Design Principles for Intelligent Environments

Michael H. Coen

MIT Artificial Intelligence Lab
545 Technology Square
Cambridge, MA 02139
mhcoen@ai.mit.edu

Abstract

This paper describes design criteria for creating highly embedded, interactive spaces that we call Intelligent Environments. The motivation for building these systems is to bring computation into the real, physical world to support what is traditionally considered non-computational activity. We describe an existing prototype space, known as the Intelligent Room, which was created to experiment with different forms of natural, multimodal human-computer interaction. We discuss design decisions encountered while creating the Intelligent Room and how the experiences gained during its use have shaped the creation of its successor.

1. Introduction

This paper describes design criteria for creating highly embedded, interactive spaces that we call *Intelligent Environments* (IEs). The motivation for building IEs is to bring computation into the real, physical world. The goal is to allow computers to participate in activities that have never previously involved computation and to allow people to interact with computational systems the way they would with other people: via gesture, voice, movement, and context.

We describe an existing prototype space, known as the *Intelligent Room*, which is a research platform for exploring the design of intelligent environments. The Intelligent Room was created to experiment with different forms of natural, multimodal human-computer interaction (HCI) during what is traditionally considered non-computational activity. It is equipped with numerous computer vision, speech and gesture recognition systems that connect it to what its inhabitants are doing and saying.

Our primary concern here is how IEs should be designed and created. Intelligent environments, like traditional multimodal user interfaces, are integrations of methods and systems from a wide array of subdisciplines in the Artificial Intelligence (AI) community. Selecting the modal components of an IE requires a careful strategic approach because of the *a priori* assumption that the IE is actually going to be embedded in the real-world. In particular, there is a need for the use of synergy (Cohen [4]) to allow imperfect modalities to reinforce and support each other.

We discuss below the design of our laboratory's Intelligent Room and how experiences gained during its use have shaped the creation of its successor. Given the increasingly widespread interest in highly interactive, computational environments (Bobick et al. [3]), (Coen [6,7,8]), (Cooperstock et al. [10]), (Lucente et al. [17]), we hope these experiences will prove useful to other IE designers and implementers in the AI community.

Some of the earliest work in this area has been done wholly outside the AI community. This is primarily due to the perception that AI has little to offer in the way of robust, ready for the real world systems. We contend that Intelligent Environments not only would benefit from AI subdisciplines ranging from knowledge representation to computer vision, but they would be severely limited without them.

Outline

Section 2 describes some sample interactions with and applications of the Intelligent Room. These range from an intelligent command post to a reactive living room. Comparison to other HCI paradigms, such as ubiquitous computing, and other embedded computational environments is contained in section 3. Section 4 presents the Intelligent Room's physical infrastructure. Sections 5 and 6 detail the Intelligent Room's visual and spoken language modalities. We document the rationales that influenced our approach, system limitations, and solutions we are pursuing in the development of the next generation Intelligent Room currently under construction in our laboratory.

2. Room Interactions

Our approach with the Intelligent Room has been to create a platform for HCI research that connects with real-world phenomena through several computer vision and speech

This material is based upon work supported by the Advanced Research Projects Agency of the Department of Defense under contract number F30602—94—C—0204, monitored through Rome Laboratory. Additional support was provided by the Mitsubishi Electronic Research Laboratories.

recognition systems. These allow the room to watch where people are moving, under certain circumstances where they are pointing, and to listen to a fairly wide variety of spoken language utterances.

The Intelligent Room supports a variety of application domains. One of these is a command center for planning hurricane disaster relief in the Caribbean. This makes use of two interactive projected displays that respond to both finger pointing and laser pointing gestures. A sample interaction with the disaster relief center is:

User: *"Computer, <pause> stay awake."*
[The room will now listen for utterances without requiring they be prefaced by the word *Computer*.]
User: *"Show me the Virgin Islands."*
Room: "*I'm a showing the map right next to you.*" [Room shows map on video display *closest* to the user.]
User: [now points at St. Thomas.] *"Zoom in. How far away is Hurricane Marilyn?"*
Room: *"The distance between Hurricane Marilyn and the city of Charlotte Amalie located in St. Thomas is 145 miles."*
User: *"Where's the nearest disaster field office?"*
[Room highlights them on the map.]
Room: *"The St. Thomas disaster field office is located one mile outside of Charlotte Amalie. Michael, there is a new weather forecast available. Do you want to see it?"*
User: *"Yes, show me the satellite image."*

