The Intelligent Room Project

Rodney A. Brooks
MIT Artificial Intelligence Lab
545 Technology Square
Cambridge, MA, 02139, USA

With contributions from: Michael Coen, Darren Dang, Jeremy De Bonet, Joshua Kramer, Tomás Lozano-Pérez, John Mellor, Polly Pook, Chris Stauffer, Lynn Stein, Mark Torrance, and Michael Wessler.

Abstract

At the MIT Artificial Intelligence Laboratory we have been working on technologies for an Intelligent Room. Rather than pull people into the virtual world of the computer we are trying to pull the computer out into the real world of people. To do this we are combining robotics and vision technology with speech understanding systems, and agent based architectures to provide ready at hand computation and information services for people engaged in day to day activities, both on their own and in conjunction with others.

We have built a layered architecture where at the bottom level vision systems track people and identify their activities and gestures, and through word spotting decide whether people in the room are talking to each other or to the room itself. At the next level an agent architecture provides a uniform interface to such specially built systems, and to other off the shelf software, such as web browsers, etc. At the highest level we are able to build application systems that provide occupants of the room with specialized services; examples we have built include systems for command and control situations rooms and as a room for giving presentations.

1 Introduction

Our modern computer interfaces are implicitly centered around the notion of drawing the human into the computer's world. This is true of both the everyday WIMP (window, icon, mouse, pointer) based interfaces, and more immersive VR (virtual reality) based interfaces. In both cases the person has to go to the interface, enter the artificial world (a desktop metaphor for WIMPs, some three dimensional constructed space for VR), and then manipulate objects within that artificial world. The computer itself has no awareness of the person *per se*, no model that it is a person, and no understanding in any deep sense of what it is that people (as opposed to other machines) do.

In the Intelligent Room project at the MIT Artificial Intelligence Laboratory we have inverted the relationship between person and computer. In the Intelligent Room people work together or alone as they would were the computer not present. There are no keyboards, mice, monitors, or virtual reality headsets. Instead the computer is drawn out into the world of people, and forced to operate there, listening to what people say, watching what they do and how they move, keeping track of what is going on, and trying to be helpful when there is something that they can do. In the Intelligent Room the person is in charge; the computer has to play by human rules.

This approach to human computer interface is made possible by using techniques that have been developed over the last thirty years in computer vision, robotics, speech understanding, and natural language processing[1].

1.1 The overall idea

The intent of the Intelligent Room is to make computation ready-at-hand. It should be available without the user of computation having to shift their mode of thinking or interaction with people. Furthermore we reject the idea of building special spaces in which such intelligent interaction can occur. The computation should adapt to the environment and to the people using it.

Our strategy then is to install the Intelligent Room in a normal room by adding only minimal decorations to that space. In particular the decorations we add are camera modules which consist of a fixed wide angle camera and a steerable narrow angle camera, microphones, a video-multiplexer so that any output image can be shipped to any display, electronic control of lighting, etc., cameras looking sheer along walls where people might be pointing to projected displays, and optionally overhead cameras. These decorations are all lightweight in the sense that we do not require

[1] The closest popular image to what we are trying to do is the way computation is used on the bridge of the Enterprise in *Star Trek, The Next Generation*. However the writers for that show did not predict 400 years ahead the utility of computer vision that can be achieved today. (Of course the original Star Trek writers had similar myopiae about technology that has become routine in the home in the last thirty years—the moral is that any technological prediction is doomed to be wrong.)

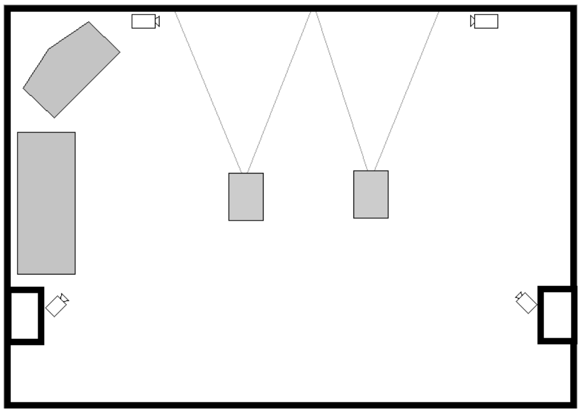

Figure 1: The current Intelligent Room layout has two ceiling mounted projectors as the primary display devices. Cameras look over the whole room from the rear, and other cameras are mounted on the walls to view where people might point at images on the walls.

instrumentation of existing structures as one might given other engineering approaches to the problem; for instance one might require that all occupants of the space wear active badges, and that all chairs, tables, shelves, etc., be instrumented with either transmitters or receivers that communicate with the badges. Nor do we require extensive calibration of the systems we install. We try to make everything automatically self-calibrating as we will see later in the paper.

