail of lower levels is hidden. This technique is useful as an analytical or modelling tool, and it would be interesting to apply it to studying the videoconference tasks of Appendix A in more detail.

From our perspective, the most important aspects of an ecological interface are those relating to indirect observation and indirect control. Because the surfaces of observation and action are physically separate, conventional interfaces require the user to perform mental mappings from displayed information to higher-level concepts and from human intentions to commands in the computer's vocabulary [77]. EID proposes, instead, that such translation be performed by the computer. This idea is similar to the concept of Ubiquitous Computing [82], which attempts to hide the computer from our awareness. While EID makes the use of the interface a subconscious activity, UbiComp goes further and seeks to remove the explicit interface altogether. The common theme is to improve our abilities to interact with technology by reducing the cognitive load inherent in its use.

UbiComp is based on the two principles of ubiquity and transparency. This means that the technology and computational power should be highly distributed, yet integrated in the environment and non-intrusive to the point where they are effectively invisible. To develop these ideas, Weiser, the principle architect of UbiComp, took the approach of creating a number of computing artifacts as replacements for objects found in our offices and homes [82]. These artifacts took the form of wall-sized interactive surfaces (boards), scrap paper (pads), and Post-it notes (tabs) [80]. Each artifact offers the same affordances as its everyday counterpart, but with the addition of computational power, memory, and communication networking, opens the door to a wealth of new applications. However, our own work with these devices confirmed Weiser's prediction that they were unlikely to achieve invisibility. This is largely due to the fact that their use requires rules of interaction that are dictated by the computer interface rather than our everyday skills.

Weiser's vision of the future of computing [81][82] can be seen in its influence on the Responsive Office Environments of Elrod et al. [25] and Buxton's philosophy of Ubiquitous Media [10]. Responsive Office Environments were a first attempt to augment a physical environment with temperature, light level, occupancy, and active badge sensors so as to exercise computer control over ventilation, heating, and lighting within the offices. Without elaborating on the details, Elrod et al. demonstrate three of the key design principles advocated by this thesis, invisibility, manual override, and feedback. Using

sensors in the room, the heating, cooling, and lighting can be adjusted automatically. With the PARCTab as an interface, users can manually adjust parameters such as desired temperature, and obtain feedback of current settings [25].

A similar approach was taken earlier by the Real-time Operating system Nucleus (TRON) Project of Sakamura [69], especially in regards to the construction of the Intelligent House, involving the use of close to 1000 computers and completed in 1989. A valuable outcome of this work was the recognition of the inadequacy of simplistic automation schemes. Sakamura illustrates this point by the example of automatic blinds that close when someone is trying to enjoy the outside view. A truly “intelligent” house should wait until the individual has moved to a different location [70].

The differences between these other research projects and our own can be summarized as follows: First, whereas Responsive Office Environments and the Intelligent House respond to state (e.g. ambient light level, presence or absence of an individual in the room), our Reactive Environments react to user activity (e.g. pressing the record button on the VCR or placing a transparency underneath the document camera) in a context-sensitive manner. We note that similar context-sensitive behaviour, albeit on a smaller scale, has also been demonstrated by the desktop systems of Clearboard [39] and the DigitalDesk [83], as well as in other computer augmented environments [43][48].

Second, our emphasis is not on the first order of technological functionality for which we have all acquired considerable experience, and hence, have little difficulty managing, such as turning lights on or off, or adjusting the thermostat, but rather, the far more challenging second order, involving tasks such as dynamically selecting the appropriate audio and video sources to be recorded during a meeting. In our opinion, it is this second order of functionality, dealing with inter-device tasks, or the problem of coordination, where the avoidable complexity lies. Hence, this is the level of control we wish to relegate to the environment.

Third, because the PARCTab and the TRON control of the Intelligent House require the use of interfaces that are not a part of the natural environment, the manual override and feedback mechanisms of these systems require conscious information processing on the part of the user, at odds with our desire for an ecological design.

Another related project is that of the Intelligent Room[2] [74] at the MIT AI Lab, which attempts to interpret human activity in order to facilitate presentations and collaborations in a shared meeting space. This work combines robotics, computer vision, natural language understanding, and intelligent information handling, to provide numerous applications through a seamless human-computer interface, and relies on customizable intelligent agents to provide computational services to the room. It should be noted that our approach differs from this concept of computer as autonomous agent [49], in that the machine is not asked to take on our goals, but rather, to carry out context-sensitive background processing as a means of assisting our efforts to accomplish these goals. In other words, the human remains in charge at all times.

The Intelligent Room is an exercise in UbiComp, aspiring to remove the computer from human awareness. Of particular relevance are the efforts to track a moving presenter (cf. Hunke and Waibel's face-locating system [37]) and automatically switch video output to one of two cameras with a more interesting view. However, careful consideration as to how technology should be deployed in order to assist human performance seems to be lacking, as evidenced by the emphasis on virtual reality displays and multimedia annotation tools. Our experience has been that technology is all too often guilty of imposing rules and constraints on its users rather than conforming to their existing habits of usage while affording increased potential. It is precisely this trend from which we are trying to break free, and in order to do so, we believe that it is first necessary to gain a perspective of the social role that the technology plays in our tasks. Only then can UbiComp be realized as a practical design philosophy and employed throughout our lives.

1.6 Summary

Building on the foundations of EID and UbiComp, this thesis introduces Reactive Environments, a concept that integrates distributed, yet transparent, computational power with seamless manual override and feedback. These capabilities were embodied by our computer-augmented media space, the Reactive Room, which demonstrates automatic, context-sensitive interoperation of multiple, complex devices, in reaction to user activity.

---

2. The Intelligent Room should not be confused with the Smart Rooms project [62] of the MIT Media Lab.

Chapter 2

Evolution of Conference Room Control

Operators err, it seems, in not being able fully to surmount the inadequacies and complexities of the equipment they must use.

CHARLES PERROW [63]

This chapter introduces the domain in which our research efforts were focussed, and outlines the evolution of control systems that were employed in the environment, from a primitive physical patchbay to a state of the art graphical user interface. Throughout this evolution, our guiding concern was to ensure that users of the room can continue to use whatever tools and techniques with which they are comfortable. For example, the electronic document camera can function as an ordinary overhead or slide projector and the traditional whiteboard is still present. The underlying design principle is to reduce complexity by enabling users to interact with the room using existing skills acquired through a lifetime in the everyday world.[1]

The contents of this chapter were originally published in Cooperstock, J., Tanikoshi, K., Beirne, G., Narine, T., and Buxton, W. Evolution of a Reactive Environment. *Proceedings of Human Factors in Computing Systems CHI '95*, Denver, CO, ACM Press, New York, pp. 170-177, May 1995. Permission to reproduce the material here was granted by ACM. The work described is a product of the Ontario Telepresence Project, and was implemented by Tom Milligan, Tracy Narine, and Dominic Richens.

FIGURE 3 The conference room.

2.1 Overview

The conference room that is the subject of this study, pictured in Figure 3, is equipped to support activities such as:

- *Videoconferencing* from the front of the room (permitting remote presentations) or back of the room (permitting remote participants to attend meetings as part of the audience)
- *Video playback* from both local and remote sites
- *Meeting capture* via videotape
- *Electronic collaborative whiteboard* that can be driven locally or remotely, such as described by Elrod et al. [24]
- *Support for computer demonstrations*, run either locally or remotely

1. By this, we mean a world containing such technological artifacts as light switches, VCRs, and televisions, as well as the overhead projector found in conventional conference rooms.

- *Overhead projection* using a video document camera capable of being seen locally and remotely, as well as

The equipment includes several cameras and monitors, a VCR, a digital whiteboard, pictured in Figure 4, and an electronic document camera, shown in Figure 5, which

FIGURE 4 The digital whiteboard in use. The design being sketched is visible to people in the room and to the video attendee, who appears on the small monitor in the left of this figure.

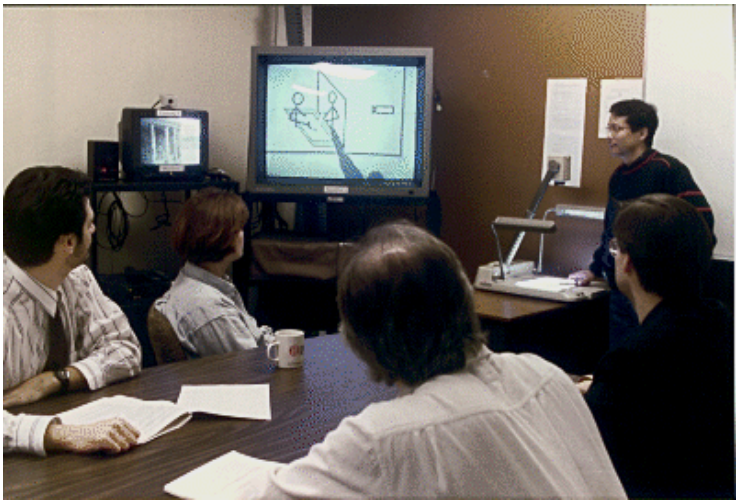

FIGURE 5 The speaker (top-right) is illustrating a diagram on the document camera. The document is displayed on the large video monitor.

replaces the standard overhead projector typically found in such environments. The digital whiteboard is actually a large data monitor driven by a computer running a draw or paint application. Because of the hardware configuration, users of the whiteboard write or draw with a light pen instead of a mouse. The output of these devices can be displayed on any of the monitors in the room and sent to electronic visitors as well.

Unfortunately, as the amount of equipment and potential functionality increases, so does the complexity of its operation. Control of the equipment is, in itself, a relatively straightforward task. However, the additional requirement of interacting with a user interface in order to specify which device or camera view is displayed on each monitor places a significant burden on the user.[2] With conventional approaches, presenters must handle multiple remote controls and explicitly establish connections between various devices. Consider, for example, the possible problems of switching from an overhead slide to a video tape sample. On what screen does the video appear? Is it the same one as for the overhead? How is the connection established? Users often complain that control of the equipment is confusing and overly complex. Presenters must either interrupt their talk to manipulate the environment, or simply avoid using the technology because it requires too much effort.

Even if all of these issues are resolved for the local audience, what does the remote person see, and how can one be sure that this is what the presenter intended? These, and a myriad of related problems confront the baffled user. We have not even addressed the basic issue of how the user turned on all of the equipment in the room, initially. Where are all the switches and controls? While usage studies indicated that we were trying to incorporate the correct functionality and deploying the components in more or less the right locations, our work had not really begun. Regardless of the tremendous potential existing in the room, if the complexity of its use was above the threshold of the typical user, the functionality simply did not exist in any practical sense.

In the remainder of this chapter, we describe the history of the conference room control system as a series of iterations, culminating in the rationale for the new approach of reactive environments. For each iteration, we discuss the design motivation, the solution

2. In the case of the VCR, the problem is even worse. When recording a meeting, the user must also specify the source (or possibly two sources) of input.

taken, and evaluate the results. It should be noted that our evaluation was informal, based on personal experiences and anecdotal evidence.

2.2 Initial Environment

Our initial room design was intended to allow remote attendees to participate in meetings. The room was equipped with a camera, monitor, microphone and speaker at the front of the room. This equipment functioned as a video surrogate in an existing media space. In short, it corresponded to most basic videoconferencing rooms.

Using this implementation, it was realized through "breakdowns" in meetings that modifications were required. For example, due to the placement of the video surrogate at the front of the room, remote attendees often spent the whole meeting watching the back of the presenter's head. At the same time, local attendees were distracted from the presenter due to the inappropriate location of the remote participant(s) at the front of the room, in the speaker's space. (Note that this situation is the norm in an embarrassingly large number of videoconferencing rooms.) It was clear that different locations of video surrogates were needed for the different social roles of meeting attendees.

2.3 First Iteration

The motivation for the first iteration was to allow remote participants to either present, attend or participate in videoconference meetings. This design involved the addition of three video surrogates at the back of the room. These surrogates were placed at the same height as the conference room table so that remote attendees would be perceived as sitting around the table. Again, an existing media space was used to support this functionality. This design worked well when remote participants were in the appropriate place. However, these users could not select their own positions within the room and it was difficult to move them from one location to another, such as when a remote attendee needed to change roles and become the presenter.

At this stage, the user interface consisted of the set of connections between devices themselves, that is, a hardware patchbay. This meant that in order for a presenter to realize a goal, such as "record my presentation," it was first necessary to determine which devices to activate, and then physically make the appropriate connections between them. Figure 6

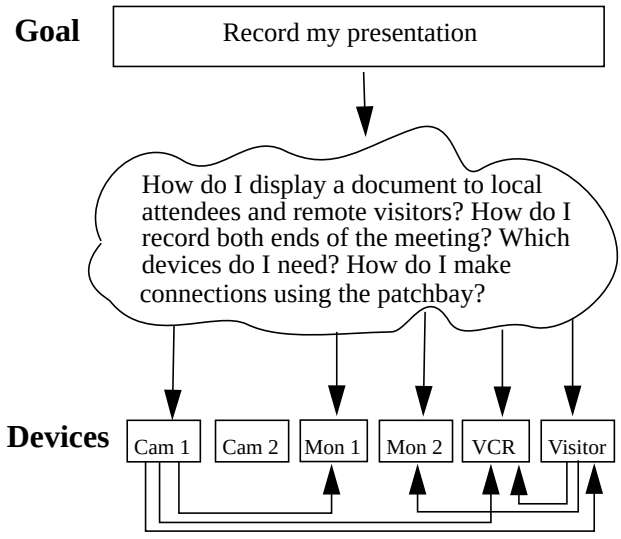

FIGURE 6 Complexity of First Iteration Interface. The inter-device lines represent physical patchbay connections that the user was required to make.

depicts the user interaction with various devices. The cognitive effort required by the user in order to achieve the high-level goal through direct device manipulation is considerable.

2.4 Second Iteration

The next step was the incorporation of an $n \times n$ software-based matrix to implement the patchbay. This is shown in Figure 7. Each row corresponds to a source device (e.g. camera, VCR output) and each column to a destination (e.g. visitor view of room, video monitor). By clicking the mouse on entry (i, j), an audio or video switch would make a connection between source *i* and destination *j*. This resulted in considerable time savings, because the user could now establish connections through a graphical user interface, rather than physical wire. However, as depicted in Figure 8, since the user was still responsible for all device connections, the cognitive effort remained unacceptably high. A major component of this effort was the need to map abstract labels to the physical devices they represent. It should be noted that this abstraction problem confronts all such user interfaces, regardless of whether they represent devices by graphical icons or textual descriptions.

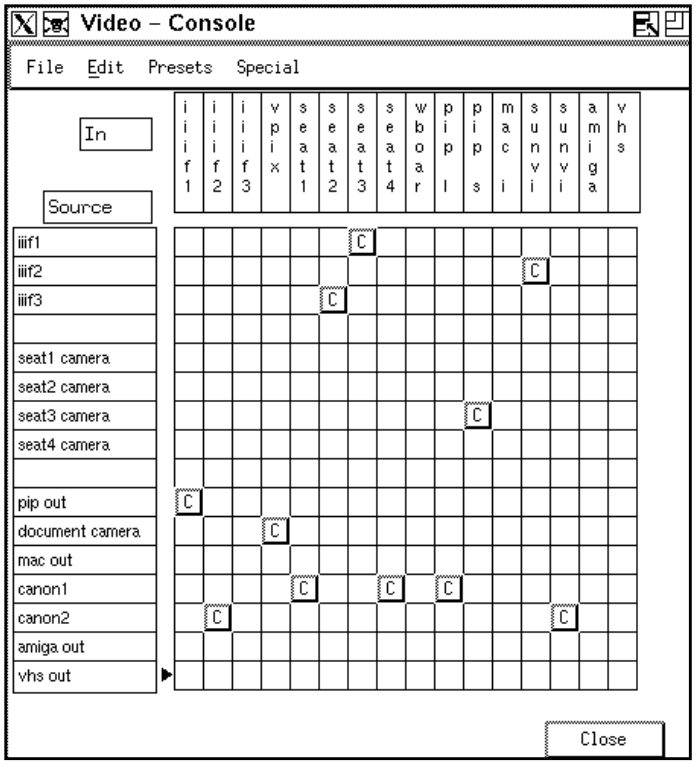

FIGURE 7 Matrix-based interface for controlling equipment (virtual graphical patchbay). The iiif1, iiif2, and iiif3 labels correspond to three potential electronic attendees.

2.5 Third Iteration

To make the system more efficient and reduce the cognitive burden associated with matrix representations, a provision was added that allowed administrators to create a list of user presets, pictured in Figure 9. This list corresponds to the typical set of room configurations required by users of our conference room. As illustrated in Figure 10, a strong incentive for the development of presets was that they allowed the user to break down a goal into a number of fairly straightforward sub-goals, without concern for the representations of individual devices [78].

Each preset invoked a series of connect and disconnect operations within the audio and video patchbays, as shown by the example in Figure 11. All existing connections not

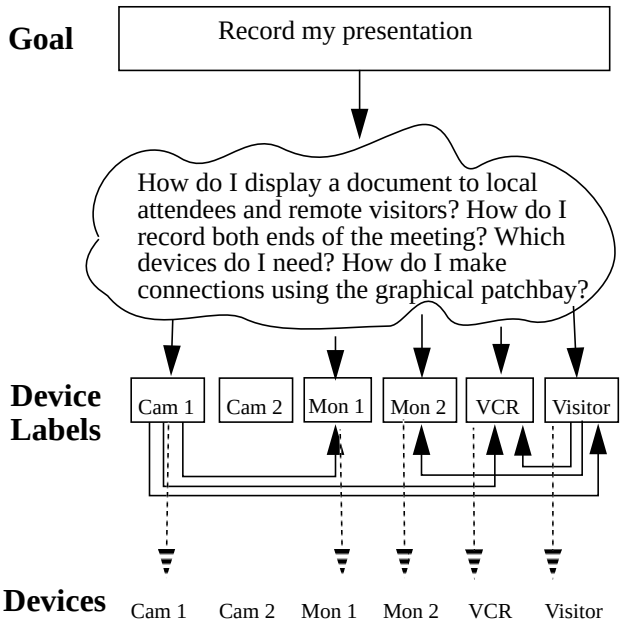

FIGURE 8 Complexity of Second Iteration Interface. The solid lines represent user interaction and the dashed lines represent tasks performed by the user interface. Note that the user is still responsible for inter-device connections, now made through the graphical user interface.

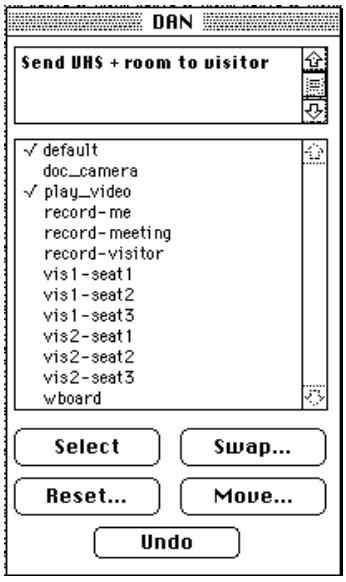

FIGURE 9 Presets Menu (DAN). As shown, the VCR output is currently being viewed.

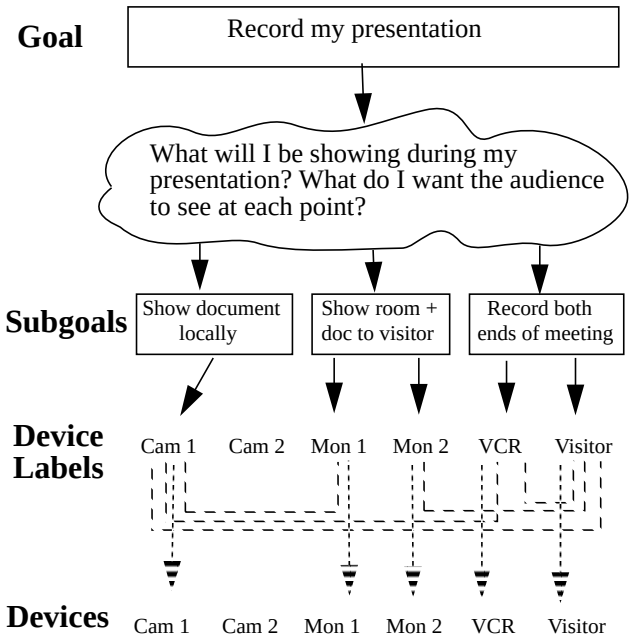

FIGURE 10 Complexity of Third Iteration Interface, using presets. Now, the user can ignore details of device representation and location, However, presets can be confusing, especially when there is more than one way to accomplish a subgoal.

affected by the preset were left intact. This allowed users to quickly switch from a face to face discussion with a remote visitor, to displaying a document, to playing a video tape, and back to a face to face discussion, with only three user interface selections. In order to reverse the effects of potentially unintended selections, or simply to restore the configuration to its previous state after concluding the use of a particular device, an "undo" operation was added. A "reset" function was also provided to restore the system to a default state.

We found that while these provisions simplified control of the patchbay, subtle distinctions existed between various presets and users could not decide which ones to choose. A more serious issue was the persistence of the distraction problem. Despite their relative simplicity, selection of presets required an interruption of the meeting and interfered with the smooth flow of presentations. In fact, many users would not operate the system[3] or forgot to operate it at the required times. Meetings involving the use of presentation equipment invariably broke down at some point, especially when electronic visitors were in attendance, because the output of the device being used was not visible to some or all of the participants.

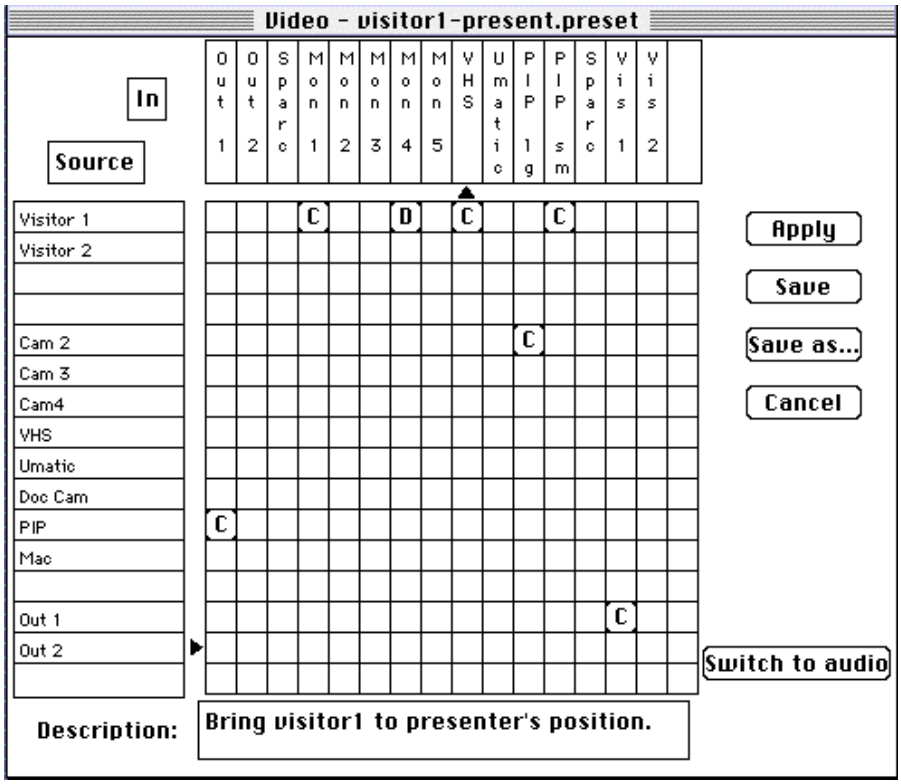

FIGURE 11 Preset configuration for moving a visitor to the presenter's position. The 'C' and 'D' entries correspond to connect and disconnect operations, respectively.

At this stage, our system was essentially the same as many commercial room-control systems, such as ADCOM's iVue [1] and AMX's AXCESS systems [2], and suffering from similar problems. Foremost of these was the need to interact with the computer, an inherently unnatural, distracting, and often time-consuming activity with respect to the task of giving a presentation.

3. To be fair, some users of the conference room did not even know that the control system existed. Feedback from other users suggests that their reluctance to operate the system was a combination of frustration with previous interfaces and a fear of making the configuration worse than it was already.

2.6 Summary

Our implementation of a conference room control system progressed through a series of iterations, from a physical patchbay to a list of high-level presets, equivalent in functionality to many state of the art commercial systems. Throughout this evolution, our goal was to reduce the complexity of operating the technology. However, as each iteration attempted to overcome the limitations of the previous interface, it also introduced new sources of cognitive load, and hence, prevented users from realizing the full functionality of the environment. In order to solve this problem, control of the room had to be automated. This is the point of departure for the work in this thesis. What follows is our contribution, which builds upon this existing base, developed by the Ontario Telepresence Project.