We are currently developing a next generation of the Intelligent Room, called *Hal* (after the computer in the movie, *2001: A space odyssey)*. Hal is furnished like a combination home/office and supports a wider range of activities than the original Intelligent Room. A scenario that currently runs within Hal is:

> I walk into Hal and lie down on the sofa after shutting the door. Hal sees this, dims the lights, closes the curtains, and then puts on Mozart softly in the background. Hal then asks, *"Michael, what time would you like to get up?"*

The goal of implementing these types of scenarios is to explore and help define what an intelligent environment should be, what sensory capabilities it needs, and to determine what roles such environments could potentially play in our lives. In the process, these scenarios provide insight into both how AI systems can participate in the real world and directions for further research in the subdisciplines whose systems contribute to the creation of intelligent environments.

3. Motivation

Intelligent environments are spaces in which computation is seamlessly used to enhance ordinary activity. One of the driving forces behind the emerging interest in highly interactive environments is to make computers not only genuinely user-friendly but also essentially invisible to the user. The user-interface primitives of these systems are not menus, mice and windows but gesture, speech, affect, and context. Their applications are not spreadsheets and word processing but intelligent rooms and personal assistants.

Intelligent environments are both embedded and multimodal and thereby allow people to interact with them in natural ways. By being embedded, we mean these systems use cameras for eyes, microphones for ears, and ever-increasingly a wide-range of sophisticated sensing technologies to connect with real-world phenomena. Computer vision and speech recognition/understanding technologies can then allow these systems to become fluent in natural forms of human communication. People speak, gesture, and move around when they communicate. For example, by embedding user-interfaces this way, the fact that people tend to point at what they are speaking about is no longer meaningless from a computational viewpoint and we can build systems that make use of this information. In some sense, rather than make computer-interfaces for people, we want to make people-interfaces for computers.

Coupled with their natural interfaces is the expectation that these systems are not only highly interactive (i.e. they talk back when spoken to) but also that they are useful for ordinary activities. They should enable tasks historically outside the normal range of human-computer interaction by connecting computers to phenomena (such as someone walking into a room) that have traditionally been outside the purview of contemporary user-interfaces.

Why this isn't Ubiquitous Computing

Intelligent environments require a highly embedded computational infrastructure; they need many connections with the real world in order to participate in it. However, this does not imply that computation need be everywhere in the environment nor that people must directly interact with any kind of computational device. Our approach is to advocate minimal hardware modifications and "decorations" (e.g., cameras and microphones) in ordinary spaces to enable the types of interactions in which we are interested. Rather than use the computer-everywhere model of ubiquitous computing – where for example, chairs have pressure sensors that can register people sitting in them or people wear infrared-emitting badges so they can be located in a building – we want to enable unencumbered interaction with non-augmented, non-computational objects (like chairs) and to do so without requiring that people attach high-tech gadgetry to their bodies (as opposed to the approach in [24,25]).

AI-based approaches have much to offer these environments. For example, although a pressure sensor on a chair may be able to register that someone has sat down, it is unlikely to provide other information about that person, e.g., her identity. Visual data from a single camera can provide far more information than simple sensing technologies. This includes the person's identity, position, gaze direction, facial expression, gesture, and activity ([13, 25,17,30,12]). While there has yet to be a coherent system that unifies all of these capabilities, many prototypes are currently under development. Furthermore, enhancing the capabilities of a computer vision system often requires modifying only the software algorithms that process incoming images and not the room's sensory components. Also, because the room senses at a distance, objects, in particular people and furniture, do not need to be physically augmented and/or wired for the room to become aware of them.

Other related work

The DigitalDesk project (Wellner [26], Newman et al. [19]) was an early and influential system that had a bird's eye view of a desktop through an overhead video camera. It recognized and responded to predetermined hand gestures made by users while interacting with real paper documents on the surface of a desk. The Intelligent Room has a desktop environment directly motivated by the DigitalDesk, which recognizes a wider range of complex hand gestures (Dang [11]).