Figure 1 shows the layout of our first Intelligent Room. We will soon have a second smaller Intelligent Office with a completely different layout. There are multiple cameras looking at the work area of the room. There are two projectors which operate in full lighting conditions as the main output devices. There are two other large displays in the room based on conventional monitors. There are two cameras looking sheer down the wall on which the projectors project. There are microphones in the room, although it still currently works best if users wear individual radio microphones. There are two VCRs under full computer control, and all the lighting in the room is under computer control.

Figure 2 shows the software arrangement for the room. There are three layers of very different sorts of software.

At the lowest level there are perceptual systems which generate real-time descriptions of what is happening in the room. These include systems which track occupants of the room, determine the types of interactions that are going on between people (e.g., shaking hands, sitting, talking, etc.), systems which determine whether and where someone is pointing to one of the projected images on the walls, systems which determine where a laser pointer is being pointed at the walls, multiple systems which interpret what is being said, systems to control the video-multiplexer, systems to control the VCRs, and systems to control the lighting. All these interactions are timestamped so that temporal sequencing can be determined by higher level systems.

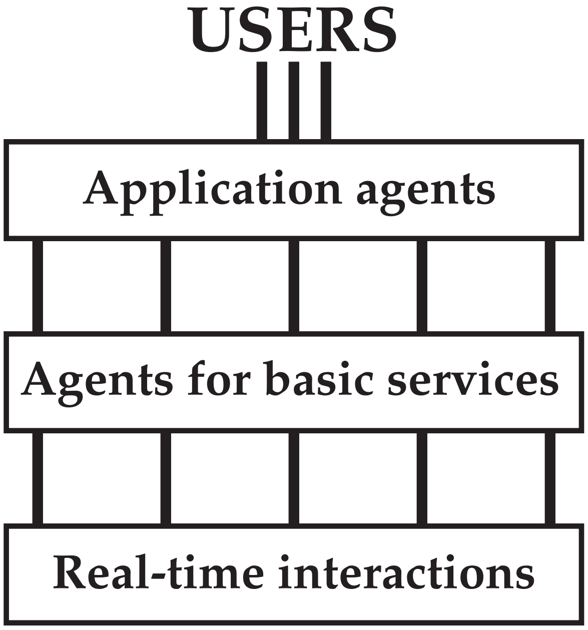

Figure 2: The software to control the room is divided into three conceptual layers. The lowest level layers provide real time interpretations of what is being done and said in the room. The next layer wraps those real-time systems and other legacy and off-the-shelf software packages in a uniform agent interface. At the top level applications are written for particular uses of the room

The second level of figure 2 provides a uniform agent-based interface to everything that is installed in the room, along with all software that might be used from within the room. Thus there are agent interfaces which report on pointing operations, and agent interfaces to off the shelf software such as Netscape. These agents hide any network issues that arise from different parts of the system running on different workstations, and from the different characteristic time frames for interactions with underlying software. They provide uniform interfaces, so that the pointing agent and the Netscape agent, for example, can talk to the same agents in the same terms. Thus it becomes trivial for higher level software to turn the detection of a person pointing with their hand to a highlighted region of text in a displayed web page into a click event for Netscape.

And thus we arrive at the third level of figure 2. On top of this uniform agent interface we are able to build application layers which provide, with fairly limited levels of knowledge, functionality for specific types of uses of the Intelligent Room.

1.2 Scenarios

We will briefly describe two scenarios, both implemented, for using the intelligent room. The first turns it into a command and control center for disaster relief, the second makes it an interactive space for virtual tours of the MIT Artificial Intelligence Laboratory.

The knowledge that the high level agents need for the disaster relief scenario includes knowledge of maps, transport, weather, and how to access relevant web resources in these areas. The idea with disaster relief is that there would be a number of different specialists working together trying to get the latest information on the disaster marshalled together and dispatch appropriate supplies and personnel to the disaster site using whatever transportation means are currently available.