Chapter 3

Design of a Reactive Environment

Everything should be made as simple as possible,
but no simpler.
ALBERT EINSTEIN

Our primary motivation for a new approach to control systems were the goals of reducing both the complexity of the technology and its intrusive nature, as discussed in the previous chapter. Our efforts to do so resulted in the formulation of the Reactive Environment, a facility that supports automatic, context-sensitive configuration of equipment in response to human activity. This concept was based on the underlying assumption that if a human operator is able to infer a user's intentions based on his or her actions, so should an appropriately designed system. In our case, the environment collects background information as context to maintain state and support explicit foreground action [9]. Thus, the cognitive load should be reduced, allowing the user to concentrate on the primary task at hand, for example, giving a presentation, rather than on the secondary task of controlling the equipment.

3.1 Methodology

The problem of constructing an environment that assists, rather than obstructs our tasks, can benefit from an obeisance of a structured design methodology. While the danger exists that the designer may ignore fundamental needs of the end-user, we enjoyed the benefit of

living both roles. Throughout the process of design, implementation, analysis, and redesign, we made use of our system components on a daily basis. Forcing the architect to live in his own house exposed many important issues that might not have been evident either at the start of this research, or during the later stages of refinement. Our methodology was strongly based on iterative design, the process of repeatedly implementing imperfect prototypes, so that their usage could be examined and evaluated informally.

At an early stage, we took an inventory of the skills of our user community, including, at the motor-sensory level, gestural capabilities, speech, and hearing, and at the social level, experience with certain presentation equipment and with speaking in front of groups. We then defined the primary tasks that can be performed easily using existing skills (e.g. turning on the lights, loading and playing a tape on the VCR) and the secondary tasks that were beyond the skill set of most users (e.g. configuring the video patchbay to insert a picture-in-picture view of the presenter while a document is being shown to remote attendees). This methodology of partitioning tasks into a primary set of manageable functionality, and a secondary set that we needed to relegate to the computer, is critical to the generalizability of our work.

Finally, it was imperative that the analysis undertaken throughout the initial deployment of the technology, as described in Chapter 2, be respected. Our augmentation of the environment could not dictate new constraints on the users. Rather, the space-function-distance relationships of the conference room had to be preserved.

3.2 Design Principles

Recognizing the failures of conventional technology from a social, rather than technical perspective, and anticipating potential weaknesses of a naive application of Ubiquitous Computing (UbiComp) precepts, we arrived at a set of three necessary principles that are required to achieve our goal: invisibility [82], seamless manual override [45], and feedback [45][59][77]. Without any of these principles, a system will ultimately frustrate its intended users. A fourth principle, adaptability, is often desirable, especially in those situations where the behaviour of the system may need to change over time, or in response to different users. The importance of these principles cannot be overstated, especially with regard to their role in safety-critical systems. Inattention to manual override and feedback

capabilities has played a major role in many accidents, as highlighted by Leveson's discussion of automated systems [45] and Neumann's summary of computer-related risks [58].

Constructing useful systems that embody these design principles raises several issues. We now attempt to explore these, highlighting the rationale behind each and discussing our solutions.

3.2.1 Invisibility

> *The goal is to achieve the most effective kind of technology, that which is essentially invisible to the user.*
>
> MARK WEISER [82]

First, how does one make technology phenomenologically *invisible*[1]? In order for a system to function invisibly, users must not perceive themselves to be involved in a two-party communication. Ideally, the user should not even be aware that interaction is taking place. In our use of the conference room, quite the opposite was true. The most obvious problem was the distraction of the technology. Every time a presenter wished to display a document, play a video tape, or sketch on the whiteboard, interaction with a user interface, and often, a number of remote controls, was required to configure the equipment appropriately and provide the appropriate view to all attendees, both physical and electronic. Even with the so-called "friendliest" of graphical user interfaces (GUIs), the need to interact with a computer was an unnatural aspect of a presentation, and hence, acted as an obstacle to intuitive use of the presentation technology. Not only was the interface distracting to the presenter, it was often time-consuming to operate, thereby leading to audience distraction as well. This problem was further exacerbated by the difficulty of producing non-standard configurations, and by breakdowns of the technology that were poorly reflected by the interface, or worse, not indicated at all.

---

1. Interestingly, Norman argues that a good design makes the functions *visible*, either through good mappings and natural relationships between controls and their outcomes, or through the physical location and mode of operation of the controls, which serve as reminders of their function [59]. It should not be inferred from our use of the term *invisible*, that we disagree. On the contrary, we believe that the computer interface to these functions should be hidden, and that by doing so, we are forced to consider how the functions themselves can be made more visible.

Correcting these problems entailed two goals. First, we needed to reduce the overall complexity of operating the technology, addressing, in particular, how much explicit knowledge is required by the user to function effectively within the conference room environment. A few basic rules of interaction with any device must be made explicit to the user, but these rules should seem natural and not require a detailed understanding of the technology. Second, we had to reduce the intrusion on meetings of managing the operational aspects of the room. If a presenter wants to show a videotape, for example, the user should be concerned only with loading the tape and starting it, not routing the VCR output to appropriate displays. These goals implied that the old paradigm of the graphical user interface, which requires additional interaction and user control of low-level tasks, is no longer appropriate.

One way to achieve both of these goals was simply to have the room "driven" by a skilled operator with a computerized room-control system. While this was the norm in most high-end conference rooms, it was not an acceptable solution for the users of our conference room, many of whom had no vested interest in the underlying technology.[2] Instead, the room had to be "walk up and use."

Our solution was a variation of the "skilled operator" theme, in which the technology itself, rather than a human, manages the low-level operation of the room. An alternative formulation would be that the room had a skilled operator, but the nature of the requisite skills were the same as those required to function in a conventional conference room equipped with a VCR, overhead projector, whiteboard, and physically present audience. The underlying assumption was that if a human operator is able to infer users' intentions based on their actions, so should an appropriately designed system.

This approach lies in the premises of UbiComp and Augmented Reality [84], that is, the removal of the computer from our scope of awareness and the distribution of computational power throughout the environment by embedding it within individual devices. In other words, we are moving away from the centralized control of the GUI and toward a distributed control strategy of multiple, communicating entities. However, the strength of UbiComp is derived not from the abilities of each device in isolation, but rather, from the interaction and communication of these devices together [80]. This

2. Moreover, this was not very interesting as a research concept.

interaction leads to an emergence of rich behavioural properties, in much the same way that communication between simple goal-satisfaction systems leads to powerful, seemingly intelligent robots, as demonstrated by Brooks [7], Connell [16], and Cooperstock and Milios [18].

To provide a mechanism for such behaviour, the integration of sensors with various devices was required. The output of these sensors allows the computer to determine when certain actions should be taken by the environment, or in other words, how the environment should react to user-initiated events. Borrowing from the robotics literature, we call this resulting system a *Reactive Environment*. The approach is similar to reactive robotics [7][16] except that in our case, the sensor input influences responses of the environment rather than an autonomous vehicle or agent. Recognizing that this approach requires sensory input from a potentially large number of sources, and that the input could easily overwhelm a monolithic, centralized process, we opted for a distributed collection of specialized background processes or *daemons*, each responsible for a specific device or related set of tasks, and each receiving only those sensor readings appropriate for its use.

At the very least, a distributed collection of computationally capable devices enables some measure of local decision-making, thereby reducing the load on a centralized user interface. Ideally, however, we can do much better. By relegating the coordination of multiple activities to the various daemons in a decentralized fashion, we can obviate the explicit two-party communication that typically takes place between user and computer, hence, hiding the user interface.

Interestingly, strong motivation for this approach comes from the study of biological motor control.[3] In his summary of the *degrees of freedom* problem [5], Turvey notes several key points [76]:

- The details of any coordinated state are contributed gradually by many subsystems working together.
- Subsystems are relatively autonomous, possibly built to minimize interaction with the external medium.

3. The author would like to thank Kim Vicente of the Department of Mechanical and Industrial Engineering, University of Toronto, for pointing out the literature in this area.

- Subsystems cooperate to generate desired states at the system level without knowing that they are doing so.
- Executive control activates a particular subsystem to achieve some end without knowing the actual outputs of that subsystem nor even if that subsystem will be the actual one that performs the job.

These points lead to the conclusion that executive, or high-level, knowledge is only approximate. However, the advantage this offers biological organisms is that the behaviour of very many degrees of freedom can be coordinated through deliberate control of only a small number of degrees of freedom. Furthermore, Turvey reasons that the generality exhibited by a biological movement system is a direct result of the individual specializations of each sub-component [76]. Applying this argument to the control and coordination of many presentation devices in the conference room, we reason that even at this relatively small scale, contrasted with the demands of biological movement (cf. Thelen's work on motor development [73]), distributed control is not only desirable, but perhaps imperative.

It should be noted that the metaphor of hiding the explicit interface is not universally applicable. There are many high-level goals that simply cannot be inferred from context alone, and hence, many tasks that a Reactive Environment cannot perform without explicit interaction with the user. In such circumstances, be they in a videoconference setting or elsewhere, the logical approach is for the required human input to be motivated in a clear, yet unobtrusive manner, so that the user is not unnecessarily distracted by the technology. This design goal, however obvious, seems to have been ignored by a vast majority of hardware and software computer manufacturers.

3.2.2 Manual override

> *The question then becomes why we should not just eliminate humans from systems and replace them with computers.*
>
> NANCY LEVESON [45]

Relegating control of technology to a collection of background processes runs the risk of preventing the user from superceding the default automatic behaviour of the system when appropriate or desired [45]. For instance, the fact that 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, or in other words, make use of the "master off switch." The need for such capabilities should be readily apparent from a usability perspective, but projects as ambitious as the European Airbus neglected these issues with dangerous consequences. The fly-by-wire controls of the Airbus were designed to limit human inputs when these are outside of the aerodynamic specifications of the aircraft. While such behaviour is appropriate under a wide range of circumstances, the simplistic application of this design approach is widely believed to have been at least partially responsible for several accidents involving the Airbus A320 [58]. As Leveson notes in her discussion of automated devices, not all conditions are foreseeable, and hence, humans are required to intervene in the operation of these systems [45].

What is perhaps less obvious is that for a manual override mechanism to be effective, its invocation must be possible with minimal cognitive effort. While this is especially true for safety-sensitive situations, such as flight control of a jet airplane, it is also important for less time-critical applications, such as the control of our conference room equipment. If use of the manual override is as complicated as the original GUI, users are unlikely to become familiar with its operation.

The question then becomes how one provides users with a simple and seamless *manual override* mechanism, to deal with those occasions where the default behaviour 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 a thick instruction manual. Similarly, users must be able to override the Reactive Room quickly and easily.

In addition, there should not be a need to "argue" with a system if it is not behaving as one desires. In the case of a dispute, the computational power of the device should seem to disappear. However, some allowance might be made while the system is learning the behaviour of a new user. To minimize the possibility of disputes, reactions to user-initiated events should be conservative.

The design challenge was to provide an intuitively obvious manual override, something that users could grasp quickly with little explanation. While the research presented here tackles this problem only in a limited scope, it raises several questions for larger projects.

Is complexity of coordination of multiple devices the limiting factor or should we be more concerned with the complexity of each device, individually? Another problem is the construction of a standard set of manual override mechanisms for multiple devices and multiple classes of devices. If the functionality of two similar devices is not identical, should manual override expose the differences by allowing an operation on one device but not the other, or should the control system attempt to hide these differences?

3.2.3 Feedback

> *The operator can intervene effectively only if the system has been designed to allow the operator to build a complete and accurate mental model of its operational status, including providing the information necessary for the operator to understand the system state and providing it in a form that is understandable under stressful conditions.*
>
> NANCY LEVESON [45]

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, in other words, diagnostic feedback. Although such information is often insufficient to allow a casual user to correct the problem, it at least reassures them that they are not the cause. This is in stark contrast to many computer systems today that promote a sense of ignorance and incompetence in all those who are not technophilic gurus.[4]

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 communicating relevant

4. As evidence of this phenomenon, consider the preponderance of books such as *DOS for Dummies*.

state information without a graphical user interface. For example, in a videoconference, how do I know that the video tape I am playing is visible to the remote attendee?

Regardless of whether the system is operating correctly or incorrectly, we consider feedback to be of great importance. This is underscored by the role of insufficient feedback in several major accidents, including the Therac-25 radiation therapy overdoses [45] and the Exxon Valdez oil spill [58], and an overdose of feedback in the Three Mile Island nuclear power plant accident [47]. In each of these incidents, more effective system feedback could have prevented the tragedy. For example, one of the contributing factors to the Exxon Valdez oil spill was the fact that neither the third mate nor the helmsman realized that their rudder controls were being ignored because the ship was on autopilot [58]. Drawing from these lessons, we believe that diagnostic feedback should be available on-demand to indicate the source of a problem during failures and a tolerable level of status feedback should provide relevant state information at all times, while not unnecessarily interfering with other important activities [77].

Keeping in mind our initial goal of getting the user away from the computer, we were faced with the problem of providing this information without the benefit of a user interface. In an attempt to understand what is both appropriate and effective, we investigated several forms of feedback for this purpose, including persistent visual cues, background audio, and even speech.

3.2.4 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 behaviour. Rather than being treated identically, users may require different default configurations, reactions to user-initiated events, levels of manual override, and forms of feedback.

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, in much the same way that humans demonstrate flexibility in adapting to new demands? If two people have different expectations as to how a system should behave, then ideally, the system will respond differently to them, assuming it can distinguish when each is responsible for

control. Again, we wish to avoid explicit interaction with the computer in order to train it, but how can this be accomplished while minimizing learning time?

Adaptability should be a result of experience with the user. Like Maes, we believe that a gradual learning process will be beneficial in terms of the user developing trust in the system [49]. Furthermore, the learning-by-experience approach offers the additional benefit of not requiring explicit tailoring or programming by the end-user. In order to provide such capabilities, a system must have a persistent state [41], in other words, memory of previous events. However, this leads to several difficult questions concerning what constitutes an event that warrants learning, when learning should take place, and how the user should be informed of such learning.

3.3 Summary

Inspired by the ideas of Ecological Interface Design, Ubiquitous Computing and reactive robotics, we proposed the concept of a Reactive Environment. Rather than requiring a skilled human to manage the operation of complex technology, the environment assumes this role, making context-sensitive reactions in response to the user's conscious actions. By supporting automatic configuration of equipment, a Reactive Environment frees the user to concentrate on the primary task, for example, giving a presentation, rather than the operational aspects of the room. To ensure usability of such an environment, the three necessary principles of invisibility, seamless manual override, and feedback must be observed. In many situations, the additional principle of adaptability is also desirable. We now turn to the task of applying these design principles to a real-world problem, that of automating control of a videoconference environment.

Chapter 4 The Reactive Room

I believe computers must be able to see and hear
what we do before they can prove truly helpful.
ALEX PENTLAND [62]

Our prototype implementation of a Reactive Environment, the Reactive Room, embodies the four design principles discussed in the previous chapter. The room is a fully functional videoconference environment, in which the presentation technology has been augmented with sensors, computers, and communications. Each device is monitored by one or more distributed software processes, or *daemons*, which collect information from sensory input. Through this background monitoring and some computation, daemons maintain awareness of activity relevant to each device and share information with each other when required. The result is a computer augmented environment that reacts to human activity in order to support and simplify the task of the presenter.

Before elaborating on the details of the Reactive Room, we first illustrate its operation by presenting a possible scenario involving our prototype videoconference environment.

> *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 video attendees 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 attendees 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 labelled as "electronic visitor" 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 again becomes visible to the remote visitors. Finally, when Nicole removes the diagram, the visitors receive a full view of her.

At this point, a new electronic attendee, 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 the above scenario, the Reactive Room satisfies our design principles of invisibility, manual override, and feedback. 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. The affordance of a direct manual override mechanism for both the room lights and the presentation devices allows users to override default behaviour seamlessly without the confusion of mapping user interface representations to the corresponding devices. Furthermore, the use of audio and visual feedback provides confirmation that various operations have succeeded. Finally, additional support for remote participants improves their sense of engagement in the meeting and allows them to adjust their views without interrupting the presenter.

In the next four sections, we examine how the Reactive Room satisfies each of our design principles. Even for the simple problem of automating the control of room lights, we have seen that achieving our goals is non-trivial, and that the problems involved are often not apparent until the systems are put to actual use. As a result, we begin the discussion of

each principle by describing how it relates to this deceptively simple task, and then proceed to the more interesting problem of the augmented videoconference environment. While it is beyond the scope of this chapter to provide a detailed task analysis of videoconference operations, the interested reader is referred to Appendix A for an anatomy of the control logistics. The final section of this chapter discusses the challenge of re-engineering the Desk Area Network in order to satisfy the requirements of our design principles.

4.1 Invisibility

4.1.1 Room Lights

Motion, 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. This same claim can be made for a wide variety of devices in everyday use, including refrigerator lights, automatic doors at the supermarket, and home heating and air conditioning systems.[1] We learn to take these systems for granted, precisely because their invisibility allows us to ignore them.

4.1.2 Videoconference Environment

In order to achieve a similar level of invisibility in the Reactive Room, a number of daemons perform background monitoring functions of the various presentation devices. 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 timeout 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

1. Note that these last two systems are generally activated by environmental factors, rather than actual human activity, but given their prevalence in our daily lives, their inclusion here is warranted.

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. Further details regarding these functions are provided in Appendix B.

4.2 Manual Override

4.2.1 Room Lights

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 latch 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[2], with conventional motion detectors. This system operates in four states or modes, as illustrated by the state diagram of Figure 12. 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 re-activated 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. Further implementation details can be found in Appendix C.

---

2. 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.

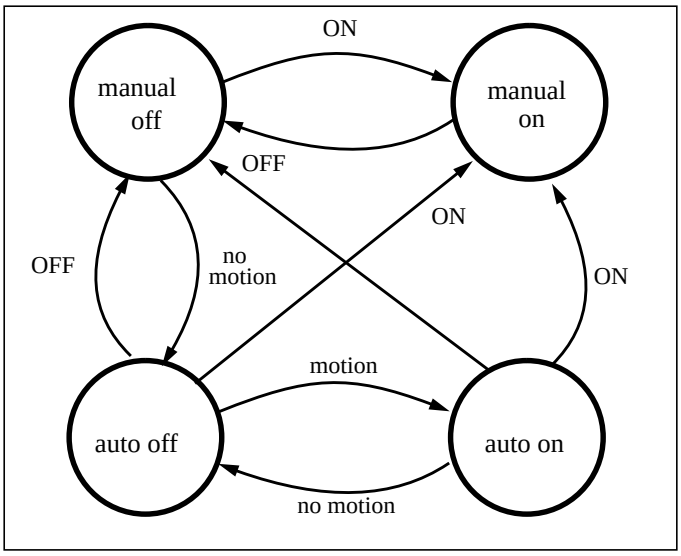

FIGURE 12 Smart Lights state diagram. The transition labels ON and OFF denote the pressing of the ON or OFF buttons, respectively.

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 turned on when the room was entered, and turned off after the room was vacated, automatically. If the occupants desired the lights to be turned off, or back on again, they could do so 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. This exercise proved itself to be a useful demonstration of the seamless manual override.

4.2.2 Videoconference Environment

In keeping with the same design rationale, users must be able to override the default behaviour of the Reactive Room by establishing connections between various devices without directly handling the computer. Resorting to the GUI patchbay of Figure 7 (Section 2.4) for manual override is simply not acceptable. Presenters who were unable or unwilling to deal with the complexities of the user interface in the past are not going to be satisfied with a system that requires an inordinate amount of effort to correct when its

behaviour is undesirable.[3] Comments from users of our conference room and users of the University of Toronto's electronic classrooms provide ample support for this claim.

Buttons and Lights

To permit this functionality, we provided a set of button-and-light modules, consisting of a single push button and an LED, attached to each device[4] 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[5] 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 individual with the output of our document camera. If this behaviour were not the default 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.

---

3. Designers of the University of Toronto's electronic classrooms likely recognized this when they provided manual light switches to override the default computerized control of the room lights.
4. For simplicity, each electronic seat, composed 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.
5. A mirror connection for an electronic seat is established by routing the camera output directly to the monitor.

Laser Pointer

The button-and-light modules were originally designed as a prototype tool to provide basic manual override capabilities to the Reactive Room. However, the need for the presenter to walk around the room in order to establish non-default connections between the various devices was too awkward for general use. Our solution was to use a calibrated laser detector to provide the required functionality. As shown in Figure 13, users can simply point to a source and destination device with a laser pointer to establish a connection between the two devices, or point to a command icon to select any other function the designer has provided. 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.

FIGURE 13 The laser pointer in use. The speaker is pointing the laser at one of the electronic seats to provide this view to a remote visitor.

The system operates as follows: A wide angle video camera with a band-pass filter, tuned to the frequency of our laser pointer, is placed at the rear of the room. The video image is digitized several times per second, and scanned for a cluster of bright pixels, corresponding to the point of a laser beam. If such a cluster is found, its location is checked against a map of the room, containing the bounding boxes of each selectable device. Such a map, generated by pointing the laser at the top left and lower right corners of each device, is shown in Table 1. Provided that the camera remains stationary after this map is produced, and is placed sufficiently high so that objects within its view are not

obscured by people in the room, we have found the system to be highly robust. A further advantage of this technique is the simplicity of reprogramming after the addition of a new device or a change to the room's physical configuration.

	Coordinates			
Device	**x1**	**y1**	**x2**	**y2**
seat1 camera	50	40	70	53
seat1 monitor	56	53	70	64
vcr	53	66	75	69
document camera	103	14	155	105
macintosh	99	64	130	93
whiteboard	42	33	49	40
visitor1	91	36	96	39
visitor2	92	43	97	47

TABLE 1 Room map for device selection by laser pointer. The coordinates are relative to the video image obtained by the laser detector camera. Any device can be selected as a source or destination of a connection by activating the laser beam inside of the bounding box defined by the two diagonal corners, (x1,y1) and (x2,y2).

Because recognition of the target is performed optically, it is trivial to extend this process to more interesting operations. For example, we have pasted an icon of the trashcan on the wall, and by virtue of detecting a laser beam in its location, we can now realize the metaphor of "dragging objects to the trash" to clear connections. However, even more powerful capabilities exist. In particular, we could place an icon above a large monitor at the front of the conference room, and point to it in order to move an electronic visitor to the presenter's position. Other icons can be added to perform control and diagnostic functions, such as permitting World Wide Web access, or querying the document camera.

A similar approach was taken in the MIT Intelligent Room project [74], except that instead of seeking a laser point, their system tracks motion of the presenter's extended forefinger, and responds whenever the finger is held static within the bounding box of a defined command icon. While this frees the presenter from using a laser pointer, its obvious disadvantage is the inability to control the room from an arbitrary position. With

increased processing power and additional cameras placed around the room, it should be possible to select a command simply by pointing to its icon. However, a speech-based interface for such functions, similar to that described for electronic visitors in Section 5.1, is probably more intuitive, as well as more robust to noise.

Administration Driver

The button-and-light modules and the laser pointer serve as effective manual override tools for most users. However, as room administrators, we often require a higher level of control over the devices and daemons within the Reactive Room. Examples of this include the need to calibrate the laser detector, as described previously, or to program the VCR remotely. At the earliest stages of our prototype effort, it was clear that such capabilities were required, and so, to provide these with a minimum of programming effort, we implemented a text-based driver.

The driver produces a list of all available daemons by consulting a file-based name-server and then allows the user to communicate with any of these, issuing commands and viewing replies. When a particular daemon is selected, the driver first asks the daemon to provide a menu of possible commands and options for reference. As the user enters commands, they are sent to the selected daemon, and the generated replies are immediately displayed. In addition to unique commands appropriate to their specific functions, each daemon recognizes and responds to the four generic commands listed in Table 2.

command	description
ping	checks communication with the daemon
menu	obtains list of commands the daemon recognizes
query	obtains state information from the daemon
quit	causes the daemon to unregister and terminate operation

TABLE 2 List of generic commands recognized by all daemons.

The driver can also be invoked with the "`-k`" (kill) flag, causing a quit command to be sent to all of the daemons, thus providing a quick shutdown of all Reactive Room processes.

4.3 Feedback

4.3.1 Room Lights

Although the behaviour of our motion-sensitive light controller is quite predictable, we still found it useful to provide state information, especially for diagnostics purposes. This was accomplished by adding an LED panel, shown in Figure 14, that indicates which of the four states the switch is in, and displays the output of the motion detector. As described below in Table 3, 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.

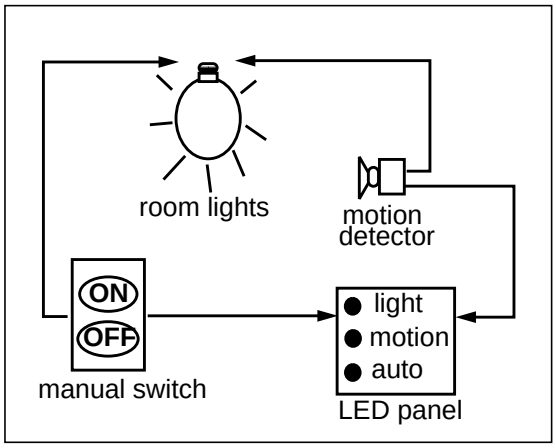

FIGURE 14 The Smart Light Switch consists of a motion detector (invisibility), a manual switch (manual override) and an LED panel (feedback).