Other substantial efforts towards highly interactive environments include an automated teleconferencing office (Cooperstock et al. [10]) and an immersive fictional theater (Bobick et al. [3]). Each of these projects makes use of embedded computation to enable unusual human-computer interactions, e.g., vision-based person tracking. However their modal processing is extraordinarily specific to their applications, and the applicability of such carefully tuned systems to other domains is unclear. The Classroom 2000 project (Abowd et al. [1]) is an educational environment that automatically creates records linking simultaneous streams of information, e.g. what the teacher is saying while a student is writing down her notes on a digital pad.

Mozer ([18]) describes a house that automatically controls basic residential comfort systems, such as heating and ventilation, by learning patterns in its occupants behavior.

Related user-interface work such as Cohen et al. [5] uses multimodal interface technology to facilitate human interaction with a preexisting distributed simulator. In doing so, it provides a novel user-interface to a complex software system, but it is one that requires tying down the user to a particular computer and a specific application. We are interested in creating new environments that support never before conceived of applications – applications that historically have not involved computation.

4. The Intelligent Room

The Intelligent Room occupies a 27'x37' room in our laboratory. Approximately half of this space is laid out like an ordinary conference room, with a large table surrounded by chairs. (See Figure 1.) This section has two bright, overhead LCD projectors in addition to several

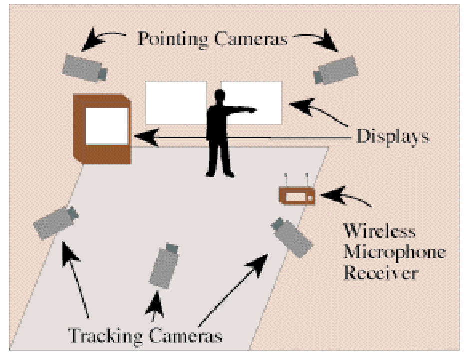

Figure 1 – A skeletal view of the conference area in the Intelligent Room

video displays. There is also an array of computer controlled video equipment which is discussed below. Mounted at various places in the conference area are twelve video cameras, which are used by computer vision systems.

Separated from the conference area by a small partition and occupying the rest of the room are most of the workstations that perform the room's computation. The section of the room is not interactive, but having it adjacent to the interactive conference area simplifies wiring, implementation and debugging.

The Intelligent Room contains an array of computer controlled devices. These include steerable video cameras, VCRs, LCD projectors, lights, curtains, video/SVGA multiplexers, an audio stereo system, and a scrollable LCD sign. The room's lighting is controlled through several serially interfaced X-10 systems. Many of the room's other devices have serial ports that provide both low-level control and status information, e.g., our VCRs can report their present position on a videotape to give us random access to video clips. The room can also generate infrared remote control signals to access consumer electronics items (namely, objects that don't have serial ports).

Room Controller

When the Intelligent Room was in the early stages of it design and construction, the most challenging research problems appeared to be developing its computer vision and speech recognition/understanding systems. What was not obvious is that interconnecting all of the rooms many subsystems and coordinating the flows of information among the room components was a non-trivial problem. Developing a software architecture that allowed the room to run in real-time and cope with vagaries of its real-world interactions emerged to be one of the room's chief research problems.

What emerged from an iterative development process is a modular system of software agents known collectively as the *Scatterbrain* (described in detail in Coen [6]). The Scatterbrain currently consists of approximately 50 distinct, intercommunicating software agents that run on ten different networked workstations. These agents' primary task is to connect various components of the room (e.g., tracking and speech recognition systems) to each other and to internal and external stores of information (e.g., a person locator or an information retrieval system). Essentially, the Scatterbrain agents are intelligent *computational glue* for interconnecting all of the room's components and moving information among them.

5. Room Vision Systems

Person Tracking

The Intelligent Room can track up to four people moving in the conference area of the room at up to 15Hz. The room's person tracking system (DeBonet [13]) uses two wall-mounted cameras, each approximately 8' from the ground. (A debugging window from the system showing the view from one of the cameras is shown in Figure 2.)

We initially decided that incorporating a tracking system in the Intelligent Room was essential for a number of reasons. It gives the room the ability to know where and how many people are inside it, including when people enter or exit. The room is able to determine what objects people are next to, so for example, it can show data on a video display someone is near. A person's location in the room also provides information about what she is doing. For example, someone moving near a video display while others are seated around the conference table might indicate she is giving a presentation.