The Intelligent Room tracks all the occupants. If someone says "Computer[2], show me a map of the disaster area" the room picks the closest display device to where the person is currently standing. Then perhaps the person points at a location on the map and says "Computer, how far away is the hurricane from here?". The room resolves the reference for "here" to be the place on the map that was just pointed to, and using its special purpose knowledge of maps is able to verbally respond. "Computer, where is the nearest airport?" is then interpreted to be the nearest airport to the recently pointed at and discussed location. "Computer, what is the weather like?" is likewise interpreted in the appropriate context, and now the room goes out on the web and finds the latest weather report for that precise geographical location and displays it appropriately.

The AI laboratory tour is intended to be used by novice users. The information for the tour is collected directly from existing web pages of lab members[3] rather than putting together a special database. There is also a library of videos that are indexed with natural language descriptions of their contents. A tourist can ask the room about particular topics, e.g., "Computer, tell me about the Cog project." All the words on the web pages have been pre-scanned by a natural language understanding system so that the vocabulary of the speech understanding systems can be updated to understand their contents. The user can ask about links on the web page: "Computer, follow the link to media hype.", or can jump around to other topics that come to mind. The room thus provides a content based surfing interface to the web, not one dictated by how the links were set up by individual page authors. In addition the room is always matching the page contents to its video library, and when there is something appropriate it takes the initiative and offers to play video for the user. Thus the web pages are automatically augmented with other data that the page authors have not considered.

Figure 3: This is the result of observing a person walking around the room for a few minutes. Data from a single monocular camera was used. The full three dimensional structure of the room has been recovered.

2 Real-time Interactions

The visual real-time interaction systems are all based on using motion of people to determine what is happenning in the scene. All the cameras used are relatively low cost with cheap lenses that can cause distortions. The systems are self-calibrating wherever possible.

In order to pick out motion from the background all images from fixed cameras filter out the background by adaptively averaging the pixels so that an image of the background is built up over time. This image is not fully static as lighting may change, or furniture may be moved within the room. The background is then subtracted from the current image and usually this corresponds to where people are as people usually make motions at higher rates than the background adaptation.

2.1 Determining the room structure

Stauffer (1997) built a system which uses the idea of *domain constraints* (Horswill 1993) to produce a three dimensional reconstruction of the room using a single uncalibrated camera, and observing people walking about in the room. The domain constraints are straightforward:

1. The floor of the room is flat.
2. The height of a person is constant as they walk around.

These two constraints have some important consequences. The first thing to notice is that they imply, for a perspective projection camera that is mounted on a wall and tilted slightly downwards, that the height

[2] We now have speech systems continuously monitoring everything all the occupants say, looking for the keyword "Computer". It takes that as the start of an utterance meant for it and tries to parse the following words into a query on a command. Sometimes it wakes up incorrectly, due to mishearing "Computer" or perhaps the occupants are just discussing computers! So we have the room make a very low but audible beep sound when it is switching into listening mode. If that is incorrect, one of the occupants can just say "Go to sleep!", and another type of beep indicates that it has understood and has switched back to word spotting mode.

[3] The MIT AI Laboratory has approximately 208 members.

of the y-coordinate of the top of a person's head is a monotonically increasing function of the distance of the person from the camera.

Given that a person is moving, their image will differ from the background pixels. Any pixels that a person obscures at any given instant must be further from the camera than the person. (Likewise any pixels that obscure the person must be closer than the person, although this information does not need to be used in the current implementation.) So the background pixels can initially each be labelled as an unknown distance from the camera, and as a person walks around and obscures them they can be tagged with a minimum distance from the camera (or at least a monotonic function of a minimum distance), based on finding the top of the motion region, and assuming that that is the top of the person's head. As a person moves around in the room, these minima get pushed further and further away from the camera, but never beyond their true depth.

Figure 3 illustrates the outcome of observing a person in the room for a few minutes. The depth map has been rotated slightly to give the three dimensional structure. It even includes the depth through a window in the room, from observing a person walking along an outside corridor.

2.2 People tracking

We have built a number of people trackers. They all rely on adaptive background differencing to extract people by their motion from the background. The base goal of a person tracker is for it to return the x and y room coordinates of the location of each person in the room. Ideally it should be able to tag individuals so that it returns the coordinates of a person as they move, not just an unordered list of locations of all people in the room.