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

TABLE 3 Probable causes of lights not turning on when a user enters the room.

4.3.2 Videoconference Environment

The need for similar feedback throughout the videoconference environment was addressed in part by 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 obtained simply by offering presenters a video monitor that reflects the view being provided to remote participants.

Buttons and Lights

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 that 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.[6] At this point, a double beep sounds, and both module LEDs turn on, indicating that the desired connection has been established. The LEDs remain illuminated until a timeout period expires. If, however, the connection fails, the LEDs are immediately extinguished. The same feedback is provided when the laser pointer, instead of physical buttons, is used to select devices. The module operations are summarized in the state diagram of Figure 15.

The importance of audio and visual feedback for these operations cannot be overstated. Users are often unsure as to whether they pressed the button with sufficient force, or if the system recognized the button press. 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.

6. 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.

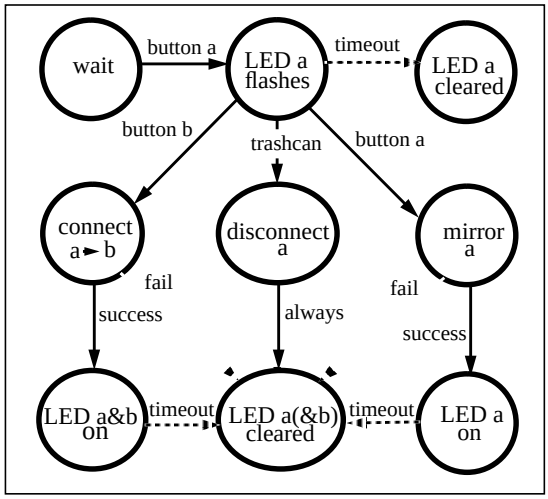

FIGURE 15 State diagram of the button-and-light modules. The first row of states corresponds to the processing associated with the first button press, while the second row represents the actions taken in attempting to form or drop connections in response to the second button press. The final row depicts 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.

PARC Tabs

While the feedback offered by the button-and-light modules is appropriate for the role of these artifacts, the operations they support are fairly limited. The administration driver provides much more control, but the textual information it returns is clearly inappropriate for background feedback, or for casual users when confronted by problems with the equipment.

As an attempt to remedy this situation, we experimented with the Xerox PARCTab [80] as a simple and portable user interface for feedback and diagnostic purposes. The PARCTab or "Tab," pictured in Figure 16, is a palmtop mobile computer that communicates with a base station using infrared signals and offers a touch-sensitive screen capable of displaying text and bitmap graphics, as well as a piezo-electric speaker for generating audio tones.[7]

The first serious application we developed for the Tab was a control and diagnostic system based on a floor plan of the room, shown in Figure 16. The control component allowed

users to establish or break connections between devices, move visitors between electronic seats, and obtain audio and visual feedback indicating the success or failure of these operations, similar to the functionality of the button-and-light modules and the laser pointer. However, the more interesting aspect was the diagnostic component that provides the ability to query the status of connections to or from any device, simply by touching the associated icons on the screen.

FIGURE 16 The room-control floor plan application running on the Xerox PARCTab.

Future extensions of the Tab were planned to allow presenters additional control over, and feedback from devices, for example, starting the VCR, and monitoring the tape counter. However, as user testing proceeded, we realized that we had run into the classic “strength vs. generality” dilemma. The more functionality we attempted to provide, the more complicated the interface became, and hence, the amount of training required became a problem. From our perspective, the PARCTab was functioning as a portable computer screen, not an invisible technology. This contradiction of our first design principle led us to seek alternative, non-computer interfaces for these purposes.

7. Since these devices also serve as active badges [79], we began using them as a means of indicating that the presenter had entered the room, thus causing the equipment to be activated. Unfortunately, despite their small size, we cannot realistically expect all of the users of our conference room to carry a Tab merely for the sake of activating equipment. As a result, we later discarded these units in favour of the calendar system, discussed in Appendix B. Another possibility would be to make use of the pass-card information from magnetic door keys to identify the occupants in the room. However, this method could easily fail to identify the presenter if anyone else uses their card to open the door.

4.4 Adaptability

4.4.1 Room Lights

Our first problem with the initial design of the Smart Light Switch was the logic concerning the *manual on* mechanism. Once the lights were switched on, manually, they would remain active until the user switched them off. This led to our lights being left on for several days, since the system would never enter an automatic mode. The alternative of returning to an automatic mode when motion ceased was also unacceptable, since a faulty motion detector would then prevent the user from activating the lights. It was obvious that an additional level of control was needed.

The Smart Light Switch satisfied three of our design principles: invisibility, manual override, and feedback. The system was invisible, since it performed its function in the background, without requiring interaction with the user. Seamless manual override was provided by a manual on/off switch, which utilizes existing skills people acquire in the everyday world. Finally, the LED panel indicated the state of the system. However, our fourth design principle, adaptability, had not been addressed.

As our requirements for the switch changed, there was no way to modify its behaviour without reprogramming the computer chip at its core. Rather than requiring such hardware adjustments, we took the approach of augmenting the circuit with two computer inputs, so that new behaviours could be programmed in software. We now had to consider three state variables (light, manual override, and computer override) and five inputs (motion, manual on, manual off, computer on, and computer off). As a consequence of the increased state space, the challenge of maintaining this device grew enormously. Manual reprogramming of the software, while far easier than hardware modifications, was still unfeasible. Our solution was to incorporate a learning mechanism that would modify the state table of the controller in response to manual override operations. This way, learning the preferences of different users could be carried out by a computer.[8] Furthermore, because the learning software can communicate with other processes, the behaviour of the light switch could take into account additional factors such as time of day and active devices.

---

8. Although this proposal precedes the discussion of adaptability for the full videoconference environment, we have not completed the implementation of the adaptive light controller. This remains a topic for further research, as described in Section 6.2.

4.4.2 Videoconference Environment

To provide an adaptive mechanism to the room, we drew on the ideas of programming by example [46][55]. 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 behaviour so that this action will be taken under similar circumstances in the future. As discussed in Section 4.5.4, learning of these rules is on an individual basis, so that different users can tailor the behaviour of the room as appropriate to their requirements.

The introduction of an adaptive component to the system poses additional challenges. In order to provide software control capability, the computer inputs must be able to override the manual modes of operation. However, we must balance the requirement for conservative computer behaviour (e.g. not re-applying an action immediately after the user had manually negated it) with the need for rapid learning of user preferences. This could be achieved by prompting the user with a suggestion and series of options, as follows, whenever the Reactive Room encounters a new situation for the first time, or notices that an automatic reaction was in conflict with a manual override:

- **suggestion:** the last manual override should be learned as a general rule
- the last manual override was a special case that should not be learned
- the computer control should be disabled until further notice
- the last (computer) automatic reaction should not be repeated in the future

If the user chooses to ignore the suggestion, the room might then proceed to modify its rule set conservatively, or simply continue its default behaviour. While this method incurs the cost of explicit interaction, it allows the user to guide the learning process more efficiently, and is thus far more likely to result in acceptably short learning times than would an unsupervised scheme. It is also worthwhile to note that our unsupervised learning mechanism could prove highly frustrating if it learned in response to manual overrides that were only made as a special case. This pitfall would be avoided through the interaction as described above.

An alternative approach would be to treat the learning of user preferences as a problem of designing a programming metaphor based on natural language or similar high-level

specification method.[9] Provided such a metaphor could be found, this offers the possibility of direct user specification of preferences, without the cognitive effort of translating human intentions and desires to the machine syntax [77]. This constitutes an interesting research problem that warrants further examination.

4.5 Implementation Considerations for the Desk Area Network

This section describes the series of changes to the Desk Area Network, introduced in Chapter 2, that were required through the implementation of the Reactive Room. The driving force behind this evolution was the need to provide a system that could easily be deployed in new locations with a minimum of administrative effort, yet was sufficiently flexible to satisfy our design principles.

The Desk Area Network (DAN) daemon sits at the heart of the Reactive Room, automatically selecting appropriate configurations of audio and video switch settings in response to the activation and de-activation of input sources, the joining and leaving of a meeting by electronic attendees, the movement of these attendees to different electronic seats in the room, and the manual override of connections by users. These events can occur in arbitrary order, and the DAN must respond in an appropriate manner in order for its operation to seem invisible. Finally, provisions for adaptability to new users and changing user requirements was necessary. Attaining this level of sophistication was by no means a trivial task.

4.5.1 Impractical Implementations

We considered, and subsequently rejected, two extensions to the initial DAN system that were conceptually quite simple, but unrealizable in practice. The first of these was to construct an exhaustive preset list, covering every possible combination of device states, electronic visitor positions, and user tasks. However, it was readily apparent that the effort to maintain a database for such an implementation would prove somewhat unwieldy. Even for the conservative requirements of three devices (VCR, document camera, and digital whiteboard), three electronic seats for two remote attendees, either of which may or may

9. The author would like to thank Toshiyuki Masui of Sony CSL for this interesting suggestion.

not be present, and two user tasks (local presentations and meeting capture via videotape), we were faced with the prospect of $3^2 \cdot 7 \cdot 2 = 126$ different presets.

Our second idea was to extend the concept of the undo function, described in Section 2.5, to operate on a stack of multiple selections, rather than on the last selection, alone. When a device became active, its preset would be pushed on the stack, so that its side effects could later be undone when the device was no longer in use. While the destructive nature of disconnects could be avoided by a careful design of the presets, we realized that applying and undoing these operations in response to an arbitrary ordering of events would not always result in the appropriate configurations.

4.5.2 Device Matrix Table

The next implementation of the DAN was based on the fact that the necessary connections resulting from an event were, in general, from the device responsible for the event, as a source, to any number of other devices, as destinations. For example, activation of the document camera causes connections to be made from the document camera to local monitors, remote attendees, and, if the meeting is being recorded, to a VCR. This led to the development of a simple connectivity matrix, shown in Table 4, that sufficed for most purposes.

source	destination						
	vis1	**vis2**	**seat1**	**seat2**	**seat3**	**vcr**	**wboard**
vcr	√	√			√		
macintosh	√	√				√	√
document	√	√			√	√	
vis1			√			√	
vis2				√		√	

TABLE 4 Connectivity matrix for the DAN. The checkmarks along each row indicate the inter-device connections to be made when the associated source becomes active or the associated visitor joins the meeting.

There were, however, a number of exceptions that required special treatment. When a remote attendee joins a meeting, the DAN must make connections not only from the

attendee as a source, but also to the attendee as a destination. Similarly, when the record button on the VCR is pressed to record a meeting, connections must be made to the VCR. In both of these situations, determining the appropriate input sources can be handled by keeping track of which devices are in use, and the order in which they were activated, under the assumption that a view of the most recently activated device should be provided.

Unfortunately, this was still insufficient for our purposes. The biggest complicating factor was allocation of the picture-in-picture (PiP) resource. When a presentation device is in use, the PiP can simultaneously provide remote visitors with a view of the meeting from a video surrogate seat and the output of the active device. But when more than one electronic visitor is attending, which video surrogate should provide the PiP input? Should both visitors receive the same view, or only the visitor who appears on the surrogate monitor? Similar questions arise when dealing with meeting capture via videotape. Should the VCR dominate the PiP or should priority be given to the visitors? Since the answers to these questions are likely to depend on personal preferences, the implementation could not incorporate any hard-coded rules. It soon became clear that the connectivity matrix was insufficient to handle these special cases, and we were forced to abandon this approach.

4.5.3 Device Descriptors

The solution we finally adopted grew directly from the realization that some events, such as the arrival of a new electronic attendee, result in more complicated behaviour than could be represented by a simple matrix-based rule set. Furthermore, our constantly evolving conference room required a flexible configuration of devices that could be easily modified by an administrator without generating new code. To permit this, we created a set of descriptors that provide a rich representation of each device, along with the user's configuration preferences. The available descriptors include:

- *name:* the text label by which the device is referred
- *type:* either H (hardware device), S (electronic seat), or V (visitor)
- *av-plugs:* the audio-visual switch plugs for input and output
- *button:* the number of the button-and-light module associated with the device
- *connect-to:* destinations to which this device connects when activated

- *seat-pref:* (for visitors) to what seat(s) this visitor should attempt to connect
- *default-in:* device that provides input if no other sources are active
- *input-pref:* any combination of H (hardware), S (seat), and V (visitor), specifying which source(s) should be displayed if more than one is active[10]
- *use-pip:* when more than one source is active, whether a PiP should be used to display them

By way of illustration, Figure 17 provides the device descriptors of our VCR and a sample visitor. The VCR *connect-to* descriptor specifies that when a tape is played, its output will be made available to both visitor1 and visitor2, as well as displayed locally on the projection screen. However, if the VCR is used to record the meeting, the input source or sources selected are context-sensitive, as determined by the *default-in*, *input-pref*, and *use-pip* descriptors. This is explained by the if-then-else rules of Figure 18, which are invoked when the VCR record button is pressed.

The visitor2 *seat-pref* descriptor states that this attendee will ordinarily appear at seat3, unless that seat is currently occupied by another visitor. In that case, the attendee will appear at seat4, instead. The *use-pip* and *input-pref* descriptors specify that the attendee will normally receive a view from the previously determined seat. However, if a hardware device, such as the VCR, is active and the PiP is available, then the visitor will receive a PiP view of both the device output and the seat view.

4.5.4 Learning for Different Users

The device descriptors format lends itself to an elegant method of user-tailoring, or adapting to different users. When a user establishes a non-default connection or destroys a default connection through the use of the button-and-light modules, 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 behaviour so that this action will be

10. In the case of a seat preference, the selected seat is taken to be the first seat found which serves as a video surrogate for a remote attendee. This is necessary because electronic seats are always assumed to be active.

name:	vcr
type:	H
av-plugs:	SV 9 DV 9 SA 11 12 DA 11 12
button:	7
connect-to:	visitor1, visitor2, projector
seat-pref:	-
default-in:	seat1
input-pref:	SVH
use-pip:	true

name:	visitor2
type:	V
av-plugs:	SV 2 DV 2 SA 2 DA 2
button:	2
connect-to:	vcr
seat-pref:	seat3, seat4
default-in:	-
input-pref:	HS
use-pip:	true

FIGURE 17 Device descriptors for the VCR and an electronic visitor. The av-plug labels SV, DV, SA, DA, correspond respectively to the plug numbers associated with each device's source video, destination video, source audio, and destination audio.

taken under similar circumstances in the future. At present, the changed behaviour is reflected by the insertion or deletion of device labels in the *connect-to* descriptors for each device. In the future, provisions for changing the *input-pref* and *use-pip* descriptors should also be incorporated, ideally through a similar mechanism.

The calendar daemon, described briefly in Appendix B, notifies the DAN at the start and termination of each user's room booking, so that new rules can be saved in the preferences file corresponding to the appropriate user. In this manner, different users can tailor the room's behaviour to suit their individual requirements, without ever interacting with the computer.

4.6 Summary

In the Reactive Room, context information from a variety of sensors is provided to a number of task-specialized daemons that interact with one another. This communication between distributed processes allows the emergence of powerful behaviours. To capitalize on the available potential, we integrated and extensively modified the previously

```
if one or more visitors are connected
     source_2 <- first visitor in connected list
     source_1 <- camera view provided to that visitor
else
     source_2 <- NIL
     source_1 <- seat1 (default in)
     if a hardware device is connected
          source_2 <- most recently activated device

if a PiP is available and source_2 <> NIL
     record source_1 in large PiP window
     and source_2 in small PiP window
else
     record from source_1
```

FIGURE 18 Algorithm for selection of VCR inputs during record operation. Note that a visitor is considered **connected** when the node associated with that visitor is in use, whereas a hardware device is **connected** only if it is currently active and the VCR appears in its *connect-to* list. A PiP is **available** when it is either not in use, or in use by a device with lower priority than the VCR.

described room-control interface so that it would flexibly accommodate new rules and adaptive behaviour.

The Reactive Room provides an interesting case study of our design principles and demonstrates that control of the devices in our videoconference environment can be greatly simplified. With the user interface effectively hidden, physically present users can now exploit the functionality of technology without incurring the overhead of complexity. In addition, our reactive environment incorporates several mechanisms to provide the user with seamless manual override and feedback capabilities.

The next challenge was to build upon the infrastructure of the Reactive Room to improve the sense of engagement of remote videoconference attendees. This research, described in the following chapter, entailed providing telepresent users with both additional control

over the technology as well as increased spatial and sensory awareness of our environment.

Chapter 5

Augmenting a Media Space for Remote Users

Any medium powerful enough to extend man's reach is powerful enough to topple his world. To get the medium's magic to work for one's aims rather than against them is to attain literacy.

ALAN KAY [40]

Traditional videoconferencing enables people to meet without being physically present at the same location. However, these systems tend towards passive, non-interactive communication [42], in which the sense of physical distance is often reinforced, rather than reduced, by the technology. Unlike the more limited videoconference environment, a media space is not associated with special events such as meetings [30]. Rather, a media space is continuously available to all of its members. This has several important implications to us, largely because we work in this environment. Our primary concern is to make the media space a seamless extension of our physical space. To accomplish this,

---

Section 5.2 of this chapter was originally published in the IEEE Pacific Rim Conference on Communications, Computers, Visualization and Signal Processing, May 1995. Section 5.1.2 and Section 5.3 were originally published in the Proceedings of Human Factors in Computer Systems (CHI), April 1996. Permission to reproduce the material here was granted by IEEE and ACM.

the mechanisms for interacting with the media space must be made as natural as possible, both for physically present and electronic or telepresent users.

The previous chapter demonstrates the benefit of a Reactive Environment to physically present users. We now explore how a similar application of our design principles simplifies the use of technology and empowers telepresent attendees with more effective interaction with our media space. First, we outline the limitations of conventional media spaces from the perspective of the remote user, then proceed to some implementation details, and conclude by discussing how our design principles have been applied to ameliorating this situation.

5.1 Limitations of Media Space Communication

5.1.1 Navigation

Imagine placing a telephone call and not connecting with your party until they dialed your number as well. This absurd situation is actually the status quo when a visitor from another site tries to contact a media space, without compatible hardware and software. While conventional systems provide a high degree of connectivity for local users, problems arise when a remote user attempts to contact an individual, using an audio-video modem (codec), without prearranging the call. In this situation, they receive, at best, a preset view of the environment and at worst, no view at all. In either case, they are stranded with no ability to navigate, for instance, to enter my office electronically. To avoid this pitfall of media space communication, visitors must prearrange their calls with a local attendee through, for example, email. Both the visitor and the local attendee must then connect to the modem at a specified time in order to videoconference.

Audio/Video Server Attendant[1]

Our solution, the Audio/Video Server Attendant (AVSA) overcomes this limitation by providing visitors with the ability to navigate independently through a media space. The

1. The Audio/Video Server Attendant was a project begun by Koichiro Tanikoshi with the assistance of several undergraduate students, notably, Shahir Daya and Radek Nowicki. Since then, Anuj Gujar completed the prototype implementation, and along with Akil Nasser, ported the code to a UNIX environment. The material describing this system is included here as an illustration of the ideas of this thesis.

AVSA instructs visitors on available commands through a video-overlay menu [33], as pictured in Figure 19, and allows visitors to issue their selections verbally. Instead of requiring the assistance of a local party to make the connections, the attendant allows them to see if individuals are present and if so, to contact them directly.

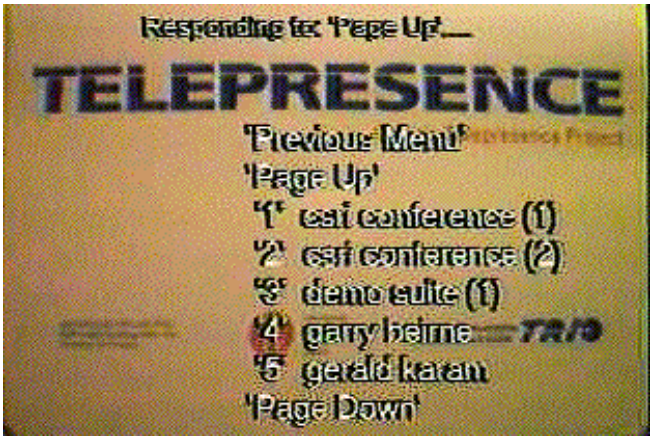

FIGURE 19 The initial Audio/Video Server Attendant menu offers a selection of people with whom the user can visit. Selections are made by uttering the desired option enclosed in quotes.

5.1.2 Camera-Monitor Mediated Vision

...active exploration is crucial for interaction.

J. J. GIBSON [31]

For the remote attendee, navigation between rooms is not the only problem. While physically present users enjoy the freedom to look around the room as they please, rapidly shifting their gaze in reaction to rich visual and non-visual stimulus, telepresent users are typically offered only sparse stimulus and are deprived of the ability to change views independently. In order to appreciate the implications of this problem, a brief comparison of human vision and camera-monitor mediated vision is in order.

Normal human vision can be conceived as consisting of two highly mobile cones of view. One is the focused foveal cone, one degree wide, while the second is the peripheral cone, or global field of view, spanning approximately 170 degrees [3]. Excellent spatial resolution is provided by the first, while the second, lower resolution view, provides us with stimulus that acts to redirect our attention.

Camera-monitor mediated vision, in contrast, suffers in resolution and due to the size of the display, uses limited azimuth of the visual field. Watching television, for instance, typically involves the foveal cone only. The narrow channel of information, both in the sense of bandwidth and field of view, imposes limitations on the ability to explore, follow conversations, check reactions, and generally sense significant actions in a remote space, such as people passing by or entering. In such situations, users must choose between a global and a focused view. With the former, resolution is sacrificed to permit a wide field of view and easy change of gaze direction. If only the focused view is provided, users obtain details but no peripheral awareness. This is typical of most videoconference settings [17][28].

MTV

One approach to overcoming this shortcoming is to support both the foveal and peripheral cones with multiple views. The problems with this approach are well understood. The Multiple Target Video (MTV) system of Gaver et. al. [29] first proposed the use of multiple cameras as a means of providing more flexible access to remote working environments. Users were offered sequential access to several different views of a remote space. However, as the authors noted, a static configuration of cameras will never be suitable for all tasks. Furthermore, switching between views introduces confusing spatial discontinuities. A further study (MTV II) by Heath et. al. [35] attempted to address this latter issue by providing several monitors, so that every camera view was simultaneously available. While this new configuration was more flexible, the inability of static cameras to provide complete access to a remote space still remained a problem. Furthermore, the various views were independent of one another, and the relationship between them was not made explicit. Consequently, spatial discontinuities persisted.

Virtual Window[2]

Another approach involved the Virtual Window concept [30], which uses the video image of a remote attendee’s head to navigate a motorized camera locally. The remote monitor is

2. The Virtual Window concept originated from the work of Bill Gaver, of the Royal College of Art, UK, along with Gerde Smets and Kees Overbeeke at the Technical University of Delft, the Netherlands. Our implementation of this system was based on a prototype developed by Bill Gaver, and refined by Koichiro Tanikoshi and Jeremy Cooperstock. The material describing this system is included here as an illustration of the ideas of this thesis.

treated as a window through which the local room can be viewed. When the attendee leans to the left or right, or moves up or down, the camera pans or tilts accordingly (see Figure 20). The zoom factor is similarly determined by the attendee's proximity to the

FIGURE 20 The head tracking camera control system in operation. The large images represent the view received by the video attendee, while the small inset images represent the appearance of the attendee in the conference room. The motorized camera appears at the top of the video monitor.

monitor. As the remote user moves closer to the monitor, the camera zooms to provide the sensation of moving towards people and objects in the room. The interested reader can find implementation details of the Virtual Window system in Section 5.2.

While our experience with this system [21] revealed a significant improvement to the user's 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 awareness often made the user unaware of important activity taking place out of view.

Extra Eyes[3]

To compensate for the limitations on vision imposed by camera-monitor mediated telepresence, we developed the Extra Eyes system, which offers to:

1. Provide both a global (peripheral) and a detail (focused) view, simultaneously. We note that this approach has already been used extensively in the Ontario Telepresence Project [9][8][68] by combining the two views through a picture-in-picture device. The same approach with multiple views was also proposed by Kuzuoka et. al. [44]. Howev-

3. The Extra Eyes system was a collaborative project of Kimiya Yamaashi and Jeremy Cooperstock.

er, as will be discussed later, providing a link between the two views is not only critical for usability, but also supports the goal of multiple views while avoiding the pitfalls of spatial discontinuities inherent in the MTV studies [29][35].

2. Provide a navigation mechanism using these views, allowing users to redirect their view in both direction and scale, through a simple user interface.

However, even with these two goals satisfied, the user is still sensorially deprived to the extent that it may inhibit social interaction. Therefore, our third goal is as follows:

3. Provide a sensory surrogate or prosthesis to compensate for the limited scope of visual information.

The interested reader can find a full discussion of the Extra Eyes system in Section 5.3.