The tracking data are useful for supplying information to other systems in the room including, to our surprise, our speech understanding system. It was clear from the start that tracking could disambiguate other room modalities, for example, by providing a foveal area for gesture recognition. However, its use in providing contextual information to the room's speech recognizer is a revealing example of how one modality can be used to help overcome the weaknesses of another. In this case, where people are in the room can sometimes provide information about what they are likely to say (see section 6).

The tracking system works via background segmentation and does 3D reconstruction through a neural network. The output image from each camera is analyzed by a program that labels and identifies a bounding box around each occupant in the room. This information is then sent through a coordination program that synchronizes the findings from the individual cameras and combines their output using a neural network to recover a 3D position for each room occupant. People are differentiated in the system using color histograms of their clothing, which the room builds when they first come inside. Because the room's configuration is fairly static and the cameras used for tracking are stationary, the tracking system can build a model of the room's relatively slowly changing background to compare with incoming images.

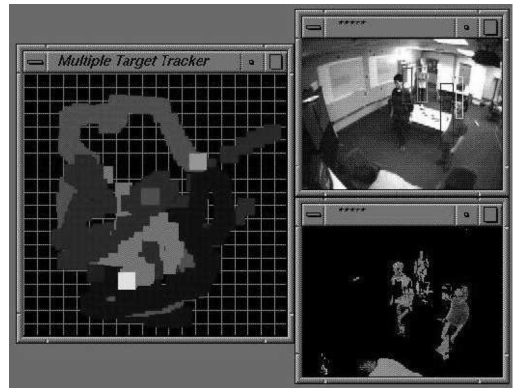

Figure 2 - Tracking System Debug Window

The tracking subsystem also controls three steerable cameras. These can be used to follow individuals as they move about the room or to select optimal views of people given their position and previous knowledge of room geometry, e.g. where people likely face when standing in particular areas of the room.

This approach differs from the overhead tracking system described in Bobick et al. [3]. Their domain had 27' high ceilings, for which it is quite reasonable to look for people from a single camera bird's eye perspective. Rooms with ordinary height ceilings do not make this possible, so a stereo vision system seems necessary for performing background segmentation.

Pointing

The Intelligent Room's two overhead LCD video projectors display next to each other on one of the room's walls. Each can display SVGA output from one of the room's workstations or composite signals from any of the room's video sources, e.g., a VCR. These projected displays support both finger and laser pointing interactions. For example, the room can track the finger of a person who is pointing within four inches of the wall where images are displayed. Alternatively, the person can use a laser pointer to interact with the display from a distance. Both of these pointing systems also allow displayed screen objects to be selected (i.e. clicked) or moved (i.e. dragged).

Additionally, the pointing systems allow people to treat the displays like virtual whiteboards. The room can draw a visible trail on top of a displayed image that follows the continuous path of a motile pointing gesture. This allows people to overlay handwritten text and drawings on top of whatever information the room is displaying. These can then be automatically recalled at a later date, for example, when the room shows this information again.

The finger pointing system uses two cameras mounted parallel to the wall on either side of the displays. It makes use of only three scan lines from each camera image to explore the region closest to the wall's surface. The laser pointing system uses a camera roughly orthogonal to the plane of the display wall to locate the laser's distinctive signature on the wall's surface. These systems run at approximately 15-20Hz, depending on the precise type of interaction, and provide resolution per display ranging from approximately 640x480 for laser pointing to 160x120 for finger pointing. Although the pointing systems are sufficiently responsive for discrete pointing and dragging events, handwriting recognition using the above mentioned drawing feature does not seem practical with out at least doubling the sampling frequency.

Interactive Table

Through a ceiling mounted camera, the room can detect hand-pointing gestures and newly placed documents on the surface of the conference table. The gesture recognition system has been used to support a wide variety of functions (described in Dang [11]). We found, however, that making gestures over the surface of a table was not a particularly natural form of interaction and required extensive practice to master. As has been widely observed in the graphical user interface community, we found that increased novelty in an interface does not necessarily lead to increased utility. This is even more pertinent in domains like the Intelligent Room, which stress natural modes of interaction.