The simplest person tracker estimates a rectangular box surrounding a single person, then borrowing an idea from Lettvin, Maturana, McCulloch & Pitts (1959) on bug detectors in frogs, finds a best match to a predetermined set of rectangles. The rectangles are predetermined by having a person walk around the room, and record their x and y coordinates at a number of locations. For each location their motion rectangle is saved. When a new motion best matches that rectangle then its associated x and y coordinates are returned by the people tracker.

This simple idea can be made much more robust, and able to handle multiple people by using two cameras at once. Figure 4 shows the results of tracking multiple occupants of the room using this simple idea. The ground tracks of the four people recorded over time are also shown. It is necessary to form an approximate color histogram of each person so that when they cross each others paths, then separate again, it is possible to determine which person is which.

2.3 Determing what people are doing

Such simple systems do have limitations however. Inoue (1996) extended this idea by having finite state models of what could possibly happen in the room. By having temporal coherence enter into the computations much more reliable interpretations can be made

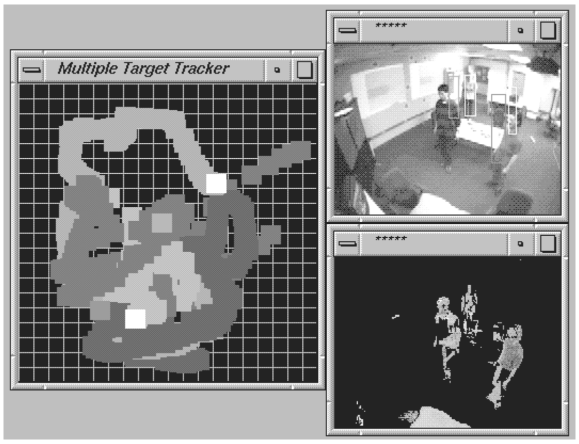

Figure 4: Multiple people can be tracked using "people" detectors based on motion and a set of rectangular cells which give x and y coordinates when that cell is the best match to the location of the person. Using two cameras and "symbolic" stereo increases the accuracy. The upper right image shows the view from the camera with the four activated rectangular cells. The bottom right shows the four segmented people. The left image shows historical tracks of the four people as they have walked around the room—this is in room floor coordinates.

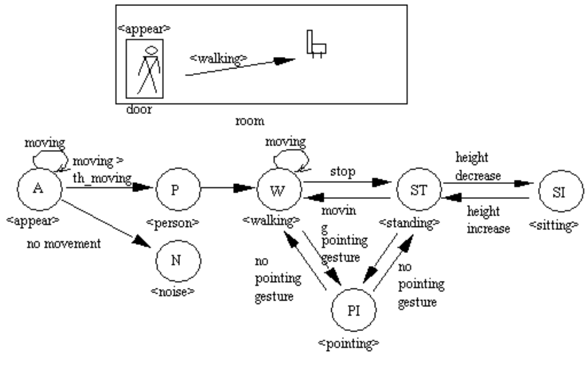

Figure 5: A person can engage in a number of simple activities in a room. These are only certain legal transitions between those activities. This grammar provides constraints on possible interpretations of images, making recognition of what a person is doing much more robust.

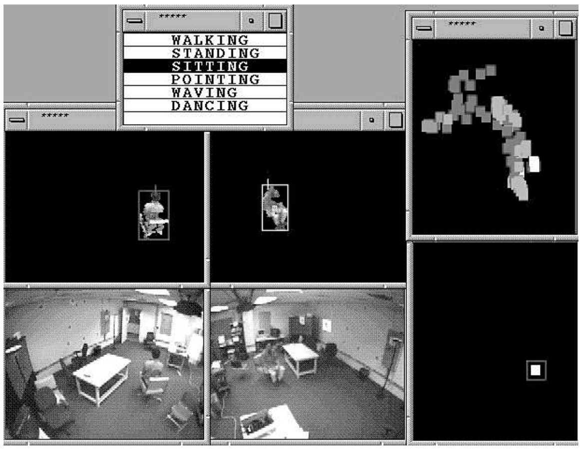

Figure 6: Using the grammar from figure 5 and viewpoints from two cameras it can be determined that a person is sitting down in the room. On the right the person's history is shown along with an accumulation of activities.

than from a single image (or pair of images).