5.2 Head Tracking for View Control

This section presents the implementation details and an evaluation of our head tracking system, based on the Virtual Window concept [30]. The reader not interested in these details may skip to Section 5.3 for an overview of the Extra Eyes system, or proceed directly to Section 5.4 for the discussion relating to design principles.

5.2.1 System Architecture

We used a computer with a frame grabber as the controller, and a motorized video camera to provide the attendee with a dynamic view of our conference room. The attendee's image is provided to the frame grabber, as shown in Figure 21. An important point to recognize is that no special equipment, other than the camera, is required at the site of the attendee. The attendee need only send his or her video image to our media space, where all of the processing is performed.

The system software operates in two stages. Stage 1 computes the position and size of the attendee's head, while stage 2 provides camera control. The video image of the attendee is captured by the frame grabber and provided to stage 1. Since the current bottleneck in our system is the transfer of image data from the frame grabber to the computer, we use a quarter-size gray-scale image of 180 x 120 pixels to minimize the cost of this operation.

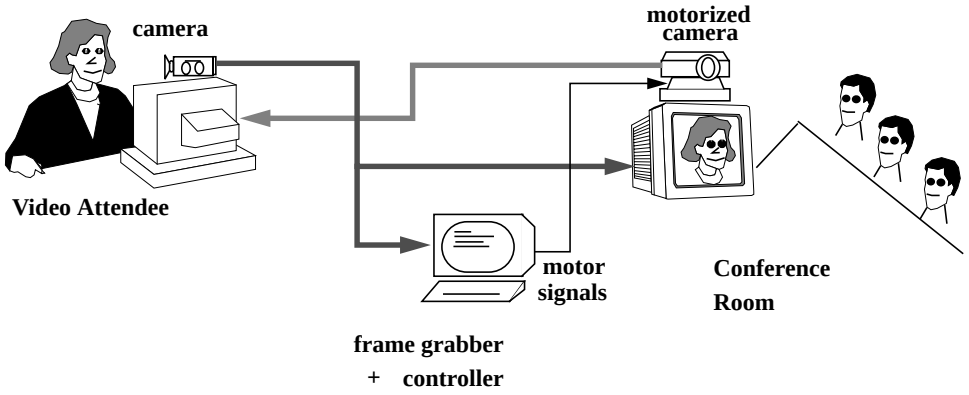

FIGURE 21 Configuration of equipment at the media space and remote site. The video image of the attendee is provided both to attendees in the local conference room, and to a frame grabber that processes the image to provide camera control motor signals.

Stage 1 produces a differential image from the current video image and a reference view, typically the first frame captured before the person entered the scene. This results in removal of the background, leaving only the attendee in the image. The top of the attendee's head is then located by scanning this image for a contiguous region of non-zero pixels, corresponding to those points that did not appear in the reference view.

In stage 2, the head position is translated to orientation of the motorized camera. In order to reduce susceptibility to noise and potentially irritating constant small-scale camera motions, we apply a low-pass filter to the head parameters and further require that any changes to the camera orientation be of some minimum magnitude. The ratio of head width to frame width is also calculated and used to determine the zoom factor. These parameters are then sent to the camera through a serial interface. Stage 1 and 2 are repeated indefinitely, approximately twice per second. The operation of this system is illustrated in Figure 21.

5.2.2 Head Detection

Due to noise in the video signal or changes in lighting conditions, pixel intensities of the background image may vary from frame to frame. As it was our intention simply to implement a basic version of the Virtual Window concept for videoconference

environments, we concentrated our investigation on three simple methods of head-detection. The interested reader is referred to the work of Hunke and Waibel [37] for a discussion of state of the art head tracking techniques.

Our initial attempt involved a comparison of the current video image with the previous frame to produce a differential image. Any pixel whose intensity had changed significantly was considered relevant. All other pixels were discarded. The problem with this approach is that little or no head movement between successive frames results in a sparse differential image of the head, which may be overwhelmed by background noise.

A second attempt involved preprocessing of each image by convolution with an edge-detection filter, and using the resulting frames to produce the differential image. Although edge information is stable under variable lighting conditions, background noise remained a problem. Furthermore, the loss of detail of the face made it difficult to locate the user's head.

Our third approach was to produce the differential image using an initial reference frame, taken without the attendee in the scene. While the need to begin the process with the user out of the camera view may be inconvenient, the results are far more stable than previous methods. Surprisingly, we found that this method also works fairly well in a large number of cases in which the reference image contains the attendee. However, this approach suffers under lighting variations or camera perturbations.

Improved capabilities are offered by faster frame grabber hardware and software, both from a technical and user perspective. Hunke and Waibel demonstrate a promising approach using preprocessed, colour-normalized input images fed into a connectionist architecture [37]. These researchers noted that normalized pixel values of red, green, and blue are relatively close for most skin colours. In order to achieve a faster and more robust head detection process, this observation led us to modify the previously described algorithms by ignoring all pixels whose normalized colours were outside of a threshold bound around the average values. However, we found that the bounds required for successful head detection were often so large that a great deal of background pixels were also included. To cope with this problem, Hunke and Waibel incorporated a neural network colour classifier in their system. Seeking a simpler, although less elegant approach, we included a calibration phase, in which the user is asked to sit directly in front of the camera while the face colour values are determined. Because the calibration process

makes the user acutely aware of the technology, this violates our design principle of invisibility. A potential solution to this problem is discussed later in Section 5.4.4.

5.2.3 Camera Control

Camera control techniques generally fall into two categories: position and velocity. Position control requires that the desired pan and tilt angles are provided directly to the camera. While this method is simple and accurate, current camera technology results in slow movements, and does not permit the interruption of a movement.

With velocity control, the controller provides a velocity vector to the camera and instructs it to stop motion when the desired orientation is reached. Through feedback, corrections to the movement may be provided. This is quick and interruptible, although more complex than position control. For simplicity, we are presently using position control.

Due to the half second delay in image processing, neither method operates in true real time. There may be a lag of several seconds for the camera to reach the correct orientation, especially for large head movements. However, our users found this delay to be quite tolerable in practice, provided that the camera begins moving within a short time of the head motion. Otherwise, users become confused. With faster video processing capabilities now available, these camera control methods need to be reconsidered.

5.2.4 Observations

Control of camera orientation via head translation seems to pose no problems on the horizontal (yaw) axis. However, dependency on translation for vertical (pitch) control can be unnatural, since people normally use nodding-like motions, not head translations, to gaze up or down. Our initial hypothesis was that with an improved head tracking mechanism, it would be possible to track facial features and use these to provide pitch control by head orientation. While this may be true, our subsequent experiments with a six degrees of freedom input device, attached to a baseball cap worn by the user, revealed the drawbacks of head orientation as a means of controlling the camera[4]. Since the monitor

4. These experiments were performed locally, as part of the Extra Eyes system development, discussed in Section 5.3.

through which the remote user viewed our site remained stationary, changes in head orientation required the user to assume awkward eye orientations.

Despite its limitations, we see this approach as being equally beneficial to other scenarios besides videoconferencing. Camera control via headtracking would be particularly advantageous in tasks such as teleoperation of a robotic device or medical surgery, in which the user's hands are required for other tasks.

5.3 Extra Eyes for Multiple Views

This section presents an overview of the Extra Eyes system, illustrating how each of our three goals was achieved, and provides the results of our user study. The reader not interested in these details may skip to Section 5.4 for the discussion relating to design principles.

5.3.1 Supporting Foveal and Peripheral Cones

It has been suggested by several vision researchers that a brain mechanism exists to drive foveating saccades[5] of the eye in response to stimulus in the periphery region [34][85]. In the discussion of their model of saccadic eye movement, Tsotsos et al. comment that these saccades play an important role in the exploration of the visual world [75]. Supporting evidence for this comes from neurophysiology. A region known as PO, which receives a representation of the periphery of the visual field, has been identified in the brains of primates [14]. Deprived of this information, individuals suffering from tunnel vision, or a loss of vision outside the fovea, exhibit severe problems navigating through their physical surroundings, even when these surroundings are familiar to them [32]. With this in mind, it becomes readily apparent that camera-monitor mediated telepresence is bound to suffer unless peripheral vision can be supported concurrently with a detailed, foveal view.

Overlaid Multiple Views

As an initial attempt to provide this support, we developed a prototype system, consisting of a large and small display, as shown in Figure 22. The large screen display provides the user with a wide angle view of the remote space while the small display provides a high

5. Foveating saccades are rapid eye movements intended to direct the fovea to a desired orientation.

resolution view of the area of interest. With the camera orientations fixed and the proper geometric positioning of the two displays, spatial discontinuities are minimized. The sensation of increased peripheral awareness obtained by this system is very powerful.

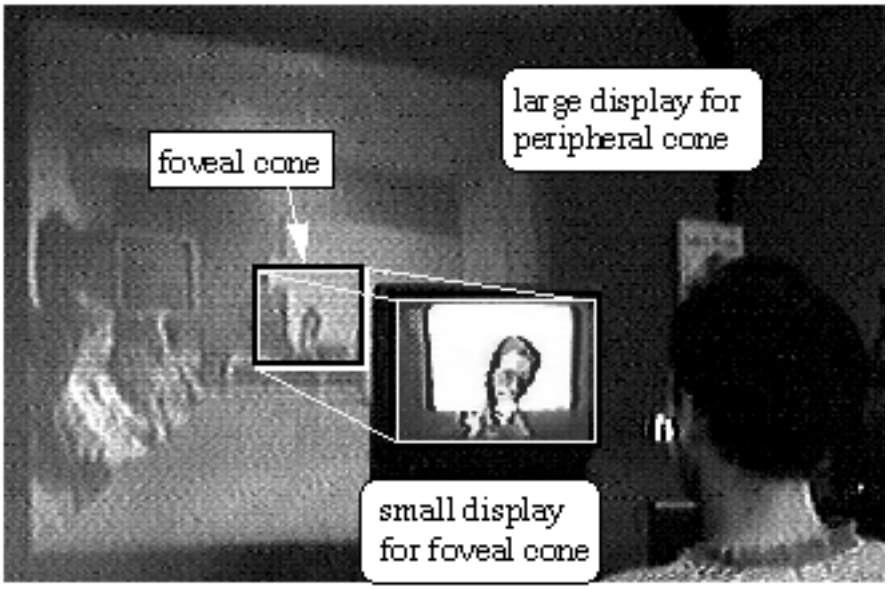

FIGURE 22 This prototype system uses a large screen display for the peripheral view and a small screen for the detail view.

We note that this prototype requires two high-resolution displays, one of them quite large, in order to achieve a significant effect. As this may be prohibitively expensive for most videoconference users, we would like to unify the two views into a single display. Unfortunately, even on a large screen display, the limited resolution would make the quality of the foveal region unacceptable. Another approach is required.

Disjointed Multiple Views

The approach we took was to display both the foveal and peripheral views separately on the same screen. Since the views are disjointed, each can have sufficient size and resolution, even with the limitations of current technology. Our implementation of this system is shown in Figure 23. The top portion of the display provides a foveal or detail view, obtained from a user-controlled motorized camera, while the lower portion provides the global view from a fixed, wide angle camera. Note that due to the small screen size, the global view is observed by the user through foveal, not peripheral vision. Thus, unlike

our earlier, large screen prototype, this implementation does not replicate the mechanics of biological vision. Rather, we are simply attempting to offer the user a high-resolution detail view, without sacrificing the important contextual support provided by the global view.

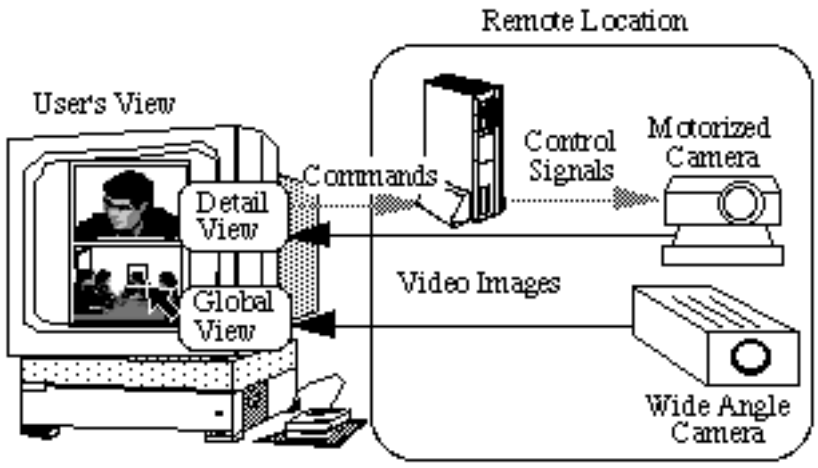

FIGURE 23 Architecture of the Extra Eyes system.

Since the views are independent of each another, there is no consistent geometric relationship between the two. This can result in an inability to locate the position of the detailed region within the peripheral view, once more bringing us back to the problem of spatial discontinuities. Navigation under these conditions is typically difficult and slow. This is especially severe when the scene being viewed is relatively homogeneous (e.g. through tele-education, a large class of students). Normal human vision does not suffer from this problem because the direction of the fovea explicitly dictates the peripheral view.

Linking Multiple Views

To address the lack of a geometric relationship between the two views, we indicate the detailed region within the global view by means of a yellow bounding box (detail frame), as shown in Figure 24. The enclosed region corresponds exactly to what is displayed in the detail view. As the detail view changes, the bounding box on the global view adjusts accordingly.

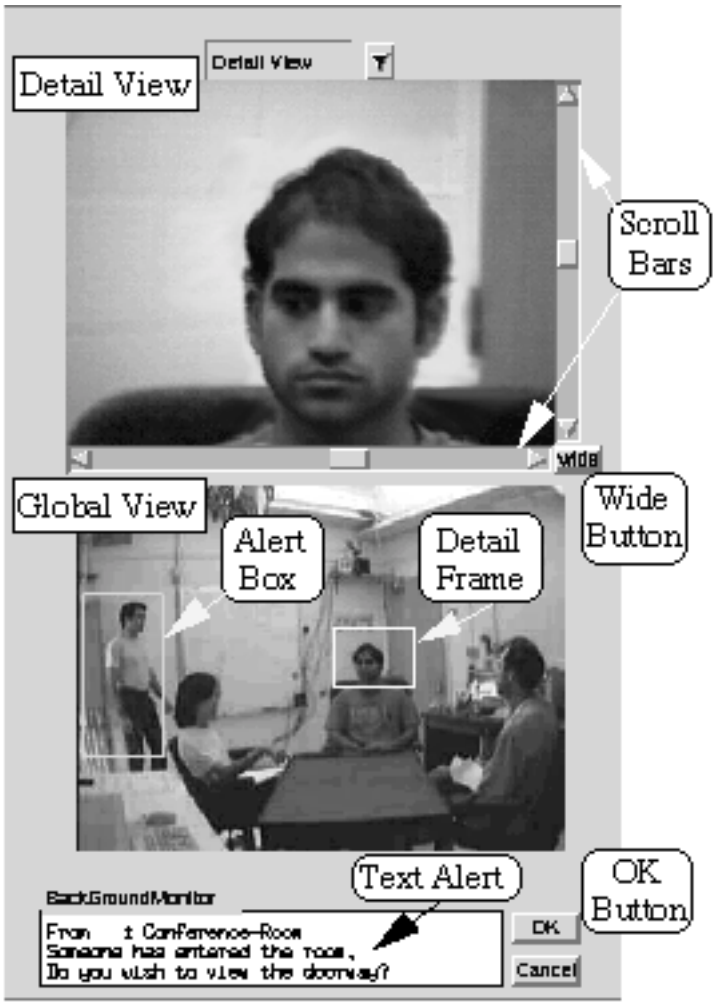

FIGURE 24 Screen layout of Extra Eyes.

Because the two views are logically linked, users can select a desired region by sweeping out a bounding box or simply *point-and-click* on the global view. In the former case, the detail view is defined by the size of the bounding box, while in the latter, the detail view is centered at the selected position and displayed at the maximum zoom. These interaction techniques with the global view permit a far more efficient navigation mechanism than the

effectively *blind* [6] view selection offered by both the original MTV system [29] and the Virtual Window system [30].

In addition to control via the global view, the detail view can be manipulated directly through the scroll bars, which provide tilt and pan control of the motorized camera. It is also possible to adjust the zoom factor of the detail view by pressing the left or right mouse button, or obtain a wide view by selecting the *wide* button.

To provide a linkage between the global and detail views, we require a mapping between the coordinate systems of each, dependent on the properties of the different cameras. We first define a global coordinate system, which covers the entire area visible to both cameras. Next, we define models for each camera, which consist of a view model, and in the case of the motorized camera, a transformation function. The models describe the relationship between pixel coordinates of each camera and the global coordinate system. In the case of our fixed wide angle camera, this is simply a one-to-one mapping. The transformation function for the motorized camera maps pixel coordinates to the appropriate motor signals. The models and relationships are described in Figure 25.

When a user selects an area of the global view, the pixel coordinates of this region are first translated into global coordinates through the wide angle view model, and then into pixel coordinates of the detail view. The detail pixel coordinates are then mapped into motor signals via the transformation function. Finally, the motor signals are sent to the detail camera. At the same time, the updated location and dimensions of the bounding box are computed, and displayed on the global view. Similarly, when a user specifies an area of the detail view directly, the pixel coordinates of this region are transformed into motor signals for the camera, and to global coordinates describing the new bounding box.

5.3.2 Sensory Surrogate

There exists no substitute for physical presence that offers the fidelity of rapidly directable stereo vision and spatially sensitive binaural audio, as manifested by the human senses. To help bridge the gap between physical presence and telepresence in this regard, our Extra Eyes system provides users with a *sensory surrogate* to increase their awareness of the

6. We use the term, *blind*, because no visual information apart from that appearing in the single view is available.

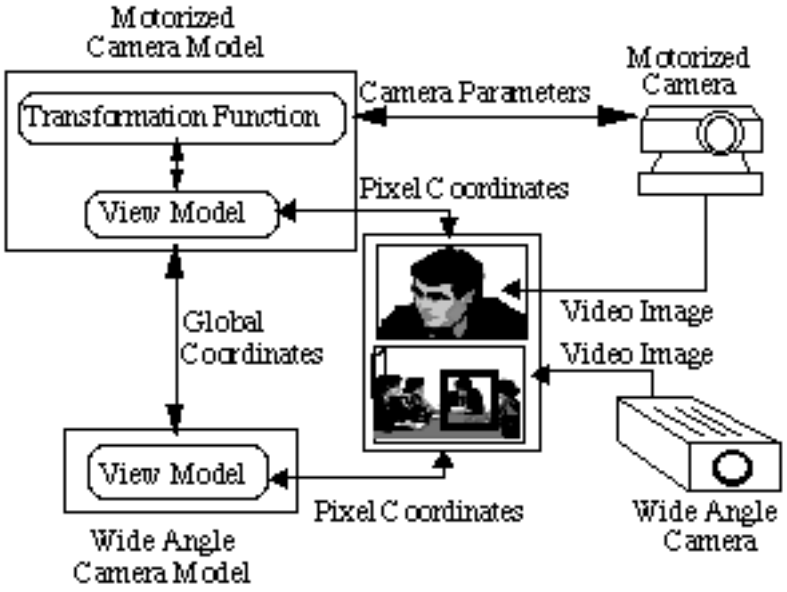

FIGURE 25 Camera models and their relationships.

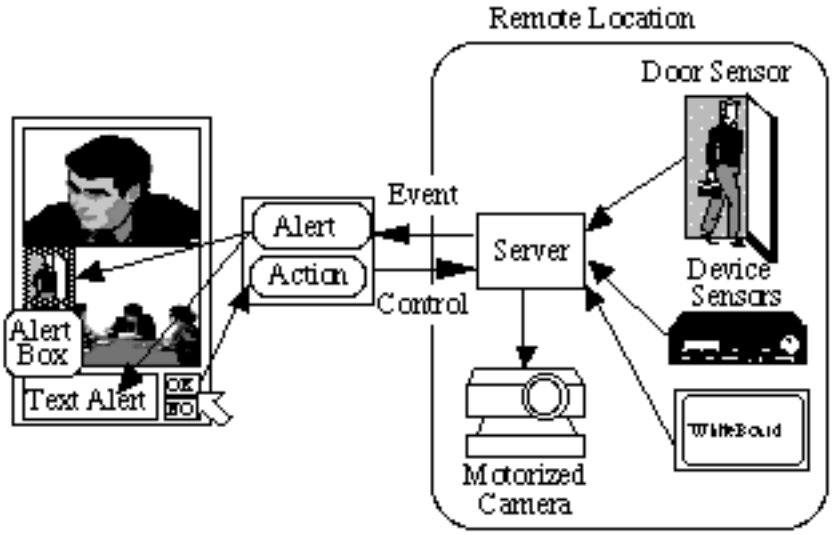

FIGURE 26 The sensory surrogate in action.

remote environment. The surrogate monitors background information obtained by sensors and reports on relevant events through the use of sound, text, and graphics, or a combination of the three. In this manner, background processing by the computer is used to improve the user's foreground awareness.

Sensors in the room [20] monitor the status of presentation technology such as the VCR, document camera, and digital whiteboard, as well as the entry of new individuals as depicted in Figure 26. When an event occurs, it triggers an *alert-action sequence*. The alert corresponds to the screen message displayed (e.g. “Someone has entered the room. Do you wish to view the doorway?”), as well as the appearance of a blue bounding box (*alert box*) in the corresponding region of the global view, as shown in Figure 24. If the user acknowledges the alert by pushing the *OK* button or selecting the *alert box*, then an appropriate action is executed by the system (e.g. control the motorized camera to display the doorway). Another possible alert message is “The VCR is now playing. Do you wish to view the tape?” with the associated action of switching the user's view to the VCR output.

FIGURE 27 Screen layout of Postcards. Images from each room are captured periodically.

The sensory surrogate concept has also been applied to increasing social awareness among individuals sharing the media space of the Ontario Telepresence Project [68]. The Postcards system (see Figure 27), based on Rank Xerox EuroPARC's Portholes [23], captures snapshots from each user's office at set intervals and distributes these to members of the media space. A sensory surrogate in the Postcards system compares every two consecutive frames from each office to determine if there is activity there.[7] This is done by counting the number of pixels that have changed by more than a certain threshold amount between the two frames. Although the algorithm is susceptible to false detection of activity due to camera perturbations, it has worked reasonably well in our environment. Stored knowledge of activity allows Postcards to determine whether individuals are in or out, or have recently entered or vacated their offices.

Users can take advantage of this background monitoring feature by asking the system to *sense* activity and notify them when any number of individuals are simultaneously present in their offices. This permits informal group meetings to be established with a minimum of effort, freeing the user from the mundane task of repeatedly checking to see who is available.

These ideas were evaluated through a user study, described in detail in Appendix D. The interested reader is invited to consult this section.

5.3.3 System Issues

The remainder of this section explores some system issues concerning the current and alternate implementations of Extra Eyes.

Bandwidth

Detractors may argue that transmitting video for the global view is too expensive. Either more bandwidth is required, or the frame rate of the detail view will suffer. We suggest that since the global view is only required to provide a sense of peripheral awareness, both its frame rate and resolution can be relatively low. In fact, we reduced our global view to a quarter size (160 x 120 pixels), and found that users were still very aware of activities

7. This frame differencing system was originally implemented by Luca Giachino, a visiting researcher from CEFRIEL, Italy. Postcards was written by Tracy Narine of the Ontario Telepresence Project.

occurring in the periphery. If the global view is transmitted at this size, along with a full-frame detail view, both at the same rate, then the decrease in frame rate of the detail view would be less than 7%, assuming constant bandwidth consumption. We strongly believe that the benefit of peripheral awareness justifies this minor expense.

Widening the Field of View

The global view provided by our present system cannot capture a view of the entire room. Other designers may prefer to use multiple cameras, or a very wide angle lens, possibly a fisheye, for this task. In the former case, some form of image processing will be required to combine the images, while in the latter, unwarping to compensate for image distortion will be necessary. This idea has been explored by Warp California's Virtual TV (VTV) system, which selectively unwarps any portion of the image from a fisheye lens. As higher resolution and lower cost frame grabbers become available, this technology will offer many advantages over motor-driven cameras.

Towards Realistic Vision Mechanics

While the sense of peripheral awareness offered by a fixed global view is a helpful navigation tool, it does not accurately replicate the mechanics of human vision, in which the periphery is dictated by the orientation of the fovea. As a result, we have extended the Extra Eyes system by using a second motorized camera to provide the global view. With this new system, any change of either the detailed or the global view causes the orientation of the other to change as well. To maximize effectiveness, we have located the smaller display near the center of the large screen. This way, the foveal and peripheral cones maintain the correct geometric relationship at all times.

Beyond the Prototype

Extra Eyes was originally implemented as a prototype using an analog video connection for the detail view and an internet connection for the slower global view. While this was sufficient as a proof of concept, we were also interested in demonstrating the system to a much wider audience. This motivated us to increase the accessibility to our media space and begin thinking about new ideas regarding the future of human-human communication. These ideas are explored in Appendix E.