One useful application of this system, however, allows people to place Post-It™ notes on the surface of the table and assign to them particular functions, such as dimming the lights or announcing the current time. Touching a given note then evokes its assigned behavior from the Intelligent Room. As a mnemonic, the function of each note can be handwritten upon it, giving the table the feeling of a virtual, very non-standard control panel. The room is oblivious to any written information on these notes, as long as it doesn't interfere with the color segmentation that allows individual notes to be recognized.

Issues

Our person tracking system uses a neural network to perform 3D reconstruction. The tracking network is trained by laying a masking tape coordinate system on the room's floor and then training the network by having a person stand at each intersection of the axes. (The grid was roughly 10x20.) Although non-network approaches to 3D reconstruction are possible, such as directly calculating the reverse projective transformation, they would all likely require a user-intensive preliminary learning period to determine the transformation between room and image space. Thus, installing our tracking system is labor intensive and requires some familiarity with how it operates.

Another difficulty with this approach is that the system is enormously sensitive to any deviation from its training conditions. For example, if one of the cameras is moved by so much as 1cm, the tracking system fails to function. Although automatic recalibration might be possible by using natural or artificial environmental fiducials, in practice these are either difficult to detect or highly intrusive when added to the environment. Thus, cameras being moved or rotated requires retraining the neural network, something a non-technical user would never want to do.

It is not accidental that so much computer vision research around the world is performed in rooms without windows. Computer visions systems, particularly ones that rely on background segmentation, can be extraordinarily sensitive to environmental lighting conditions. For example, early in the Intelligent Room's development, ambient light coming through a newly opened door could completely disrupt all of the room's vision systems. While this was an early design flaw, in general it can be extremely difficult to compensate for changing lighting conditions.

Shadows are also particularly difficult phenomena to cope with and we took great pains to avoid them. We disabled some of the room's overhead fluorescent lighting and used upward pointing halogen floor lamps instead. Additionally, we selected a fairly dark colored carpet, which is better at shadow masking. The tracking system also used a color correction mechanism for shadow elimination. However, a static color correction scheme was only partially useful because the tracking cameras were affected by the dynamic lighting of the projected video displays.

Solutions

Our research agenda for computer vision systems for Hal has changed drastically from the approach used in the Intelligent Room. Rather than incorporating the state of the art in visually based interactions, we have become far more interested in robust vision systems that require little calibration and are essentially self-training.

We have enabled the room's vision systems to reinforce one another. For example, our multi-person tracker may temporarily lose people when they occlude one another or blend into the background. One way to help offset this is to have the finger pointing system provide information useful for tracking. Someone finger pointing at the projected display must be standing somewhere near that position on the room's floor. By knowing where someone is pointing, the tracker can focus its attention on that section of the room. Conversely, the tracking system allows the room to identify the person who is pointing at the wall. By determining which tracked person is closest to the pointed at position, the room can distinguish among its inhabitants during finger pointing gestures.

Various devices in the room can also interact with its vision systems. The software agents that control the room's drapes and electrical lights notify the vision systems before they do anything that might affect the room's ambient lighting. This allows each vision system either to recalibrate or to deactivate itself until conditions favorable to its correct operation are restored and also avoids incorrect event recognition due to luminosity changes.

Although dynamic person tracking seemed essential during the design of the Intelligent Room, it became clear in retrospect that the vast majority of the tracking system's output is thrown away. Few applications need or can make use of real-time trajectory information for the room's occupants. Rather, what is particularly important is to know where someone is when she stops moving (i.e. next to or sitting on some piece of furniture) or when she has crossed a particular threshold (i.e. the room's doorway).

It is far easier and computationally less demanding to build systems that provide these kind of relatively slowly changing data without resorting to real-time occupant tracking. They look for people at rest in places where they are expected to be found, such as sitting on a couch or standing by a display, or for people crossing through a narrow, well-defined region such as a doorway.

We have implemented and experimented with several such systems, which we call *static person locators* and *threshold detectors.* These include a template-based *couch detector*, which locates people either sitting or lying down on a chair or sofa. This system is easily trained and quite robust. We have also implemented a dedicated *doorway tracker* for distinctly determining when someone enters or leaves the room, and thereby it also keeps track of how many people are currently present. Both of these systems are algorithmically quite simple and far less sensitive to environmental variations than our initial tracking system. They have proved quite robust, and their initial detection accuracy in varying light conditions and over a wide range of individuals is over 90%.