Figure 5 shows a grammar of possible actions of a single person in a room. A person can be engaged in *walking*, *standing*, *sitting*, or *pointing*. The grammar shown enables a process to pick up a person as they first enter the room, and shows all possible transitions. It also allows for the failure of pickups due to noise in the system. Note that this grammar has embedded in it a number of implicit constraints which, as for the other vision algorithms, are domain constraints. This time, rather than being physical domain constraints as in the case of determining the room structure or tracking people, they are social constraints imposed by the ordinary activities that people engage in in business type environments or rooms.

Figure 6 shows an interpretation of a person sitting, using the grammar from figure 5.

The grammar of figure 5 is for a single person. The system incorporates two higher levels of grammars that build on top of these primitive action recognition grammars. At the next level are grammars for two or three people. These are used to recognize activities like *shaking hands*, or *talking*. On top of this are still higher level grammars for group activities like *meeting*, *presentation*, etc.

Figure 7 shows a higher level recognition of two people sitting and talking to each other. The accumulation of sitting activities and the tracks of where people have walked enable the system to infer where chairs and tables are within the room—since these can move from day to day, this *behavior-based recognition* (based on the behavior of people) can be very useful.

2.4 Understanding pointing

There are at least two possible ways people can communicate with other people by pointing to items in projected displays. One is by pointing to an item in a display with their hand, the other is using a laser pointer to indicate some particular area of interest.

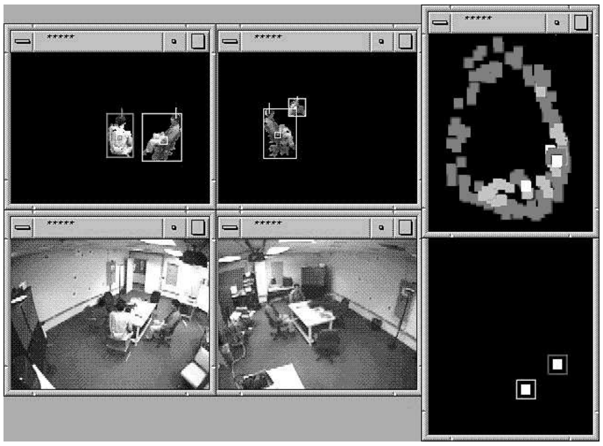

Figure 7: Using a higher level grammar it can be recognized that the two people here are sitting and talking to each other.

Figure 8: Two cameras look sheer along the wall where the images from the projector point. By determing the y coordinate in each camera of an intrusion in a narrow vertical slice of each image it is easy to localize where the person's finger is pointing to with about 2.5cm accuracy. Calibration is achieved by temporarily attaching cardboard "fingers" to the wall at known locations.

In order for the Intelligent Room to understand such behavior we have implemented two different ways of interpreting pointing.

To understand how people point with their hands we have cameras (see figure 1) that look along the surface of the wall where the projectors project. Within each camera image we look for horizontal intrusions (again using differencing from the adapted background) into narrow vertical slices of the image. This is illustrated in figure 8. From the two y coordinates, that are so determined, a lookup table can produce a location of the person's finger to within about 2.5cm. Calibration of this lookup table is achieved by temporarily attaching objects at known locations on the wall.

To determine laser pointing, color processing is done on images of the wall itself from a camera at the rear of the room. It is easy to pick out the bright fixed wavelength of the laser pointer, and again a simple calibration scheme allows for reasonably accurate determination of where the laser is pointing.

For a page of text projected on the wall, localization by either pointing method is sufficient to play the role traditionally supplied by a mouse for clicking around on hypertext. It is not sufficiently accurate to run a drawing program however.

2.5 Enhanced reality

Another vision technique we have used is *enhanced reality*. Here real images are enhanced by extra information; this is in contrast to virtual reality where the whole image is synthesized.

There are two applications for enhanced reality.

The first is in an environment where users are wearing glasses that both let them see the world and lets them see additional registered information. The glasses must have cameras mounted on them looking in the direction that straight ahead eyes would look. Mellor (1995) has built a system which can solve for the viewing position of the camera in real-time. This position can then be used to compute a graphical image to overlay in registration with what the person is seeing. We have used this in the intelligent room to define "virtual X windows". These are X windows hanging at a fixed location in space where text and graphics can be displayed. Wherever the person wearing the glasses is, they see this virtual window at the same fixed location out in the world. Having to wear special glasses is somewhat against our general philosphy, but we believe that in the future all eye-glasses, and perhaps even contact lenses, will have this capability built into them.