5.4 Design Principles

In the preceding sections, we described the design and implementation of several systems aimed at improving the ease with which remote attendees can interact with our media space and exploit its technological functionality. Having done so, we now turn to examine how our design principles have been applied toward this end.

5.4.1 Invisibility

The AVSA and head tracking systems both demonstrate how useful background processing can be performed by the computer, without physically intruding in the workspace. From the perspective of the remote user, these systems require no additional equipment beyond the video camera, monitor, and microphone that are already in place for videoconferencing.

While the AVSA requires the explicit interaction of voice commands, it is clear that tasks such as navigating through a media space cannot dispense entirely with the interface. Hence, our aim, as discussed in Section 3.2.1, should be simply to minimize the intrusion of the interface as much as possible. In this case, the support of user-invoked video-overlay menus coupled with speech recognition permits a style of interaction that is much closer to human-human communication than the traditional GUI would allow. Furthermore, the AVSA offers the practical benefit of reducing the amount of equipment required at the remote site.

By comparison, the head tracking system does effectively hide the interface. The remote user carries out no explicit interaction with the computer, yet enjoys far more control over the received view. In our preliminary testing, we found that the capability of controlling the orientation and zoom factor of a camera in a remote location adds significantly to the user's sense of engagement in our media space. Because its use only requires individuals to perform the everyday action of looking through a window, anyone can use our system effectively with no special skills or training.

As it is intended to be a GUI computer application, Extra Eyes makes no attempt at hiding the interface. However, by linking the peripheral and detail views, the system allows users to carry out almost all of their navigational tasks without requiring conscious processing. Invisibility may not have been obtained but the obeisance of Ecological Interface Design

(EID) leads to a system that is inherently far more usable than a conventional interface. Beyond the linkage of views, what sets Extra Eyes apart from other videoconferencing tools is the integration of our design principles of manual override and feedback. These lead to a system that is pioneering in its support of awareness of a remote environment.

5.4.2 Manual Override

The usability of each of the systems described in this chapter relies strongly on manual override capability. For the AVSA, which responds only to direct command requests, this is obvious. Therefore, in the following discussion, we will deal with the importance of manual override to the head tracking system and to Extra Eyes.

A potential problem with head tracking for camera control is that the video attendee must remain still in order to continue viewing the same scene. To address this issue, we initially introduced a freeze mechanism that locks the camera orientation if the attendee's head remains relatively stable for a certain period (ten seconds, in our implementation). This permits the attendee to select a desired view and once it is locked, freely move about. A simple gesture, such as covering the attendee's own camera lens, can be used to unlock the camera movement and resume head-following. However, these mechanisms are not intuitively apparent and hence, do not satisfy our requirement of seamlessness. Without training and conscious human information processing, users would by unable to exercise these features.

The solution to this problem arrived with the AVSA. By incorporating video overlay and speech recognition, we could provide new users with the necessary instructions (e.g. "say *freeze* to lock the camera position") to invoke the function. Admittedly, this is not truly invisible, but we hope that after limited experience, users could exercise this function with a minimum of cognitive effort. The AVSA-style approach additionally offers users the ability to label several views, and later return to these by simple verbal instruction.[8]

Extra Eyes offers an interesting example of how the benefits of automatic, reactive behaviour may be improved by the incorporation of a manual override mechanism. To understand this point, it is worthwhile to begin by considering how the remote user

8. To date, however, only the freeze mechanism has been implemented.

without manual override capabilities would experience a presentation run from the Reactive Room. In general, the attendee would receive the output of the device most recently activated, possibly in combination with a picture-in-picture view of the presenter. When no device is active, the received image would consist of a fixed camera view of the presenter. While this scenario is adequate for passive participation in a videoconference,[9] it is unacceptable if the intent is to reduce the attendee's sense of physical distance.

In contrast, Extra Eyes allows users to select desired views dynamically by controlling the orientation and zoom of the motorized camera. Furthermore, when a new device is activated, the attendee is offered the option of switching views, rather than being presented with the new view automatically. Together, these two features afford active exploration of the remote environment and engender a greater sense of control, thereby permitting more effective interaction.

5.4.3 Feedback

The importance of effective feedback was repeatedly made clear throughout the development of the AVSA. In the early prototype stages, the speech recognition system was highly unreliable.[10] As a result, it was often necessary to repeat simple voice commands several times before the desired command was invoked. Without visual feedback from the recognition system, we found it extremely difficult to anticipate when and how our speech was being interpreted. To minimize the probability of incorrect interpretation, the command set vocabulary was deliberately chosen to be very small, consisting primarily of the numbers *one* through *five*. However, this design decision led to a potential problem if users lacked confidence in the recognition of their menu selections and repeated them unnecessarily. In such cases, multiple commands could accidentally be invoked in response to a correct interpretation of these utterances.

Addressing these problems entailed the provision of feedback in two forms. First, the status of the recognition system is displayed in a visible, yet unobtrusive, window of the video overlay screen, so that users can observe how their utterances are being interpreted.

---

9. In fact, as comments from remote attendees to the Reactive Room have suggested, this is a great improvement over the status quo.

10. We were using the PC-based Voice Assist package during the prototype development. The current version is running on an SGI Indy with much improved performance. Note that neither of these systems should be considered state of the art with respect to speech recognition capabilities.

Second, upon successful recognition of a command word, a text overlay message appears, indicating that the command is being invoked, for example, “Responding to fast-forward request.” This provides additional confirmation of successful recognition and also serves to warn the user that a visible effect of this command may not be immediate [15]. Once the command has successfully completed (in the case of our example, when the VCR has confirmed that the tape is being advanced), the text message disappears, indicating termination.

In the case of Extra Eyes, video output from the motorized camera serves as direct feedback for control operations. Additional context information is provided by the bounding box drawn on the global view. Finally, text and audio alerts bring important events to the user’s attention. The relevant factor here is that apart from the minor cognitive overhead of reading occasional text messages, users receive all necessary feedback from the system in a direct, visual manner.

5.4.4 Adaptability

While none of the systems described in this chapter presently exhibit adaptability, some theoretical discussion of this design principle is in order.

The head tracking system could benefit greatly from adaptability to individual users. It should be possible to obviate the calibration phase, described in Section 5.2.2, by dynamically modifying the expected facial colour values. This could be accomplished by observing the values of those pixels that change between successive frames. Barring excessive background noise, these pixels would most likely correspond to motion of the user.

For Extra Eyes, adaptability would be strongly suited to interaction with the sensory surrogate. While the control offered by this system provides remote attendees with a greater sense of engagement in our media space, some users would prefer that the system reverts to the automatic view selection normally provided by the Reactive Room. This mode could be invoked after noticing that a particular user is always agreeing to the change of view suggested by the sensory surrogate. Naturally, our design principle of manual override stipulates that the user would maintain the ability to cancel the automatic mode at any time, simply by exercising a manual view selection. These points are particularly relevant to our World Wide Web-based version of Extra Eyes, described in

Appendix E, since the sensory surrogate messages for that particular implementation are often highly distracting.

5.5 Summary

While human vision permits rapid gaze control and peripheral awareness, camera-monitor mediated interaction deprives telepresent users of these affordances. Further limitations on the user's ability to navigate through a conventional media space often reinforce the sense of physical distance and lead to a poor sense of engagement of remote users. Through the Virtual Window head tracking system, the Audio/Visual Server Attendant, and Extra Eyes, we addressed these limitations and demonstrated the applicability of our design principles.

The shortcoming of gaze control was addressed by tracking the head positions of remote attendees, and using these to control the orientation of a motorized camera in our environment. This allows users to select their own view, using the natural metaphor of looking through a window. Taking this approach one step further, the Audio/Video Server Attendant allows remote attendees to navigate through our media space using natural language commands and observing the effects directly. The principles of EID are evident here, as is our design principle of invisibility, since the interface is hidden and the remote attendees need not even have a computer.

Extra Eyes provides electronic attendees with two simultaneous, linked views to help overcome the lack of peripheral awareness. One view offers a wide angle, global perspective, while the second presents the desired region in detail. Building on our previous work, we utilized sensor data from the augmented environment to provide telepresent attendees with feedback concerning relevant events, such as the entry of an individual into the room or a status change of a presentation device. Since users have the option of ignoring this information, or responding to it by requesting that the view be changed to the appropriate device, the system supports a seamless manual override. In combination, the linkage of views and the sensory surrogate significantly increase the user's ability to sense and respond to important events, thereby leading to an improved sense of engagement.

Table 5 provides a summary of the embodiment of our design principles in each of the three systems described in this chapter. Since adaptability has not yet been addressed in any of the current implementations, this column is not included.

System	Design Principle		
	Invisibility	**Manual Override**	**Feedback**
Head Tracking	uses everyday skill of looking through a window	locking of camera view; voice commands through AVSA	
AVSA	user gives commands in natural language and does not require a computer	implicit, since all actions are invoked manually	speech recognition display and text-overlay messages indicating command status
Extra Eyes		direct specification of desired visual area; choice of view	bounding box in global view; text and audio alerts

TABLE 5 Design principles by system.

Chapter 6

Contributions, Future Directions, and Conclusions

But the only way of discovering the limits of the possible is to venture a little way past them into the impossible.

ARTHUR C. CLARKE [13]

When we first proposed the concept of Reactive Environments as a solution to the problems of complexity plaguing the users of our videoconference room, several critics voiced their concern that this approach would never succeed. They argued that recognizing the goal behind each human action and reacting to it sensibly was beyond the capability of a background computer process. Although this is certainly the case for present-day models of general human behaviour, we have shown that the same does not necessarily hold true for a reasonably constrained environment.

6.1 Contributions

This thesis can be summarized in terms of its major contributions, listed as follows:

- An extensive exploration and validation of the Ubiquitous Computing design concepts in an imbedded environment for doing real work;
- extending the UbiComp model with a context-sensitive approach to deal with the complexities governing the control and coordination of multiple devices;
- demonstrating the interdependence of the design principles of invisibility, seamless manual override, feedback, and adaptability, through the proof of concept Reactive Room, which simplifies control of technology to the degree where effective coordination of the presentation technology requires minimal training, little distraction, and low cognitive demands on the user;
- promoting a design methodology that involves a partitioning between primary tasks, i.e. those that can be performed easily by most users, and secondary tasks, i.e. those which require background support from the technology, based on an inventory of users' skills at both the motor-sensory and social levels;
- introducing simultaneous multiple linked views and a sensory surrogate to improve peripheral awareness in videoconferencing.

As mentioned in the introduction, this research was intended to expose relevant issues in the design of usable technology and motivate further exploration of our proposed approach. 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. Before embarking on such a study, an attempt to formalize the results of this research may be worthwhile. As a first step toward this end, a model should be developed, describing the interactions between user and system that must take place in a reactive, as compared to a traditional, state of the art, videoconference environment. A good starting point may be to consider applying a goal and task hierarchy model such as *GOMS*[1] [12] or a linguistic model such as the *task-action grammar (TAG)* [61] to the device control tasks of Appendix A.

1. GOMS is an acronym for Goals, Operators, Methods and Selection.

Early feedback encourages our belief that the Reactive Environments 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 in which 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.

6.2 Future Directions

The serious problems of life are never fully solved.
If ever they should appear to be so, it is a sure sign
that something has been lost. The meaning and
purpose of a problem seem to lie not in its solution
but in our working at it incessantly.

C.G. JUNG

We now consider where Reactive Environments can and should go from here. It must first be recognized that this approach is applicable to a wide range of problems in which high-level interaction with the environment specifies a context in which certain reactive behaviours are appropriate. For such problems, interaction with the computer is not, inherently, the goal, but rather, a level of abstraction that replaces direct manipulation of some other device, be it a video switch or the flight controls of an aircraft. This is in contrast to computer-centric tasks, such as visualizing a complex mathematical function or sending an electronic mail message, which are not amenable to our methodology. For

these activities, direct human-computer interaction, in the form of view specifications or simple text entry, must dominate.

6.2.1 Extensions of the Model

The purely reactive approach to the operation of technology is useful for the chosen videoconferencing task, but it will likely prove to be insufficient for larger, more complex, dynamic environments. Rather, we must consider a hybrid approach that integrates deliberative, symbolic planning and reasoning with a non-deliberative, reactive mechanism, as does Ferguson's *TouringMachine* architecture [26]. The relevance of such extensions to autonomous agents is broader than theoretical interest, especially in light of the role that agents already play in safety-critical systems.

Additional research is required into the issue of adaptability. Learning the limited range of behaviours that the Reactive Room encountered could be considered a toy problem. Similar positive results have been shown for system prediction of repetitive user tasks through programming by demonstration [22][55], including the TELS system [54], which generalizes iterative operations to procedures that can contain branches and loops. Attaining similar results for larger, more complex, and distributed tasks remains a challenge being tackled by a number of researchers in the field of machine learning [53].

6.2.2 Application to Other Domains

One direction to consider is applying the principles of our design methodology to other technology-rich environments, such as in aircraft, industrial plants, nuclear power stations, and emergency medical rooms. These examples are particularly important because of the consequences of the design to issues of human safety. While these domains have already witnessed a great deal of automation, system implementations often suffer from poor or limited manual override and diagnostic feedback [58]. Adaptability may be inappropriate for such environments, which should, according to other researchers, consist of single use, dedicated and formally proven systems that are immutable.[2]

---

2. The author would like to thank Heather Hinton of the Department of Computer Science, University of Toronto, for this suggestion. In this context, "formally proven" refers to the concept of computer system reliability, that is, the behaviour of the system is consistent and deterministic.

However, it is perhaps more appropriate and even more interesting to turn our attention to environments that have traditionally been less technology-rich, such as our homes. As a result of the gradual infusion of devices as simple as thermostats and light switches to more complex telephones, microwave ovens, and entertainment systems, we spend a good deal of our lives interacting with home technology, often because its behaviour (or lack thereof) is not appropriate to the context.

Several large projects dealing with so-called “smart house technology” emphasize security and high-tech gadgetry, but little attention has been given to the far more important issue of usability. For example, the highly publicized home of Bill Gates, controlled by 100 microcomputers, will feature digitized artwork displayed on the high-resolution monitors imbedded in the walls. Lights will gradually turn on and dim as guests walk down a hallway and music will follow them from room to room. To enjoy this functionality, guests will be required to carry special pin-style active badges so that their location can be tracked. While the technology may be dazzling, little mention has been made of how visitors will be able to perform such basic tasks as turning off the music without finding the appropriate computer. Even more troubling are the potential consequences of a misplaced or non-functional badge.

Other visions of the future include the inappropriate placement of a personal computer on a kitchen counter, as pictured in Figure 28. What makes this photograph absurd is that the computer monitor occupies valuable counter space and the interface consists of a traditional keyboard and mouse.[3] Note however that the concept of computing power in this environment is entirely justified. Many kitchen appliances today already come equipped with microcontrollers and touchpad interfaces. The issue is not whether the computer belongs, but in what form it should take and what function it should provide.

Most homeowners are unlikely to invest thousands of dollars in a microwave oven that heats up their dinner in response to a telephone call from the airport.[4] However, there may be significantly more interest in a microwave that remembers how hot you like your dinner. Similarly, a security system that can be programmed to monitor certain rooms selectively via a keypad interface may be useful, but one that provides the same

3. It is unlikely that such a configuration would survive more than two days in the author’s kitchen.
4. This is typical of the “smart house” functionality, which advocates claim will soon be needed by us all.

FIGURE 28 The kitchen computer. Photo courtesy of Compaq Canada.

functionality by allowing the operator to click on a live video display of each room, is far more usable.[5]

6.2.3 Implications for Consumer Electronics

At home I used to have a very intelligent VCR with near perfect voice recognition and knowledge of me. I could ask it to record programs by name and, in some cases, even assume it would do so automatically, without my asking. Then, all of a sudden, my son went to college.

NICHOLAS NEGROPONTE [57]

Again, we must ask whether technology is improving our lives, or only complicating our activities. How often must we lower the stereo volume when we are speaking with someone on the telephone and how many times has a ringing telephone disturbed our dinner? It is well within reach of current technology for the stereo to be aware of our use of the telephone and similarly, for the telephone and answering machine to be aware of our dinner hour. So far, however, these issues seem to have been neglected.

5. The author has already created such a prototype that can be run over the World Wide Web.

While society is being overwhelmed by the functionality now offered, advocates of technology tell us that we will adapt to the complexity of its operation. Marcus cites a survey of R.H. Bruskin & Associates, indicating that one-third of American VCR owners have given up programming these devices because they cannot understand the instructions and controls [51]. For these users, it is clear that technology has already passed their threshold of frustration. If the manufacturers of consumer electronics recognize this problem, they are hopefully thinking about products such as a VCR that turns itself on when you sit in front of the television or automatically records your favourite program if you forget to watch it one week. Such reactive behaviour would benefit most users far more than the suite of functions offered by the latest thirty-button remote controls.

We note that precisely this sort of evolution took place in the 35mm camera industry. Consumers were first offered a very limited device, one with a single button to release the shutter, one lever to advance the film, and another to rewind it. The photographer had only two decisions to make when taking a picture: *what* to photograph and *when* to press the shutter release, in other words, “point and shoot.” Later, as technology progressed, amateur photographers were overwhelmed and frustrated by the multitude of dials, switches, and buttons on the modern, supposedly “improved” cameras. Eventually, consumer forces made their impact on the market, and manufacturers began supplying cameras with auto-focus, auto-aperture, auto-shutter, etc. These could now be used as true “point and shoot” cameras, but could also be operated in full-manual mode, in which case, the user must set all of the switches and dials. In other words, consumer demand called for a device that could be operated with minimal cognitive effort, but on demand, would allow low-level control of individual functions.

The difference between the camera analogy and our conference room is that in the latter, coordination of multiple devices is required for correct operation, whereas the former is almost always used in isolation. Our problem is that we live in a modular world in which components have been designed as individual entities with no capability of talking to one another. The combination of present frustration with the status quo and elevated expectations of technological capability leads us to anticipate a growing demand for consumer electronics that behave as imagined above. Companies such as X-10, Echelon, Home Systems Plus and IBM are already manufacturing and promoting home automation systems, utilizing many sensors of various types that could be applied to construct a Reactive Environment. Unfortunately, current deployment of the technology has so far

been limited to glorified remote controls, just as with conference room control systems. As was demonstrated by the Reactive Room, it is not through fancy user interfaces but through coordination and sharing of context that interesting and useful behaviours emerge.

A key to making this type of interaction commonplace is the establishment of a standard protocol allowing appliances to be linked together easily. Several such possibilities exist or are currently in development. The Electronic Industries Association designed the Consumer Electronics Bus (CEBus) for transmission of data and control information over a variety of transport media, ranging from home power wiring to fiber-optic cable. Several other standards also exist for low speed peripherals, including the Universal Serial Bus (USB) being promoted by Intel, and Apple's GeoPort. A likely candidate for the universal standard is the versatile, high-speed IEEE 1394 Standard for a High Performance Serial Bus [38]. This protocol, also known as FireWire, has already been adopted by a large number of manufacturers. However, another possibility to consider is the next generation Internet Protocol (IPv6) [6]. The task force responsible for the design of IPv6 has in mind the concept that eventually, every television set will become an Internet host [36]. It is conceivable that in the future, most electronic devices in our homes and offices will be similarly networked and capable of communicating with each other. Beyond the basic protocol, a uniform data and command format must be adopted at the higher levels of inter-device communication. One such protocol, already being promoted in Japan, is the flexible TRON Application Databus (TAD) [70].

As a side note, it is interesting to consider that similar problems face developers in the software domain. The failure of integrated application suites (e.g. Microsoft Office) to impact the way people use computers can be attributed, at least in part, to the limits of potential imposed by the applications themselves. The solution, Raskin[6] proposes, is for vendors to supply interoperable command sets such as spell checkers, equation solvers, and music functions, rather than bundled applications [65]. Just as the Reactive Room daemons communicate with one another so that videoconference equipment can be operated seamlessly, so too should computer command sets cooperate. In both cases, we

6. Jef Raskin was the creator of the Macintosh project at Apple Computers. Raskin's philosophy can be seen in part in the concept of the Macintosh clipboard, a facility that allows users to copy, cut, and paste objects seamlessly between any applications, regardless of the data types involved

should be able to perform our tasks without being concerned by details of the technology, whether we are using presentation devices or computer software.

6.3 Conclusions

Having concluded the prototype effort and explored its implications, we now turn to several questions that will ultimately determine the success of our approach.

6.3.1 Social Issues

In constructing the tools of our media space, we were frequently reminded of the many factors governing social acceptability of the technology. With respect to our particular situation, foremost among these was the issue of privacy. If its use is not fully understood, even a single video camera or microphone can create an atmosphere of distrust. To overcome the “Big Brother is watching” fear, several important rules were followed from the earliest stages in the design of our Telepresence media space [68]. These rules follow a common theme of making our electronic interactions approximate our interactions in the physical world as closely as possible. While they are particularly suited to the personal space of one’s office, they are equally valid for a more public area, such as a videoconference room:

- *Door states*: just as I can open, close, or lock the physical door to my office, so too can I control my electronic accessibility;
- *Audio alerts*: when you glance at me or connect to my node, I receive notification in the form of a doorbell or other appropriate sound;
- *Reciprocity*: if you can see me, I can see you;
- *Physically accessible desktop devices*: I can always cover the camera lens or turn off the microphone.

In the Reactive Room, these rules are still obeyed, except for Internet visitors, where reciprocity must be sacrificed in favour of accessibility. As a partial solution, we display the host name of the machine connected to the room, to provide some indication of the visitor’s identity. However, with our media space wide open to the outside world through the AVSA and the World Wide Web, the notion of door states is especially important.

While the ability to define different levels of accessibility for local and remote visitors is currently lacking, this will likely be an important provision for the future.

6.3.2 Scalability

The videoconference environment was selected as our initial research testbed because it was technology-rich, yet relatively limited in terms of number of possible states and events. This latter point was imperative in achieving our design goals in a manageable amount of time. With the prototype completed, we must now consider the scalability of our approach, especially as it relates to future efforts in other problem domains.

The Reactive Room easily accommodates new devices and sensors by the simple insertion of a set of device descriptors along with a modest programming effort for the device interface and specification of the event-generated messages and associated reactions. However, the need for explicit programming every time an architectural change is made poses a problem for much larger systems. While some programming associated with the device interface and signal processing cannot be avoided entirely, message generation and associated reactions could be semi-automated, possibly based on additional device descriptors. This approach is likely the best-case scenario with respect to present-day software capabilities, and could certainly support the construction of much larger systems than the Reactive Room.

6.3.3 The Interface Consistency Problem

By assuming responsibility for the high-level switching of audio and video signals, the Reactive Room hides the computer-based user interface, but leaves the physical device interfaces exposed. The user must still control the zoom and lighting settings of the document camera, fast-forward or rewind video tapes to the desired positions, and raise or lower the monitor volumes. For one or two distinct devices that require such manual operation, this does not pose a serious problem. However, as the number of devices grows to include multiple VCRs, slide projectors, and audio systems, each with a different physical interface or remote control, the complexity of the technology soon reappears.

In our own conference room, we were initially faced with six distinct physical interfaces and remote controls for the VCR, document camera, video monitors, whiteboard display, motorized camera, and the PiP device. While the Reactive Room reduces our reliance on

some of these and entirely eliminates the need for others, we still find ourselves toggling switches, pressing buttons, and fumbling with remote controls, each with a different layout. Perhaps such device manipulation is unavoidable, but if so, the interface complexity must be reduced for the technology to be acceptable to users.

One approach to this problem is to unify the manual control of all these devices into a single package, such as a large control panel or a touch-screen interface. While this reduces the number of distinct interfaces, it does not alter the complexity. Putting all of the buttons together in one package, as has been done for several room-control systems, simply overwhelms the user with choices. This problem can be avoided by organizing the controls in a hierarchy-by-device, as the number of options presented at each level is more manageable, but the obvious drawback is the need to navigate through a menu to reach the desired functions. In either case, centralizing the control of multiple devices in a single, unified interface has the potential for greater consistency, but suffers from the same abstract representation problem that confronted us in Section 2.4.

An alternative approach worthy of consideration is a gestural interface that would, for example, allow users to invoke actions such as lowering the volume by pointing at a video monitor and then towards the ground. More powerful possibilities exist, especially if such a system were used in combination with a voice recognition system. Presenters could point to a VCR and say, “fast-forward two minutes,” or hold two fingers on a page under the document camera and say, “zoom to these corners.” Like our button-and-light modules and the laser pointer tracker, this system eliminates the abstraction problem by allowing the user to indicate the physical device itself rather than an arbitrary representation. This interface could be standardized to permit the same gesture or verbal phrase to have similar effects across multiple devices, while still recognizing and responding to variant inputs for greater robustness and flexibility.

6.3.4 Final Remarks

Extending the UbiComp paradigm, Reactive Environments break through the barriers of traditional keyboard-and-mouse computing and offers us 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.