We are also creating generic chair locators using a ceiling mounted vision system. Our assumption is that occlusion of a chair provides evidence that someone is sitting in it, and this person can be located using prior knowledge of the chair's position. This system will use low dimension eigenspaces for approximating object manifolds under varying pose and lighting conditions (Stauffer [21]). The advantage to this approach is that the system need not be given in advance an explicit model of the chairs it will be locating. The system can construct object manifolds itself by having a user rotate any new types of chairs she brings inside.

6. Speech Interactions

Among the earliest decisions regarding the Intelligent Room was that it should support spoken language interactions. In particular, we were interested in allowing these interactions to be unimodal – i.e. ones that did not tie the user to a video display to verify or correct utterances recognized by the system nor require a keyboard for selecting among possible alternative utterances. We also wanted to avoid simply creating a keyboard replacement that allowed speech commands to substitute for actions ordinarily performed by typing or mouse clicking. Finally, we wanted to allow interaction with multiple applications simultaneously and thus not have interactions that monopolized the user. In the process, we have tried to allow the Intelligent Room to engage in dialogs with users to gather information, correct misunderstandings, and enhance recognition accuracy.

People in the Intelligent Room wear wireless lapel microphones that transmit to the speech understanding system described below. By default, the room ignores the spoken utterances of its inhabitants, which are generally directed to other people within the room. This state is known as "the room being asleep."[1] To obtain the room's attention, a user stops speaking for a moment and then says the word *"Computer"* out loud. The room immediately responds with an audible, quiet chirp from an overhead speaker to indicate it is paying attention. The user then has a two second window in which to begin speaking to the room. If the room is unable to recognize any utterances starting within that period, it silently goes back to sleep until explicitly addressed again. However, if what the user says is recognized, the room responds with an audible click and then under most circumstances it returns to sleep. This hands- and eyes-free style of interaction coupled with audio feedback allows a user to ignore the room's *computational presence* until she explicitly needs to communicate with it. There is no need to do anything other than preface spoken utterances with the cue *Computer* to enable verbal interaction. Thus, a user can interact with the room easily, regardless of her proximity to a keyboard or monitor.

The Intelligent Room is capable of addressing users via the Festival Speech Synthesis System (Black et al. [2]). Utterances spoken by the room are also displayed on a scrollable LCD sign in case a user was unable to understand what was said. The room uses its speech capability for a variety of purposes that include conducting dialogs with users and getting its occupant's attention without resorting to use of a visual display. Sometimes, the room chooses to respond vocally to a question because its video displays are occupied by what it considers high priority information. For example, if a user asks, *"What's the weather forecast for New York City?"* the room can simply read the forecast to the user, rather than put up a weather map containing forecast information if its displays are occupied.

For processing spoken utterances, we use both the Summit (Zue et al. [27]) and DragonDictate speech recognition systems in parallel. Each of these has different strengths and used together they have fairly robust performance. The Summit system recognizes continuous speech and is particularly adept at handling syntactic variability during recognition. By entering bigram models, it is fairly straightforward to build topically narrow but syntactically unconstrained sets of recognizable utterances. Bigram models, however, make it quite difficult to exclude particular statements from being erroneously recognized by the system and require that we heavily post process

1 The room's vision systems continue to function and respond to users even when it is not listening for verbal input.

Summit's output. This is performed primarily by the START natural-language information retrieval system (Katz [15]).

DragonDictate is a commercially available system primarily used for discrete speech dictation, meaning that users must pause after each word. This, when coupled with its relatively low word accuracy, would be an intolerable speech interface to the room. However, DragonDictate also supports explicit construction of continuous speech, context-free recognition grammars. Via a special library, it also provides complete control over low-level aspects of its behavior to external applications, which makes it ideal for incorporating into other systems.

Issues

There is a tradeoff between making the room's recognition grammars sufficiently large so that people can express themselves somewhat freely versus making the grammars small enough so that the system runs with high accuracy and in real-time. We tuned DragonDictate's performance by creating sets of specialized grammars for different room contexts and having the room's software agent controller dynamically activate different subsets of grammars depending on the context of the activity in the Intelligent Room (Coen et al. [9]). This allows us to overcome the combinatorial increase in parsing time due to incorporating natural syntactic variability in the recognition grammars.