Appendix A

Anatomy of a Videoconference

Without the benefit of reactive technology, operating a videoconference environment poses a formidable challenge. Even before a presentation or meeting can begin, turning on all of the equipment in the room can require a great deal of effort. In our own videoconference environment, shown in Figure 29, this process typically involves turning on (1) the room lights, (2) the computer monitor, (3) the document camera, (4) the VCR, (5) the large data monitor, and (6) three smaller television monitors. This requires a minimum of three switches and the use of multiple remote controls. If the room control system (the Desk Area Network application) and the videoconference manager (the Telepresence application) are not presently running on the computer, these applications must be started as well.

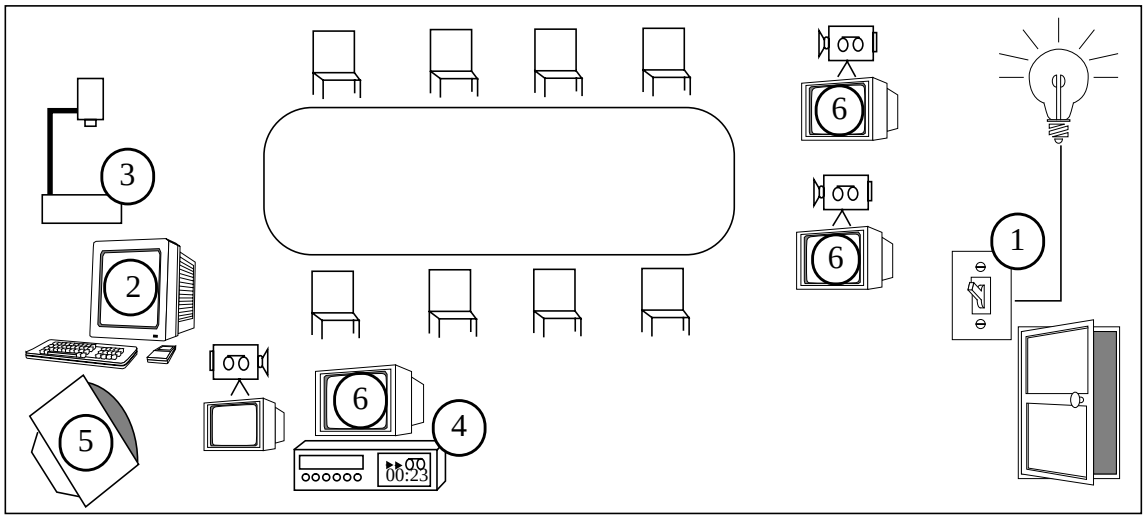

FIGURE 29 The steps required to turn on the equipment in the conference room in preparation for a meeting or presentation.

When a remote attendee joins the meeting, the audio and video outputs from a microphone and camera in the remote location are typically connected through a codec (an audio/video modem) to a speaker and monitor in the conference room, and similarly in the reverse direction [9]. This scenario is pictured in Figure 30.

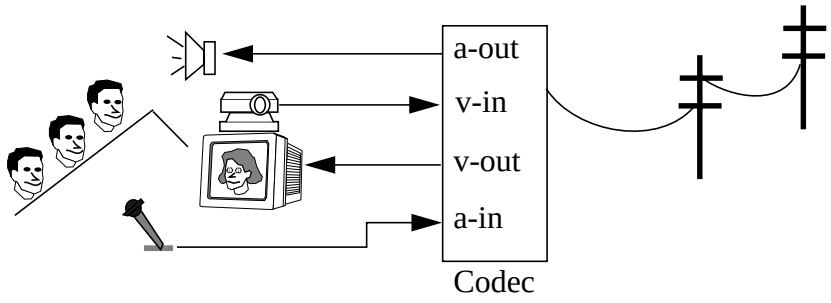

FIGURE 30 A simple videoconference configuration.

This configuration is by no means static. Figure 30 illustrates what happens when the presenter plays a videotape. The video and audio outputs from the VCR must be routed to an appropriate monitor and speaker, while the same signals are sent to the remote attendee. In order to ensure that all participants can still hear the presenter, audio output from the VCR must be mixed with that of the microphone before reaching the codec. Similarly, for visibility of the presenter, video output from the VCR must be combined with the camera through a picture-in-picture (PiP) device.

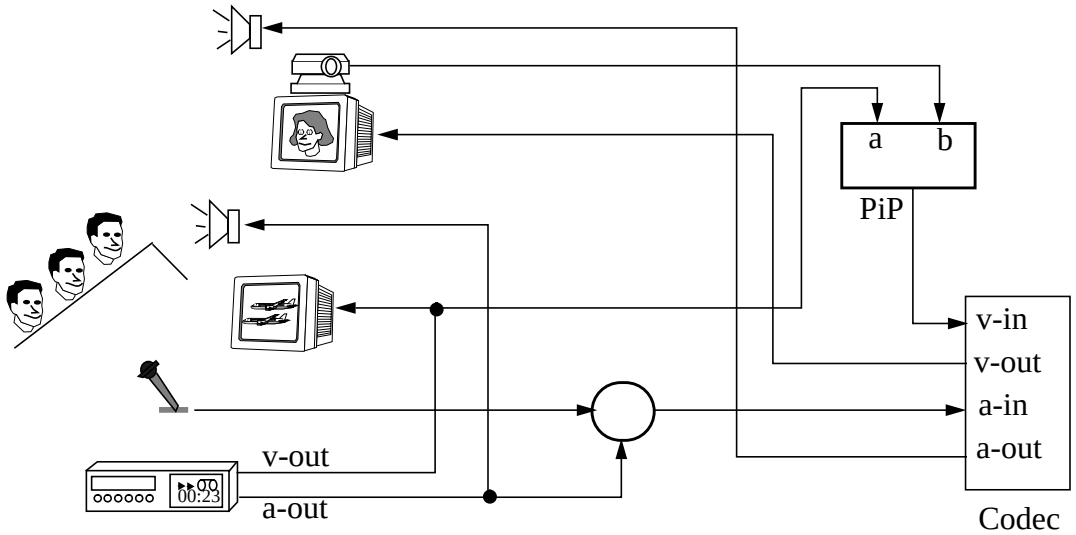

FIGURE 31 Playback of a videotape.

Note that the preceding example assumes the existence of multiple monitors and speakers in the videoconference room. The wiring would be quite different if only one monitor or speaker was available.

Now, consider the new configuration that is required if the presenter wishes to record the meeting, as shown in Figure 30. This time, audio and video from the remote attendee must be combined with that of the presenter and provided as input to the VCR.

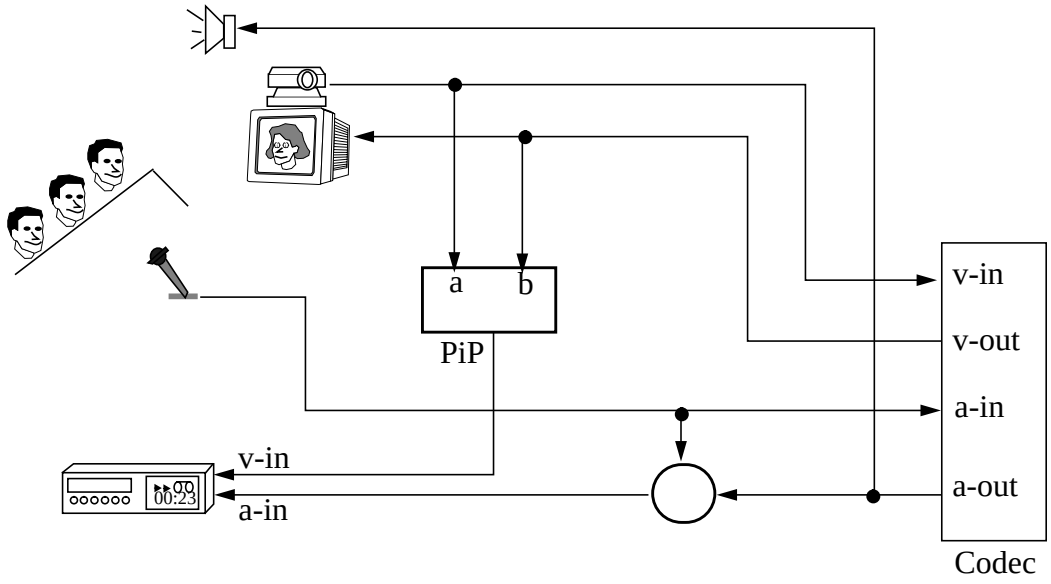

FIGURE 32 Meeting capture by videotape.

If, while the meeting is being recorded, the presenter uses the document camera to display an overhead transparency, it may be appropriate to record the document rather than a camera view of the presenter. Regardless, the document should be provided to the remote attendee. However, since the PiP is being used by the VCR, the remote attendee will no longer be able to watch the presenter as well.[1]

1. Naturally, an alternative configuration is possible, in which the remote attendee receives a PiP view of both the document and the presenter. With one PiP device available, this means that the VCR could only record the video from a single source. The device configuration file for the Reactive Room, described in Section 4.5.3, permits users to codify these rules, simply by specifying whether a remote visitor or the VCR has priority for the PiP.

Obviously, it is far too much to expect the typical presenter to manage such wiring manually. Instead, most videoconference systems allow some form of preset button for each of the configurations described above. This means that playing or recording a videotape is a multi-step operation in which the preset is selected and then the VCR is operated [9]. Each time the presenter uses a new device, a different preset must be invoked.

To complicate matters further, special circumstances often arise that necessitate alternate configurations of equipment. Hence, any attempt to provide high-level control should not limit users in their ability to configure the detailed low-level connections manually. Otherwise, as we have witnessed in many scenarios, the system is bound to be insufficient and its use frustrating. In contrast, the Reactive Room allows a seamless low-level manual override, while simplifying the general case scenario. For the above examples, all that is required is for the user to place the tape in the VCR and press the appropriate button, either *play* or *record*. Based on the context, the appropriate configuration is invoked automatically.

Appendix B Daemon Implementation

Daemons check for requests from each other and regularly perform background monitoring functions. The generalized daemon processing loop is illustrated in Figure 33.

```
Register(daemon_name)
do
     if requests pending from another daemon
          ServiceRequests
     BackgroundProcessing
until Terminate
```

FIGURE 33 Generalized daemon processing loop.

These processes run on distributed hosts and communicate with each other over network connections (see Figure 34). Daemons share much in common with intelligent agents [40][49][46], but differ in the sense that they do not make use of any AI planning or reasoning techniques. Instead, our daemons are either primitive finite state automata or, at best, adaptive rule-learning systems with sensory input. Following the reactivist school of robotics, popularized by Brooks [7], we hope to obtain complex emergent behaviour from the interaction of these simple processes. In terms of augmented environments, this concept is similar to the notion of a “distributed cyberspace” [41], in which many objects act together to support a rich set of tasks. In our case, the set of tasks involves the control of equipment in the videoconference environment. By transferring responsibility for this

control to the Reactive Room, we reduce the cognitive load on users and hence, the distraction of technology.

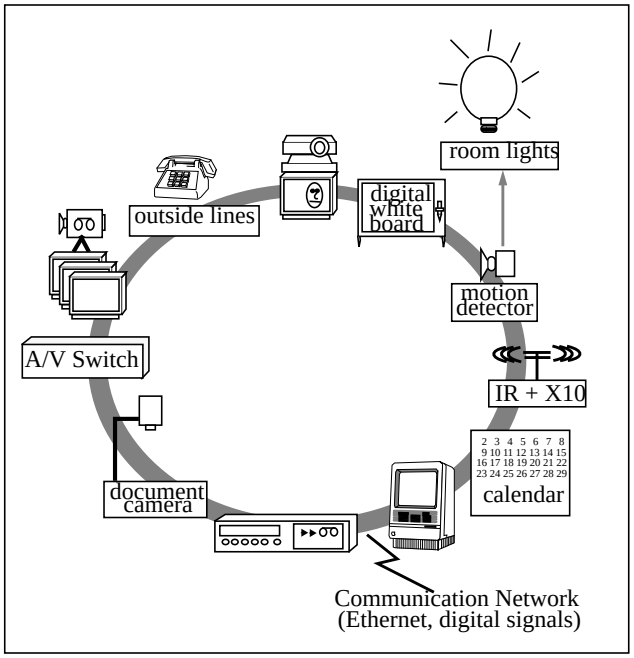

FIGURE 34 The Reactive Room contains a variety of videoconference equipment. Associated with each device is a software daemon, which communicates with other daemons in order to control the equipment in support of the presenter's activity.

B.1 Daemon Functions

Although not strictly accurate in terms of implementation, this section provides a description of the daemons that monitor and control the Reactive Room. The daemons are arranged in the three categories of general videoconference control, device co-ordination, and computer resource management. All communication initiated by each daemon is explicitly noted.

B.1.1 General Videoconference Control

The first category of daemons perform the general-purpose functions of controlling the equipment within a videoconference environment. While the calendar daemon is, strictly

speaking, non-essential for such control, its centrality to the operation of the room merits its inclusion here.

- **DAN (Desk Area Network)**[1]**:** runs the audio and video switches in a manner appropriate to the current presenter. When a connection is made with an electronic attendee, the DAN displays the attendee on a local monitor and ensures that the attendee receives an appropriate view. The DAN also establishes any connections requested by the user through the button-and-light modules, described in Section 4.2.2.
- **Telepresence**[2]**:** establishes and monitors teleconference connections with remote attendees or presenters.
- **Signals:** controls the X-10 (computer-switched) power supplies, the button-and-light modules, and the motorized drapes, and monitors various digital signals such as the motion detector output. Since people often make casual use of the conference room to view video tapes without first booking the room, the signals daemon also ensures that the monitors are turned on, and switched to the correct display mode when a tape is played.
- **IR:** generates the infrared commands required to perform various operations such as turning on monitors when the room is activated. Since the signals daemon is often able to perform such functions independently, using the X-10 supplies, most daemons issue their requests for device control to the signals daemon, which then decides whether or not to forward the command to the IR daemon.
- **Laser:** executes selected commands based on the location of the end point of the laser beam, as described in Section 4.2.2.
- **Calendar:** maintains a schedule of room bookings, created from user requests through a World Wide Web-based form interface[3]. The calendar daemon notifies other daemons when a scheduled booking time has

---

1. The name, “Desk Area Network” is an historical artifact of the Ontario Telepresence Project. The original DAN application was based on the metaphor of a desktop patchbay that allowed users to select what they saw and what they sent, from a number of possible A/V sources and destinations.
2. Again, the name is an historical artifact, referring to the main Telepresence application that provides connectivity to other users in our media space.
3. The calendar interface was written by Chitra Brahme, an intern student in our group.

arrived, so that the room can be configured appropriately for the current presenter.

B.1.2 Device Co-ordination

The second category of daemons are specialized processes, each dealing with a particular videoconference presentation device. While the tasks that each daemon performs are relatively similar, the implementations are not.

- **Document:** monitors the video output of the document camera.
- **VCR:** uses a control-L[4] interface to control the VCR and polls its status to determine when buttons such as play or record have been pressed.
- **Whiteboard:** monitors the status of the microswitch installed in the pen holster. When the pen is picked up, the switch opens, indicating that the digital whiteboard is in use.
- **Motor Camera:** controls a motorized camera as part of a head-tracking system for remote attendees, described in Section 5.2, or in response to requests from the World Wide Web interface, described in Appendix E.

B.1.3 Computer Resource Management

The final category of daemons deal with the computer resources of video frame grabbers, audio devices, and accessibility to the Internet. Since these resources must be shared between several processes, each is managed by a dedicated daemon, which accepts and services requests for each.

- **Video:** grabs and digitizes a frame from the selected video input. For the World Wide Web interface, the digitized frames are then returned in JPEG format. However, to avoid the unnecessary overhead of transmitting video frames between processes, the video daemon locally carries out most of the image processing required by other daemons, such as the document and laser daemons, and passes back the results of the computation.
- **Audio:** generates beeps and other diagnostic feedback sounds in response to user activity, and plays greeting messages when users enter the room. This

4. Control-L is an inter-device communications and control protocol developed by Sony.

daemon is also responsible for generating warning messages or audio cues when the room is being viewed or heard by a WWW visitor. As part of its background monitoring, the audio daemon also polls the other daemons, once per hour, and produces a warning message when any are not running (and someone is in the room to hear the warning).

- **WWW:**[5] allocates time slots to World Wide Web users who wish to view or control our media space, and selectively grants or denies them access permission. The WWW daemon is also responsible for notifying Internet visitors of relevant activity taking place in the room, and displaying text messages, entered by WWW visitors, on a computer console in the Reactive Room.

B.2 Communication

To ensure reliable exchange of messages, daemons use TCP [64] as the network protocol, making calls to a standardized set of socket functions, provided by the AFDP [19] socket utilities library. Each message transaction is either a send-receive or a blocking send-receive-reply, as determined by the sender. In either case, the connection is immediately torn down at the conclusion of the transaction. Send requests are typically short command strings or status information conveyed between daemons, while reply messages contain the data requested, or serve as acknowledgments indicating the success or failure of the previous command. The format of messages is shown below in Figure 35.

Asynchronous (non-blocking) send-receive-reply transactions are sometimes necessary for real-time video processing, in which the daemon requesting video frames cannot afford to block while waiting for the data. In this situation, the requesting daemon simply uses a send-receive message to instruct the video process to send a frame when it is ready.

Since the Reactive Room technology was first deployed in a prototype setting, and later replicated in our actual conference room, it was necessary to distinguish between the various daemons that are responsible for the control of each environment. For this

5. The World Wide Web interface allows Internet visitors to control the motorized camera and interact with the videoconference equipment while obtaining live audio and video. However, these services are not provided by the WWW daemon, itself, but rather, by a number of `cgi-bin` programs that request permission from the WWW daemon.

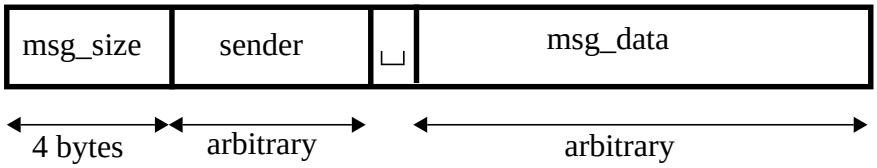

FIGURE 35 The message format used for daemon communication. The lengths of the sender and msg_data fields are arbitrary, so long as their total is less than the maximum message size (currently 5000 bytes), minus one. The msg_size field is a `long int`, the sender field is an array of `char`, while msg_data can be any type.

purpose, a different session name is assigned to the two sets of daemons. To facilitate communication, each daemon registers itself with a file-based name-server on start up. The name-server maintains each daemon's name, session, process id, host, and port number. When one daemon wishes to communicate with another, it first checks its cache for the host name and port number of the target daemon. If no cache information is available for that target, the name-server is consulted, instead. The name-server is also consulted when an attempt to communicate with a known target fails, usually as a result of the daemon being restarted on a different host or port.

For debugging and analysis purposes, daemons write important transactions to log files. Each log message is prepended by the current date and time. For robustness, minor exceptions are logged and execution is resumed whenever possible. However, when a daemon is forced to terminate on a trapped signal, it first unregisters itself from the name-server so that other daemons will not attempt to send messages to a non-existent process.

B.3 Daemon Interaction

Communication between daemons provides the basis for complex interactions. For example, the document daemon may instruct the VCR daemon to record the document view rather than the whiteboard, during the taping of a meeting. Similarly, the VCR daemon may tell the signals daemon to turn on a monitor when a tape is played. 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 are sufficient to trigger the appropriate sequence of events.

In general, context plays an important role in determining the intended behaviour 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” (PiP) device. However, if no remote participants are attending, then the VCR need only record the local view. In our reactive environment, knowledge of whether an electronic visitor is attending the meeting is obtained through communication with the Telepresence daemon, responsible for controlling connections to our media space.

Interaction between daemons can be arbitrarily complex. For example, returning to our sample scenario at the beginning of Chapter 4, when Nicole enters the room a second time, the signals daemon notifies the calendar daemon that motion was detected. This causes commands to be issued to the DAN, signals, and Telepresence daemons. The Telepresence daemon establishes a connection with the conference room, and then notifies the DAN that an electronic visitor is now active. Meanwhile, the signals daemon sends requests to the IR and VCR daemons. The latter, recognizing that it is unable to activate the VCR through the control-L interface, asks the IR daemon to transmit a `vcr_power_on` command. This process is illustrated below in Figure 36.

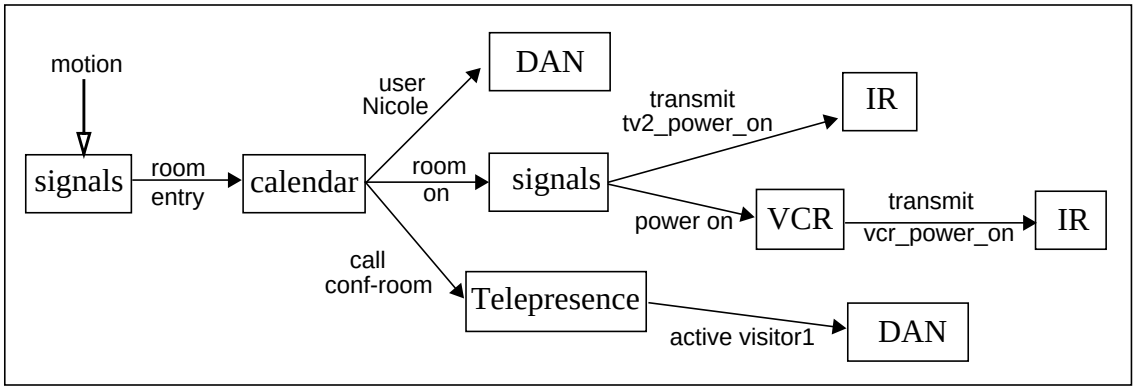

FIGURE 36 The sequence of events that occur when a scheduled presenter enters the Reactive Room. The hollow arrow on the left indicates the motion detector output, provided to the signals daemon. Each box denotes a daemon, and the solid arrows between them denote messages. For simplicity, this diagram does not include details of the operations performed directly by any of the daemons, for example, the signals daemon control of the X10 power supplies or the Telepresence daemon establishing a connection with the conference room.

Appendix C

Smart Light Switch Implementation

This appendix describes the implementation of the Smart Light Switch, pictured in Figure 37.

The state diagram of Figure 12 is converted into the state table of Table 6, which addresses potential instabilities arising from asynchronous circuit design and assigns sensible state transitions for all possible values of input variables. While it is recommended that such a circuit be used in practice, it is possible to simplify the state table by ignoring all of the *don't care* conditions. This leads to the derivation of the following Boolean expressions for the control logic signals:

$auto = (\overline{on} \bullet \overline{off})$	(internal signal, indicating automatic mode is in effect)
$light = (\overline{\overline{off}} \bullet motion) + on$	(light is on)
$manual\text{-}off\text{-}reset = off \bullet \overline{motion}$	(resets the manual off override mode)

This logic was incorporated into our actual light controller, which drives the room lights via a pair of 40 VDC relays. The schematic is shown below in Figure 38.

The design and implementation of the Smart Light Switch was a collaboration with Sidney Fels, then a graduate student in the Neural Networks group in Computer Science.

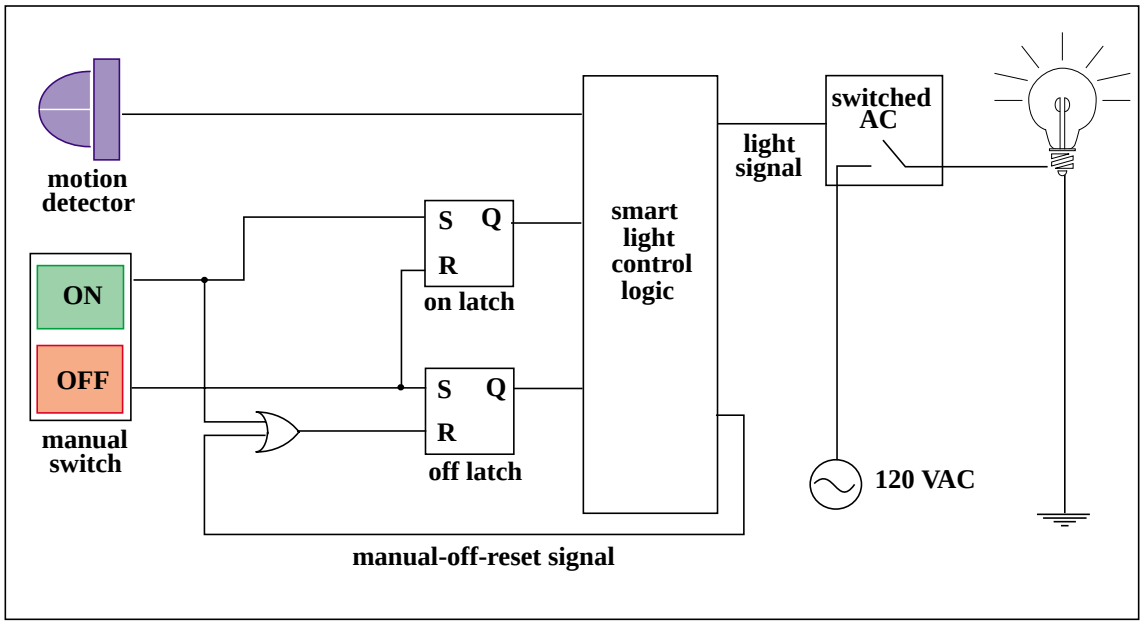

FIGURE 37 Architecture of the Smart Light Switch.

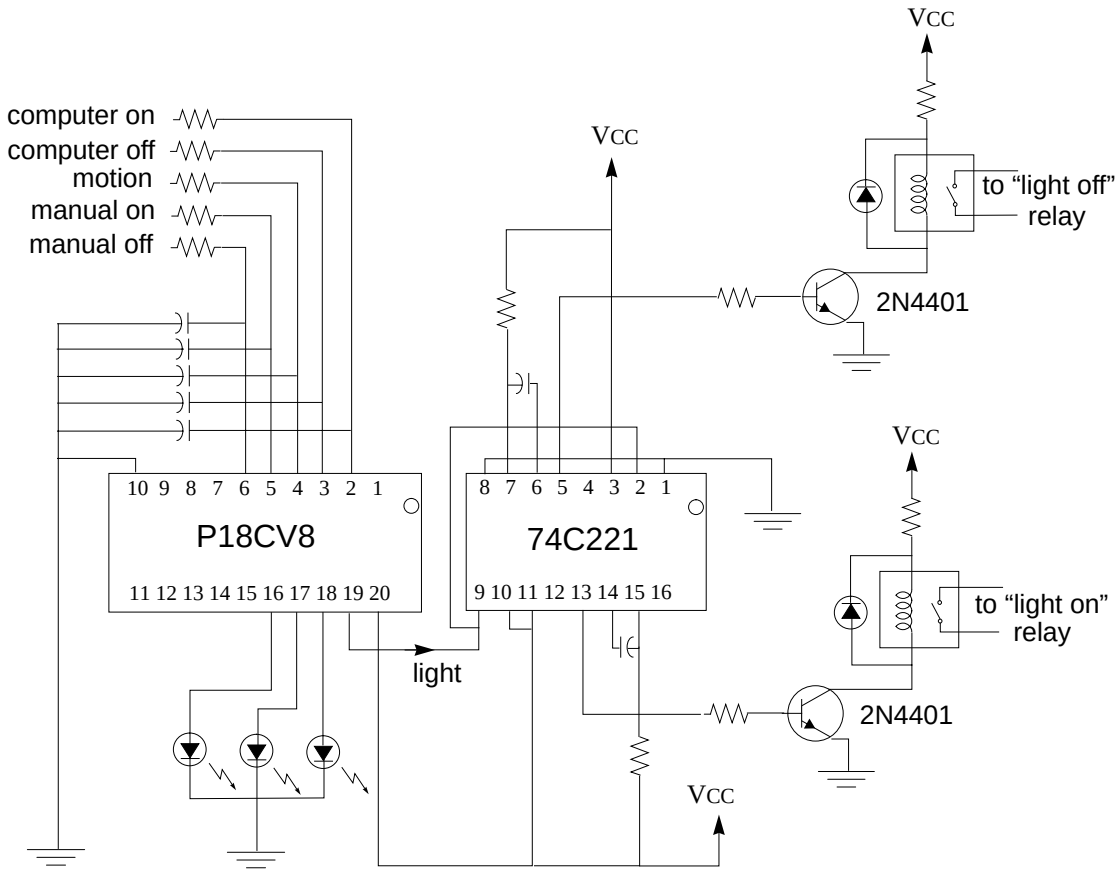

FIGURE 38 Schematic diagram of the Smart Light Switch and controller.

State				Switch		Next State		
label	auto	light	motion	OFF	ON	label	auto'	light'
manual off	0	0	0	0[a]	0	auto off	$\overline{OFF} \bullet \overline{ON}$	$\overline{OFF} \bullet ON$
	0	0	0	0	1	man. on		
	0	0	0	1	0	man. off		
	0	0	0	1	1	man. off		
	0	0	1	0	0	man. off	0	$\overline{OFF} \bullet ON$
	0	0	1	0	1	man. on		
	0	0	1	1	0	man. off		
	0	0	1	1	1	man. off		
manual on	0	1	0	0	0	man. on	0	$\overline{OFF} + ON \bullet O$
	0	1	0	0	1	man. on		
	0	1	0	1	0	man. off		
	0	1	0	1	1	man. on		
	0	1	1	0	0	man. on	0	$\overline{OFF} + ON$
	0	1	1	0	1	man. on		
	0	1	1	1	0	man. off		
	0	1	1	1	1	man. on		
auto off	1	0	0	0	0	auto off	$\overline{OFF \oplus ON}$	$\overline{OFF} \bullet ON \bullet O$
	1	0	0	0	1	man. on		
	1	0	0	1	0	man. off		
	1	0	0	1	1	auto off		
	1	0	1	0	0	auto on	$\overline{OFF \oplus ON}$	$\overline{OFF} + ON$
	1	0	1	0	1	man. on		
	1	0	1	1	0	man. off		
	1	0	1	1	1	auto on		
auto on	1	1	0	0	0	auto off	$\overline{OFF \oplus ON}$	$\overline{OFF} \bullet ON$
	1	1	0	0	1	man. on		
	1	1	0	1	0	man. off		
	1	1	0	1	1	auto off		
	1	1	1	0	0	auto on	$\overline{OFF \oplus ON}$	$\overline{OFF} + ON$
	1	1	1	0	1	man. on		
	1	1	1	1	0	man. off		
	1	1	1	1	1	auto on		

TABLE 6 Smart Light State Table with bidirectional return-to-center switch.

a. OFF = 0 is achieved by resetting the OFF flip-flop when $off \bullet \overline{motion} = 1$.

Appendix D Extra Eyes User Study

We evaluated the performance of Extra Eyes through the following user study. Three television monitors were arranged in a remote location, as shown in Figure 39. Letters of the alphabet were displayed on a randomly chosen monitor, one at a time. The user's task was to use the Extra Eyes system to identify these letters as they appeared, as quickly as possible, while minimizing the number of errors. Each letter would remain on the monitor until the user had identified it, by typing its corresponding key. Once the letter was identified, it would be replaced by another letter on a different monitor. The font size was sufficiently small so that a zoom factor near the maximum was required for legibility.

D.1 Experimental Conditions

Our seven subjects were selected from a population of seasoned computer users, all of which had extensive experience using a mouse and GUI. All were given a brief demonstration of the system in operation before beginning the experiment. We tested each of these subjects on the following conditions, the order being randomly varied, with 20 repetitions per condition:

1. **No Global:** Only the detail view is visible. This situation is equivalent to typical telepresence systems.
2. **No Global + Text:** Same as 1. In addition, a text alert indicates the display on which the current letter appears.
3. **Unlinked:** Both the global and detail views are simultaneously visible, but are not linked (i.e. neither view has effect on the other). This is equivalent to the MTV system.

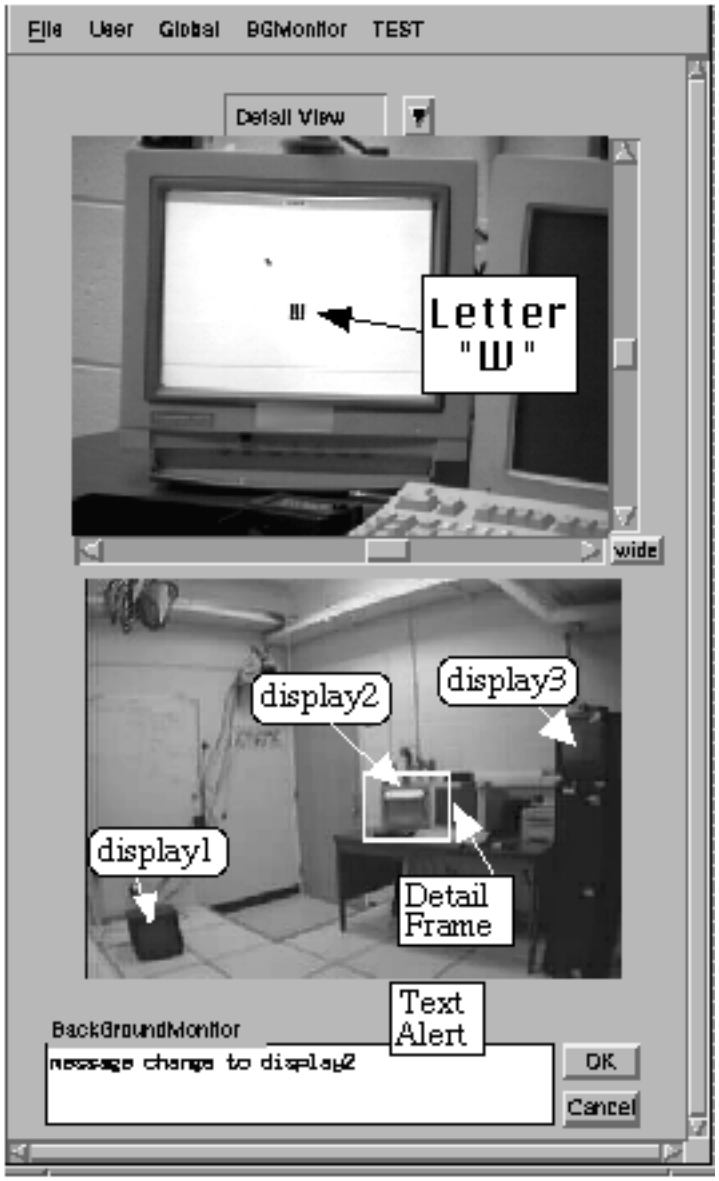

FIGURE 39 Configuration of user study.

4. **Linked:** Both the global and detail views are simultaneously visible and linked.
5. **Linked + Text:** Same as 4. In addition, a text alert indicates the display on which the current letter appears.
6. **Linked + Action:** Same as 5. In addition, an *alert box* appears, and the user can invoke the action corresponding to the alert by pushing the *OK* button or by clicking anywhere within the *alert box*. The action causes the camera to point directly to the new letter with maximum zoom factor.

D.2 Results

For the first three conditions, users exhibited two strategies to identify the various letters. When no information beyond that of the detail view was available, users consistently zoomed out to obtain a wide angle view, then panned and tilted the camera to center the letter, before zooming in again. This *zoom-out* strategy, represented by the solid line in the space-scale diagram [27] of Figure 40a, requires over three camera operations, on average, to identify each letter. When an alert message was added, indicating the display on which the new letter appears, users tended to change their strategy. Knowing the approximate location of the desired monitor from past experience gathered during the study, users often tried to find this monitor by repeatedly panning and tilting the camera, as shown by the solid line in Figure 40b. This strategy is quite similar to searching for an object in a familiar room, while in the dark. Because users cannot accurately select a desired position with the *pan-tilt* strategy, this method often requires more operations than the *zoom-out* strategy. The same *pan-tilt* strategy was used when the global view was provided, but not linked to the detail view. For the remaining three conditions, users were able to identify the letters with only a single camera operation.

Figure 41 and Figure 42 present the results of our user study, indicating the average number of camera operations users required to identify each letter, as well as the average completion time with 95% confidence error bars, with each of the six experimental conditions.

Analysis of variance (ANOVA) showed that both number of operations and trial completion times were significantly affected by the experimental conditions. For number of operations, $F(5, 30)=55.2$, $p<0.001$. For completion time, $F(5, 30)=40.1$, $p<0.001$.

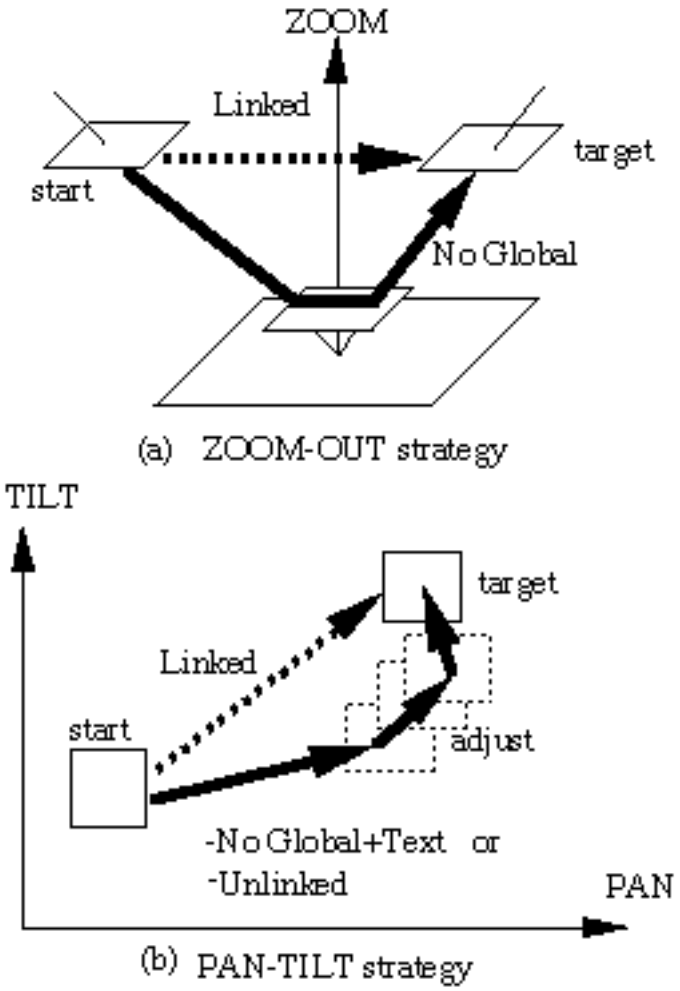

FIGURE 40 Space-scale diagram of camera movement. The solid arrows indicate the users' strategy typically adopted without linkage available, while the dashed arrows indicate the strategy taken when the global and detail views were linked.

As measured by number of operations (Table 7), Fisher's protected LSD posthoc analyses showed that all linked conditions were significantly different from the Unlinked and NoGlobal conditions ($p<0.05$). However, there is no significant difference among linked conditions. The difference between Unlinked and NoGlobal, as well as Unlinked and NoGlobal+Text is also insignificant.

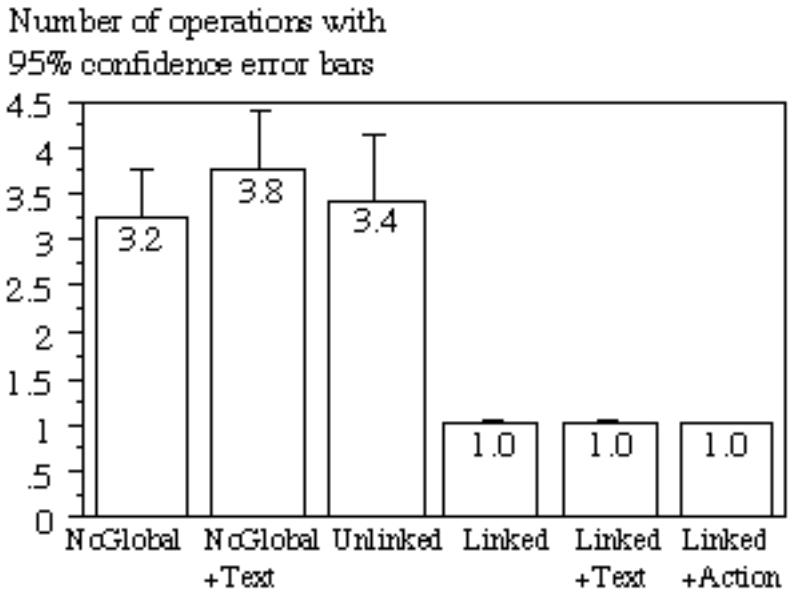

FIGURE 41 Means of number of operations in each experimental condition.

As measured by completion times (Table 8), Fisher's protected LSD posthoc analyses showed that all conditions were significantly different from each other ($p<0.05$), except Linked+Action vs. Linked+Text condition ($p=0.64$) and NoGlobal vs. Unlinked ($p=0.66$).

D.2.1 Importance of Linkage

These results confirmed our hypothesis that linkage between views is very important. When the two views were linked, navigation in the remote environment via selection in the global view was effortless. Any desired (visible) target could be selected directly with a single camera operation, as indicated by the dashed lines of Figure 40 (see also Figure 41). In this case, the previous indirect strategies of *zoom-out* and *pan-tilt*, which require almost twice as much time as direct selection, were never used. Users expressed their opinion that the direct selection mechanism was more natural than the indirect methods. Indeed, all linked conditions were significantly better than the unlinked one in terms of both number of operations and trial completion time.

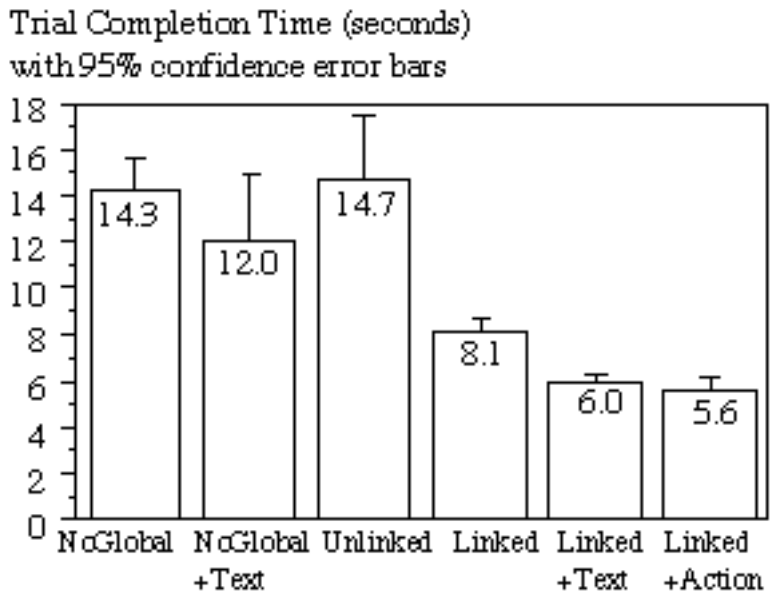

FIGURE 42 Means of completion time in each experimental condition.

Further user feedback was also highly informative. Some commented that the *detail frame* was useful as an indication of direction of camera motion. Furthermore, when the two views were not linked, users had to remain conscious of their current position in order to reach the desired view. This was a result of spatial discontinuities [29]. Linkage between the two views reduced the effect of these discontinuities, because a user action on one view has a direct effect on the other.

D.2.2 Importance of Sensory Information

The time improvement from linked views to linked views with a text alert ($p<0.05$, see Table 8 indicates the added value of sensory information. As most users explained, the alert allowed them to reduce the size of the visual search area. Users also appreciated the audio feedback of a *beep*, provided simultaneously with an alert message, indicating that a new letter was about to appear.

We note that sensory information may have compensated for the low update rate (approximately 1-2 frames/s in our present implementation) of the global view. In many

Condition	Vs.	Diff.	Crit. diff.	P-Value	
Linked + Action	Linked+Text	.008	.528	.9757	
	Linked	.008	.528	.9758	
	NoGlobal+Text	2.753	.528	.0001	S
	NoGlobal	2.226	.528	.0001	S
	Unlinked	2.416	.528	.0001	S
Linked+Text	Linked	4.2E-5	.528	.9999	
	NoGlobal+Text	2.746	.528	.0001	S
	NoGlobal	2.218	.528	.0001	S
	Unlinked	2.408	.528	.0001	S
Linked	NoGlobal+Text	2.746	.528	.0001	S
	NoGlobal	2.218	.528	.0001	S
	Unlinked	2.409	.528	.0001	S
NoGlobal+ Text	NoGlobal	.528	.528	.0500	S
	Unlinked	.337	.528	.2022	
NoGlobal	Unlinked	.191	.528	.4660	

TABLE 7 Posthoc analysis of six experimental conditions: number of operations. The rows marked with an 'S' indicate that these conditions were significantly different at the level 0.05.

instances, the indication of various alerts preceded the appearance of a new letter on the global view by one second or more. This enabled users to begin their navigation toward the desired monitor before the letter was actually visible.

Although the differences in time and number of operations between Linked+Text and Linked+Action were not statistically significant, users indicated that the graphic alerts were more useful than text messages. The graphic alerts completely specify the relevant visual regions, as opposed to text alerts, which require the user to read and then perform a search. Many users simply did not read the text alerts, preferring instead to watch only the graphics display.

Condition	Vs.	Diff.	Crit. diff.	P-Value	
Linked + Action	Linked+Text	.440	1.872	.6350	
	Linked	2.554	1.872	.0092	S
	NoGlobal+Text	6.423	1.872	.0001	S
	NoGlobal	8.762	1.872	.0001	S
	Unlinked	9.172	1.872	.0001	S
Linked+Text	Linked	2.115	1.872	.0282	S
	NoGlobal+Text	5.983	1.872	.0001	S
	NoGlobal	8.323	1.872	.0001	S
	Unlinked	8.732	1.872	.0001	S
Linked	NoGlobal+Text	3.868	1.872	.0002	S
	NoGlobal	6.208	1.872	.0001	S
	Unlinked	6.617	1.872	.0001	S
NoGlobal+ Text	NoGlobal	2.34	1.872	.0160	S
	Unlinked	2.749	1.872	.0054	S
NoGlobal	Unlinked	.409	1.872	.6585	

TABLE 8 Posthoc analysis of six experimental conditions: time (seconds). The rows marked with an ‘S’ indicate that these conditions were significantly different at the level 0.05.

Appendix E The World-Wide Media Space

The Americans have need of the telephone, but we do not. We have plenty of messenger boys.

SIR WILLIAM PREECE,
CHIEF ENGINEER OF THE
BRITISH POST OFFICE, 1876

With increasingly international economic, scientific and social activity, we see a growing demand for computer services that enable and facilitate world-wide human-human communication. In the near future, we predict that the telephone will no longer suffice as a means to provide such communication. Instead, multimedia-capable audio-visual conferencing systems will become more widespread and as a result, access to each other's electronic spaces will be required.

While media spaces [8][17][29] have been explored extensively as a means of supporting such communication, these are typically confined to limited areas such as local analog networks, or special high-bandwidth telephone lines.[1] As a result, media spaces do not have the necessary coverage to make them general communication tools.

In contrast, computer networks such as the Internet do have this coverage, as is seen with World Wide Web (WWW). These networks now face the challenge of supporting

1. While audio/video modems (codec) permit videoconferencing over great distances, these devices are often prohibitively expensive, and many are not compatible with others.

synchronous audio and video communication. Several tools exist, but they impose user constraints that are unnatural in face-to-face communication. In the physical world we can walk around a room, change our view as desired and move to different rooms to visit with other people. As we do so, our senses are aware of our surroundings. Conventional tools, however, do not even provide electronic, or “video space” users with a selection of views from a remote site. Furthermore, the sense of activity and status of the remote site is nonexistent.

Our goal was to overcome these limitations on the electronic user as much as possible, and in the process, to make the media space globally accessible, hence, our World-Wide Media Space (WMS). Previous work has addressed the problems of navigation [29][86] and access to environment information [30][86]. However, these approaches have so far been limited to analog media spaces. With WMS, we offer a solution to a far wider domain, namely, the Internet. While true bidirectional communication will only be possible when other nodes become similarly equipped, we have tried to provide the motivation and carry out the initial effort using this approach.

To expand the media space beyond its current boundaries, we make use of the Internet and a collection of server processes. Our implementation employs the *server push* technique to provide a dynamic image and live, streaming audio, over the WWW.

E.1 Navigation

In order to allow video space users mobility similar to that of physical space users, we provide the former with access to multiple camera views. However, experiments by Gaver [29] point out that this approach risks introducing spatial discontinuities as users switch between different views. Furthermore, because communication over the Internet is often limited in bandwidth, this problem can be exacerbated by the time required for users to select a desired view from the available cameras.

Our solution to these problems is to display an active floor plan, as shown in Figure 43, that indicates the locations of all devices, such as video cameras, digital white board, VCR, and document camera, as well as relevant physical objects such as desks and doorways. The floor plan indicates which devices can be selected and the view each will provide by the use of thick lines and bold letters. For example, when a user clicks on the

icon of a camera or document camera, our servers begin transmitting the selected view as a continuously refreshed image. When a user selects the VCR icon, the VCR begins playing a tape. This technique allows video space users to *walk around* and *explore* a remote site.