Instead of keeping a single enormous recognition grammar active, the room keeps subsets of small grammars active in parallel, given what it currently expects to hear. The key assumptions here are that certain types of utterances are only likely to be said under particular circumstances, and these are circumstances among which the room is capable of distinguishing. These may be related to where someone is spatially, the history of her previous interactions (i.e. what room applications are active), how she is gesturing, what devices in the room are doing, etc. At the simplest level, this can range from the implausibility of someone saying *"stop the video,"* when none is playing, to more complex dependencies, such as the meaninglessness of someone asking *"What's the weather there?"* if no geographic entity has somehow been brought to the room's attention.

We have generalized the notion of linguistic context to include the state of and goings on in the room and have put this contextual knowledge into the room's software agents rather than its linguistic data structures. For example, if the room starts showing a video clip, the agent that controls the showing of videos activates the grammars that involve VCR operation. When the clip stops, these grammars are in turn deactivated. More interesting cues can involve the location of someone inside the room. The fact that someone has moved near an interactive displayed map causes the room to pay attention to spoken utterances involving geographic information. Thus, information from the room's other systems can help overcome computational limitations in the room's speech recognition and understanding systems.

Verbal interactions can also be extremely useful for dealing with the room's other modalities. They can be used to gather information about what the room is observing, to modify internal representations of its state, or to correct a perceptual error. It is also of enormous benefit to be able to verbally interact with the room's vision systems while developing or debugging them, because it is generally impossible to manually interact with them at a workstation while remaining in the cameras' fields of view.

7. Conclusion

Our experience with the Intelligent Room has led us to reevaluate many of our initial assumptions about how a highly interactive environment should be designed. Intelligent environments need to be more than rough assemblages of previously existing systems. In particular, careful selection and communication among modalities can lead to synergistic reinforcement and overall, a more reliable system. The modalities must also be carefully selected in order to make the environment easy to install, maintain, and use under a wide range of environmental conditions.

Systems that dynamically adjust to the room's activity, such as our speech understanding system, and systems that can train themselves and avoid extensive manual calibration, are essential to an IE's success. We hope the issues addressed in this paper will both stimulate further discussion and prove useful to other designers of intelligent environments.

8. Acknowledgements

Development of the Intelligent Room has involved the efforts of many people. This includes Professors Tomas Lozano-Perez, Rodney Brooks, and Lynn Stein. Graduate students who have been or are involved in the room include Mark Torrance, Jeremy De Bonet, Chris Stauffer, Sajit Rao, Darren Phi Bang Dang, JP Mellor, Gideon Stein, Michael Coen, Josh Kramer, Brenton Phillips, Mike Wessler, and Luke Weisman. Postdocs associated with the project include Kazuyoshi Inoue and Polly Pook. Many undergraduates have been or are currently working on it. These include Kavita Thomas, Nimrod Warshawsky, Owen Ozier, Marion Groh, Joanna Yun, James Clark, Victor Su, Sidney Chang, Hau Hwang, Jeremy Lilley, Dan McGuire, Shishir Mehrotra, Peter Ree, and Alice Yang.

References

1. Abowd, G., Atkeson, C., Feinstein, A., Hmelo, C., Kooper, R., Long, S., Sawhney, N., and Tani, M. Teach and Learning a Multimedia Authoring: The Classroom 2000 project. *Proceedings of the ACM Multimedia'96 Conference*. 1996.