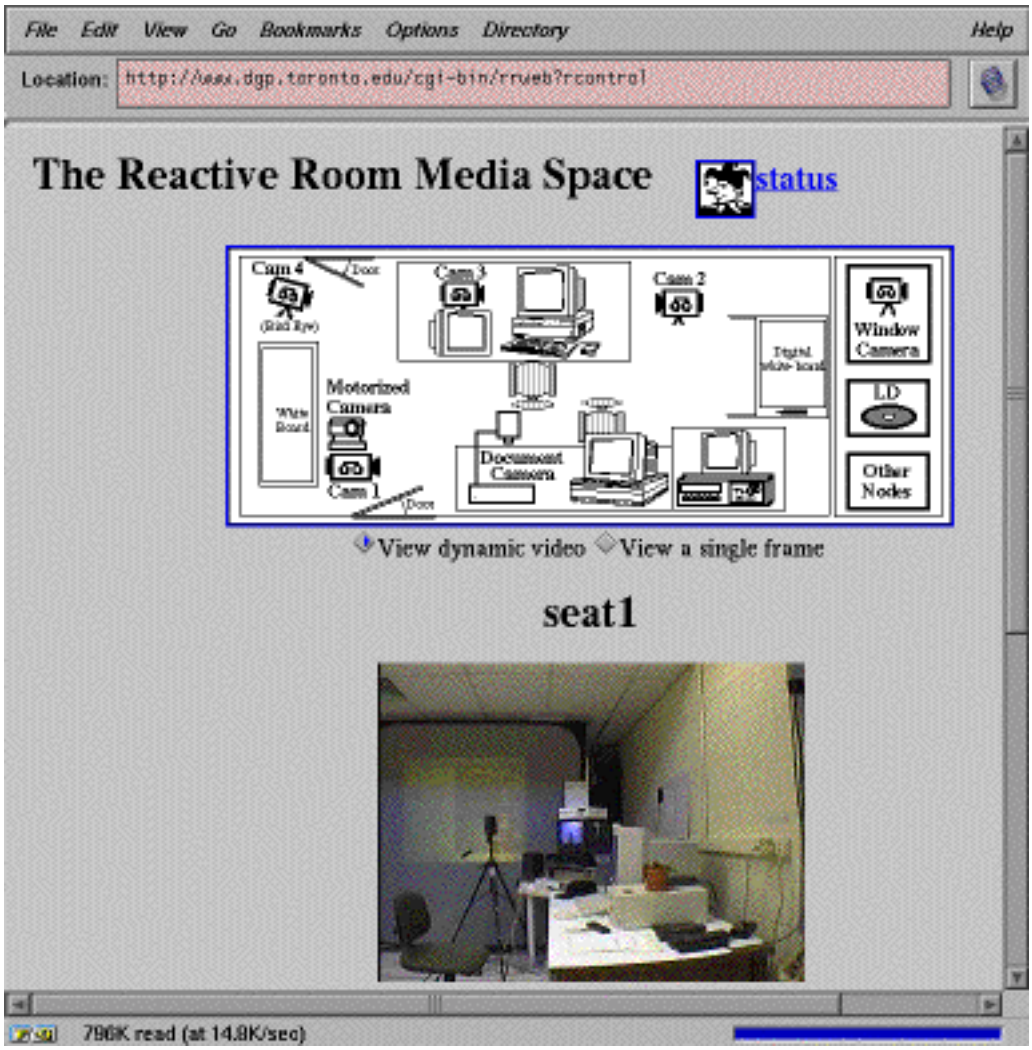

FIGURE 43 Web browser view of the WMS.

To provide mobility beyond the confines of a single room, the WMS also allows users to select an individual from a list of members of the University of Toronto Media Space. If the selected person has allowed remote access, our server automatically establishes a connection through analog lines, warns the individual that they are being observed, and provides their audio and video to the remote user.

E.2 Sensor Information

Similar to the approach taken in the Extra Eyes system [86], we utilize low bandwidth sensor information to provide users with the status of our media space. including the presence of people, the time the room was last entered or exited, and the activity associated with various devices in the room such as the document camera and VCR. This information, as displayed in Figure 44, is useful for feedback. For example, if someone wants to look at Bill's office but he is not in, the system can tell the user what time he left. This information is provided using only a few bytes, as opposed to the high bandwidth demands of live video.

FIGURE 44 Display of the room status.

E.3 Security

Because the WMS is globally accessible, a security system is in place. This allows anyone to observe our media space and engage in limited interaction (e.g. play a video tape), but prevents unauthorized machines from accessing other individuals in the media space.

Authorization is presently performed by checking the IP address of the Web client. However, password verification could also be added as an extra level of security.

E.4 Extensions

With the increased versatility offered by Java Applets running on the client machine, we extended the WMS to incorporate a realistic Web-based implementation of the Extra Eyes system, pictured in Figure 45. The main advantage is that user input can be processed without requiring the time-consuming refresh for a new Web page. Thus, users can select new views without the temporal discontinuity inherent in the initial server-push system. In addition, the sensory surrogate, described in Section 5.3.2, brings relevant activity in our media space to the user's attention through pop-up alert boxes, as illustrated in Figure 45. Java also supports simultaneous streams of audio and video, thereby enabling the creation of a true communications tool.

E.5 Evaluation

Our Web-based media space breaks the physical barrier of distance and allows world-wide access from any Internet-connected location. The WMS went on-line in June 1995, and was visited by approximately 1400 people in the successive six months, over 10% of them returning for future visits. Of these accesses, over 150 were from outside of North America. For some users, visiting our media space has become part of their weekly routine.

Ultimately, our intent is for this model to be used for true bidirectional communication. Realizing this goal entails the construction of similar nodes or servers at other locations. To assist in this endeavor, we have made public a library of Web software that facilitates the implementation of a WWW-based audio and video media space node.[2] In the meantime, we believe that the World-Wide Media Space has been a highly successful effort to put our environment on the World Wide Web, allowing visitors to interact with and navigate independently through our media space.

---

2. The URL is `http://www.dgp.toronto.edu/~rroom`.

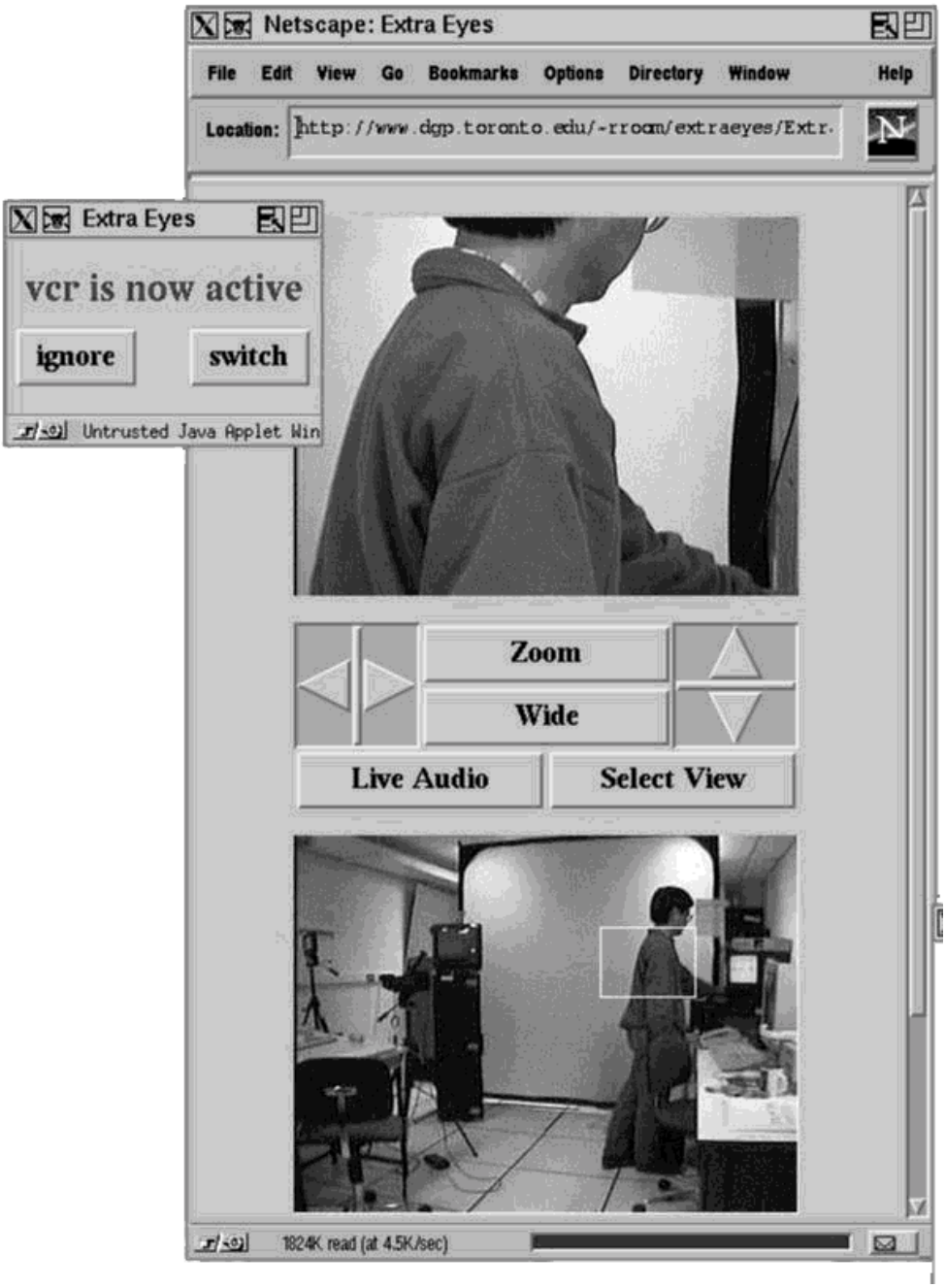

FIGURE 45 The Java-based Extra Eyes system provides the Web browser with two live, linked video streams, in addition to the option of audio. The user can select a desired region to view in detail by dragging a marquis over the area. The sensory surrogate also provides Web users with notification of important events, along with a hotkey to switch to the relevant view.

References

1. ADCOM Inc. *iRoom Technical Specification.* Product Information, 1996.

2. AMX Corporation. *Advanced Remote Control Systems*. Product Information, 1993.

3. Barlow, H.B. and Mollon, J.D. *The Senses*. Cambridge Texts in the Physiological Sciences. Cambridge University Texts, 1982.

4. Bly, S., Harrison, S. and Irwin, S. Media Spaces: Bringing people together in a video, audio and computing environment. *Communications of the ACM*, 36,1, (Jan. 1993), 28-47.

5. Bernstein, N. *The coordination and regulation of movements.* Oxford, Pergamon, 1967.

6. Bradner, B. and Mankin, A. *The Recommendation for the IP Next Generation Protocol.* RFC 1752, January 1995.

7. Brooks, R. A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, RA-2, (April 1986), 14-23.

8. Buxton, W. Telepresence: Integrating Shared Task and Person Spaces. In *Proceedings of Graphics Interface '92*, (May, Vancouver BC). Canadian Human-Computer Communications Society, 1992, pp. 123-129.

9. Buxton, W. Integrating the Periphery and Context: A New Model of Telematics. In *Proceedings of Graphics Interface '95*, (May, Quebec City, PQ). Canadian Human-Computer Communications Society, 1995, pp. 239-246.

10. Buxton, W. Living in Augmented Reality: Ubiquitous Media and Reactive Environments. In *Video Mediated Communication*, K. Finn, A. Sellen and S. Wilber, Eds., Erlbaum, Hillsdale, NJ, in press.

11. Buxton, W. and Moran, T.P. EuroPARC's Integrated Interactive Intermedia Facility (iiif): Early Experience. In *Proceedings of the IFIP WG 8.4 Conference on Multi-user Interfaces and Applications* (Heraklion, Crete). Elsevier Science Publishers B.V. North-Holland, 1990, pp. 11-34.

12. Card, S.K., Moran, T.P. and Newell, A. *The Psychology of Human Computer Interaction*. Lawrence Erlbaum Associates, New Jersey, 1983.

13. Clarke, A.C. Hazards of Prophecy: The Failure of Imagination. In *Profiles of the future; an inquiry into the limits of the possible*. Harper and Row, New York, 1962.

14. Colby, C.L., Gattass, R., Olson, C.R. and Gross, C.G. Topographical Organization of Cortical Afferents to Extrastriate Visual Area PO in the Macaque: A Dual Tracer Study. *Journal of Comparative Neurology*, 269 (1988), 392-413.

15. Conn, A.P. Time Affordances: The Time Factor in Diagnostic Usability Heuristics. In *Proceedings of Human Factors in Computing Systems CHI '95*, (May, Denver, CO). ACM Press, New York, 1995, pp. 178-185.

16. Connell, J. H. *Minimalist Mobile Robotics: A Colony-style Architecture for an Artificial Creature*. Perspectives in AI. Academic Press Inc., 1990.

17. Cool, C., Fish, R.S., Kraut, R.E. and Lowery, C.M. Interactive Design of Video Communication Systems. In *Proceedings of CSCW '92*, (October, Toronto, ON). ACM Press, 1992, pp. 25-32.

18. Cooperstock, J. and Milios, E. Self-supervised Learning for Docking and Target Reaching. *Journal of Robotics and Autonomous Systems*, 11, (Nov. 1993), 243-260.

19. Cooperstock, J. and Kotospoulos, S. Why Use a Fishing Line When You Have a Net? An Adaptive Multicast Data Distribution Protocol, in *Proceedings of USENIX '96 Technical Conferenc*e, (Jan., San Diego, CA). Usenix Press, 1996, pp. 343-352.

20. Cooperstock, J., Tanikoshi, K., Beirne, G., Narine, T., and Buxton, W. Evolution of a Reactive Environment. In *Proceedings of Human Factors in Computing Systems CHI '95*, (May 7-11, Denver, CO). ACM Press, New York, 1995, pp. 170-177.

21. Cooperstock, J., Tanikoshi, K. and Buxton, W. Turning your Video Monitor into a Virtual Window. In *Proceedings of IEEE Pacific Rim Conference on Communications, Computers, Visualization and Signal Processing*, (May, Victoria, BC), 1995.

22. Cypher, A., Ed. *Watch What I Do - Programming by Demonstration*. The MIT Press, Cambridge, MA, 1993.

23. Dourish, P. and Bly, S. Portholes: Supporting Awareness in a Distributed Work Group. In *Proceedings of Human Factors in Computing Systems CHI '92*, (May, Monterey, CA). ACM Press, New York, 1992, pp. 541-547.

24. Elrod, S., Bruce, R., Gold, R., Goldberg, D., Halasz, F., Janssen, W., Lee, D., McCall, K., Pedersen, E., Pier, K., Tang, J. and Welch, B. Liveboard: A large interactive display supporting group meetings, presentations and remote collaboration. In *Proceedings of Human Factors in Computing Systems CHI '92*, (May, Monterey, CA). ACM Press, New York, 1992, pp. 599-607.

25. Elrod, S., Hall, G., Costanza, R., Dixon, M. and Des Rivieres, J. Responsive Office Environments. *Communications of the ACM*, 36,7, (July 1993), 84-85.

26. Ferguson, I. A. Integrating Models and Behaviors in Autonomous Agents: Some Lessons Learned on Action Control. In *Proceedings of AAAI Spring Symposium on Lessons Learned from Implemented Software Architectures for Physical Agents*, (March, Palo Alto, CA). 1995.

27. Furnas, G. and Bederson, B. Space-Scale Diagrams: Understanding Multiscale Interfaces. In *Proceedings of Human Factors in Computing Systems CHI '95*, (May, Denver, CO). ACM Press, New York, 1995, pp. 234-241.

28. Gaver, W. Realizing A Video Environment: EuroPARC's RAVE System. In *Proceedings of Human Factors in Computing Systems CHI '92*, (May, Monterey, CA). ACM Press, New York, 1992, pp. 27-35.

29. Gaver, W., Sellen, A., Heath, C. and Luff, P. One is not Enough: Multiple Views in a Media Space. In *Proceedings of INTERCHI '93*, (April). ACM Press, 1993, pp. 335-341.

30. Gaver, W., Smets, G., and Overbeeke, C. A Virtual Window on Media Space. In *Proceedings of Human Factors in Computing Systems CHI '95*, (May, Denver, CO). ACM Press, New York, 1995, pp. 257-264.

31. Gibson, J. J. *The Ecological Approach to Visual Perception*. Houghton Mifflin Company, Boston, 1979.

32. Grusser, O. and Landis, T. Visual Agnosias and Other Disturbances of Visual Perception and Cognition. *Vision and Visual Dysfunction*, 12, (1991), 240-247.

33. Gujar, A. Daya, S., Nowicki, R., Wang, D. Cooperstock, J., Tanikoshi, K., and Buxton, W. Talking Your Way Around a Conference: A speech interface for remote equipment control. In *Proceedings of CASCON '95*, (November 7-9, Toronto, ON). 1995, p. 289.

34. Hallett, P. Primary and Secondary Saccades to Goals Defined by Instructions. *Vision Research*, 18, (1978), 1279-1296.

35. Heath, C., Luff, P. and Sellen, A. Reconsidering the Virtual Workplace: Flexible Support for Collaborative Activity. In *Proceedings of ECSCW '95*, (Sept., Stockholm, Sweden). 1995.

36. Hinden, R.M. IP Next Generation Overview. *Communications of the ACM*, 39,6, (June 1996), 61-71.

37. Hunke, M. and Waibel, A. Face Locating and Tracking for Human-Computer Interaction. In *Proceedings of the 28th Asilomar Conference on Signals, Systems and Computers*, (Nov., Monterey, CA). 1994.

38. *IEEE-1394-1995 Standard for a High Performance Serial Bus*, Institute of Electrical and Electronics Engineers, 1995.

39. Ishii, H., Kobayashi, M., and Grudin, J. Integration of Interpersonal Space and Shared Workspace: Clearboard Design and Experiments. *ACM Transactions on Information Systems (TOIS)*, 11,4, (Oct. 1993), 349-375.

40. Kay, A. Computer software. *Scientific American* 251,3, (Sept. 1984), 52-59.

41. Kellogg, W., Carroll, J., and Richards, J. Making Reality a Cyberspace. In *Cyberspace: First Steps*, M. Benedikt, Ed. MIT Press, 1991.

42. Kelly, P. H., Katkere, A., Kuramura, D. Y., Moezzi, S., Chatterjee, S., and Jain, R. An Architecture for Multiple Perspective Interactive Video. In *Proceedings of ACM Multimedia*. ACM Press, New York, 1993, pp. 201-212.

43. Krüger, M. Environmental technology: Making the real world virtual. *Communications of the ACM*, 36,7 (Jul. 1993), 36-51.

44. Kuzuoka, H., Kosuge, T. and Tanaka, M. GestureCam: A Video Commutation System for Sympathetic Remote Collaboration,. In *Proceedings of CSCW '94,* (Oct., Chapel Hill, NC). ACM Press, 1994, pp. 35-43.

45. Leveson, N.G. *Safeware: System Safety and the Computer Age*. Addison-Wesley, Reading, MA, 1995.

46. Lieberman H. An Example Based Environment for Beginning Programmers. In *Artificial Intelligence and Education*, R. Lawler and M. Yazdani, Eds. Ablex, Norwood, NJ, 1987.

47. Lombardo, T.G. TMI: An insider's viewpoint. *IEEE Spectrum*, (May 1980), 52-55.

48. Mackay, W, Velay, G, Carter, K, Ma, C & Pagani, D. Augmenting reality: Adding computational dimensions to paper. *Communications of the ACM*, 36,7 (Jul. 1993), 96-97.

49. Maes, P. Agents that Reduce Work and Information Overload. *Communications of the ACM*, 37,7 (Jul. 1994), 31-40.

50. Mantei, M., Baecker, R., Sellen, A., Buxton, W., Milligan, T. and Wellman, B. Experiences in the use of a media space. In *Proceedings of Human Factors in Computing Systems CHI '91*. ACM Press, New York, 1991, pp. 203-208. Reprinted in *Groupware: Software for Computer-Supported Collaborative Work*, D. Marca and G. Bock, Eds. IEEE Computer Society Press, Los Alamitos, CA, 1992.

51. Marcus, A. The Future of Advanced User Interfaces in Product Design. In *Proceedings of Tenth TRON Project International Symposium*, (Tokyo, Japan). IEEE Computer Society, 1992, pp. 14-21.

52. Martin, G.L. The utility of speech input in user-computer interfaces. *International Journal of Man/Machine Studies*, 30, (1989), 355-375.

53. Michalski, R.S., Carbonell, J.G., and Mitchell, T.M., Eds., *Machine Learning: An Artificial Intelligence Approach*. TIOGA Publishing Company, Palo Alto, CA, 1983.

54. Mo, D.H. and Witten, I.H. Learning text editing tasks from examples: a procedural approach. *Behaviour & Information Technology 11*,1, (1992), 32-45.

55. Myers, B. *Creating User Interfaces by Demonstration*. Academic Press, San Diego, 1988.

56. Negroponte, N. *The Architecture Machine; Towards a more Human Environment*. MIT Press, Cambridge, Mass., 1970.

57. Negroponte, N. *Being Digital*. Knopf, New York, 1996.

58. Neumann, P. G. *Computer Related Risks*. ACM Press, New York, 1995.

59. Norman, D.A. *The Psychology of Everyday Things*. Basic Books, New York, 1988.

60. Overbeeke, C. and Stratmann, M. *Space through movement*. Unpublished doctoral thesis, TU Delft, The Netherlands, 1988.

61. Payne, S.J. and Green, T.R.G. Task-action grammars: a model of mental representation of task languages. *Human-Computer Interaction*, 2,2, (1986), 93-133.

62. Pentland, A. Smart Rooms. *Scientific American* 274,4, (April 1986), 68-76.

63. Perrow, C. The President's Commission and the Normal Accident. In *Accident at Three Mile Island: The Human Dimensions*, D.L. Sills, C.P. Wolf, and V.B. Shelanski, Eds. Westivew Press, Boulder, CO, 1982.

64. Postel, J. Transmission Control Protocol. *RFC 793*, USC/Information Science Institute, September 1981.

65. Raskin, J. Down With GUIs! In *WIRED* 1.6, Wired Ventures, December 1994.

66. Rasmussen, J. The Human Data Processor as a System Component: Bits and Pieces of a Model. Technical Report RISØ-M-1722, Danish Atomic Energy Commission, Roskilde, Denmark, 1974.

67. Rasmussen, J. The role of hierarchical knowledge representation in decision making and system management. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15, (1985), 234-243.

68. Riesenbach, R. The Ontario Telepresence Project. In *Conference Companion of Human Factors in Computing Systems CHI '94*, (April, Boston, MA). ACM Press, New York, 1994, pp. 173-174.

69. Sakamura, K. Human Interface with Computers in Everyday Life. In *Proceedings of Ninth TRON Project International Symposium*, (Tokyo, Japan). IEEE Computer Society, 1992, pp. 2-12.

70. Sakamura, K. Infrastructures for an Age of Computerized Environments. In *Proceedings of Tenth TRON Project International Symposium*, (Tokyo, Japan). IEEE Computer Society, 1993, pp. 2-14.

71. Schmandt, C. Phoneshell: the Telephone as Computer Terminal. In *Proceedings of ACM Multimedia*. ACM Press, 1993, pp. 373-382.

72. Stefik, M., Foster, G., Bobrow, D., Kahn, K., Lanning, S. and Suchman, L. Beyond the chalkboard: Computer support for collaboration and problem solving in meetings. *Communications of the ACM*, 30,1, (July 1987), 32-47.

73. Thelen, E. Motor Development. *American Psychologist*, 50, (1995), 79-95.

74. Torrance, M. Advances in Human-Computer Interaction: The Intelligent Room. Unpublished manuscript.

75. Tsotsos, J. K., Culhane, S. M., Wai, W., Y., K., Lai, Y., Davis, N., and Nuflo, F. Modeling Visual Attention via Selective Tuning. *Artificial Intelligence*, 78,1-2, (1995), 507-547.

76. Turvey, M.T. Coordination. *American Psychologist*, 45, (1990), 938-953.

77. Vicente, K. and Rasmussen J. A Theoretical Framework for Ecological Interface Design. Technical Report RISØ-M-2736, Risø National Laboratory, Denmark, August 1988.

78. 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.

79. Want, R., Hopper, A., Falcao, V., and Gibbons, J. The Active Badge Location System. *ACM Transactions on Information Systems*, 10,1, (1992), 91-102.

80. Want, R., Schilit, B.N., Adams, N.I., Gold, R., Petersen, K., Goldberg, D., Ellis, J.R., and Weiser, M. An Overview of the PARCTAB Ubiquitous Computing Experiment. *IEEE Personal Communications*, (Dec. 1995), 28-43.

81. Weiser, M. The Computer for the 21st Century. *Scientific American*, 265,3 (1991), 94-104.

82. Weiser, M. Some Computer Science Issues in Ubiquitous Computing. *Communications of the ACM*, 36,7, (July 1993), 75-83.

83. Wellner, P. Interacting with Paper on the DigitalDesk. *Communications of the ACM*, 36,7, (July 1993), 87-97.

84. Wellner, P., Mackay, W. and Gold, R. Computer-Augmented Environments: Back to the Real World. *Communications of the ACM*, 36,7, (July 1993), 24-26.

85. Whittaker, S. and Cummings, R., Foveating Saccades, *Vision Research* 30,9, (1990), 1363-1366.

86. 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*, (April 13-18, Vancouver, BC). ACM Press, New York, 1996, pp. 50-57.