2. Black, A. and Taylor, P. Festival Speech Synthesis System: system documentation (1.1.1) Human Communication Research Centre Technical Report HCRC/TR-83. University of Edinburgh. 1997.
3. Bobick, A.; Intille, S.; Davis, J.; Baird, F.; Pinhanez, C.; Campbell, L.; Ivanov, Y.; Schütte, A.; and Wilson, A. Design Decisions for Interactive Environments: Evaluating the KidsRoom. *Proceedings of* the *1998 AAAI Spring Symposium on Intelligent Environments*. AAAI TR SS-98-02. 1998.
4. Cohen, P., "The role of natural language in a multimodal interface," *Proceedings of User Interface Software Technology (UIST'92) Conference*, Academic Press, Monterey, California, 1992.
5. Cohen, P., Chen, L., Clow, J., Johnston, M., McGee, D., Pittman, K., and Smith, I. Quickset: A multimodal interface for distributed interactive simulation, *Proceedings of the UIST'96 Demonstration Session*, Seattle. 1996.
6. Coen, M. Building Brains for Rooms: Designing Distributed Software Agents. *Proceedings of the Ninth Conference on Innovative Applications of Artificial Intelligence. (IAAI97).* Providence, R.I. 1997. http://www.ai.mit.edu/people/mhcoen/brain.ps
7. Coen, M. Towards Interactive Environments: The Intelligent Room (a short paper). *Proceedings of the 1997 Conference on Human Computer Interaction (HCI'97).* Bristol, U.K. 1997.
8. Coen, M. (ed.) *Proceedings of the 1998 AAAI Spring Symposium on Intelligent Environments.* AAAI TR SS-98-02. 1998.
9. Coen, M; Thomas, K; Weisman, L; Groh, M; and Yee, A. A Natural Language Modality for an Embedded Multimodal Environment. Forthcoming.
10. Cooperstock, J; Fels, S.; Buxton, W. and Smith, K. Environments: Throwing Away Your Keyboard and Mouse. *Commmunications of the ACM*. 1997.
11. Dang, D. Template Based Gesture Recognition. SM Thesis. Massachusetts Institute of Technology. 1996.
12. Davis, J. and Bobick, A. The representation and recognition of action using temporal templates. *Proceedings Computer Vision and Pattern Recognition (CVPR'97).* pp.928-934. 1997.
13. DeBonet, J. Multiple Room Occupant Location and Identification.1996. http://www.ai.mit.edu/people/jsd/jsd.doit/Research/HCI/Tracking_public
14. Druin, A.; and Perlin, K. Immersive Environments: a physical approach to the computer interface. *Proceedings of the Conference on Human Factors in Computer Systems (CHI'94),* pages 325-326, 1994.
15. Katz, B. Using English for Indexing and Retrieving. In *Artificial Intelligence at MIT: Expanding Frontiers*. Winston, P.; and Shellard, S. (editors). MIT Press, Cambridge, MA. Volume 1. 1990.
16. Lien, J., Zlochower, A., Cohn, J., Li, C., and Kanade, T. Automatically Recognizing Facial Expressions in the Spatio-Temporal Domain. *Proceedings of the Workshop on Perceptual User Interfaces (PUI'97).* Alberta, Canada. pp.94-97. 1997.
17. Lucente, M.; Zwart, G.; George, A. Visualization Space: A Testbed for Deviceless Multimodal User Interface. *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments*. AAAI TR SS-98-02. 1998.
18. Mozer, M. The Neural Network House: An Environment that Adapts to its Inhabitants. *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments*. AAAI TR SS-98-02. 1998.
19. Newman, W. and Wellner, P. A Desk Supporting Computer-based interaction with paper. *Proceedings of the Conference on Human Factors in Computing Systems (CHI'92).* p587-592. 1992.
20. Saund, E. Example Line Drawing Analysis for the ZombieBoard Diagrammatic User Interface. http://www.parc.xerox.com/spl/members/saund/lda-example/lda-example.html. 1996.
21. Stauffer, C. Adaptive Manifolds for Object Classification. 1996. http://www.ai.mit.edu/people/stauffer/Projects/Manifold/
22. Stiefelhagen, R., Yang, J., and Waibel, A. Tracking Eyes and Monitoring Eye Gaze. *Proceedings of the Workshop on Perceptual User Interfaces (PUI'97).* Alberta, Canada. pp.98-100. 1997.
23. Torrance, M. Advances in Human-Computer Interaction: The Intelligent Room. *Working Notes of the CHI 95 Research Symposium*, May 6-7, Denver, Colorado. 1995.
24. Want, R.; Schilit, B.; Adams, N.; Gold, R.; Petersen, K.; Goldberg, D.; Ellis, J.; and Weiser, M. The ParcTab Ubiquitous Computing Experiment. Xerox Parc technical report.
25. Weiser, M. The Computer for the 21st Century. *Scientific American*. pp.94-100, September, 1991.
26. Wellner, P. The DigitalDesk Calculator: Tangible Manipulation on a Desk Top Display, *Proceedings of UIST'91*. pp.27-33. 1991.
27. Zue, V. Human Computer Interactions Using Language Based Technology. *IEEE International Symposium on Speech, Image Processing & Neural Networks.* Hong Kong. 1994