The second use of enhanced reality is in intelligent teleconferencing. In this case there is no issue of registering on the viewing direction of a user. Instead, information is intelligently added to the images that are being shipped as part of teleconferencing. The simplest example of this is to label someone with their name; once this has been bootstrapped somehow (in the Intelligent Room we can simply say "My name is Rodney") the person tracker provides the system with enough information to keep the label attached to the appropriate person.

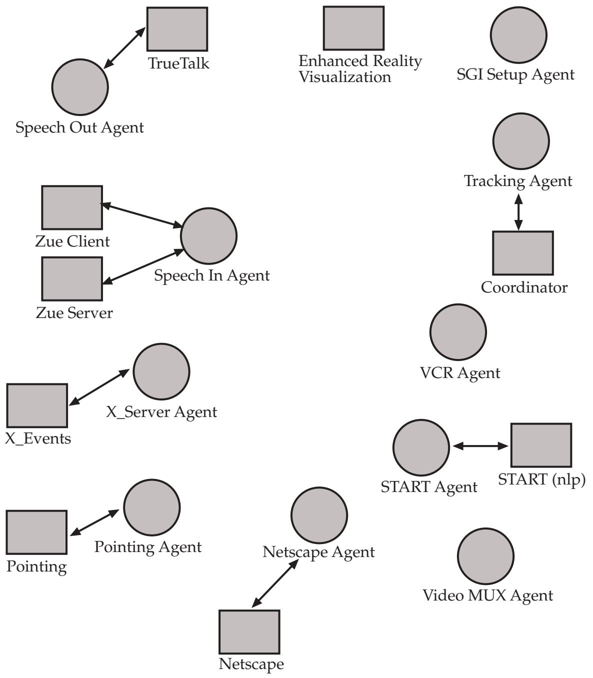

Figure 9: All devices drivers, vision code, speech understanding systems, and off the shelf software are wrapped with agents written in the SodaBot language. This hides all implementation details from higher level application agents.

3 Agent-based Layers

Coen (1994) has developed the *SodaBot* agent language for the Intelligent Room project. SodaBot agents can span multiple workstations and are mobile. They completely separate data transport mechanisms from the agent programmer. This frees the agent programmer to concentrate on what is to be communicated rather than details of how or when it is to be communicated. The upper two layers of figure 2 are implemented in SodaBot.

The second layer is illustrated in more detail in figure 9. This layer is application independent, and wraps agents around all the subsystems built in other languages. These subsystems include three categories:

1. Real-time systems, mostly vision based as described in the previous section, that were built specifically for the Intelligent Room.

2. Other human computer interaction specific systems that we have obtained from elsewhere, such as the speech system of (Zue 1994), or the natural language interface to databases (Katz 1990).

3. Standard off the shelf software systems that have been built with the intention of having more conventional user interfaces, such as Netscape, and the X windows system.

The agent wrappers provide abstract interfaces to all such systems. There is no need for higher-level

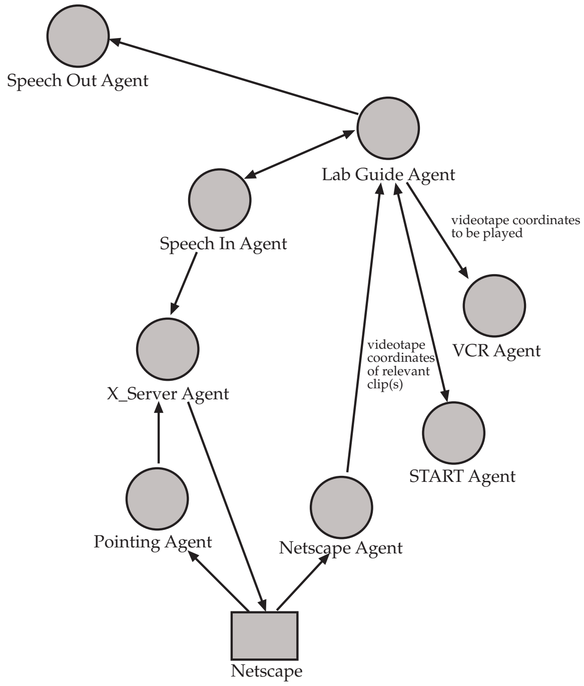

Figure 10: This shows a set of communications that occur between agents for the scenario described in section 3.

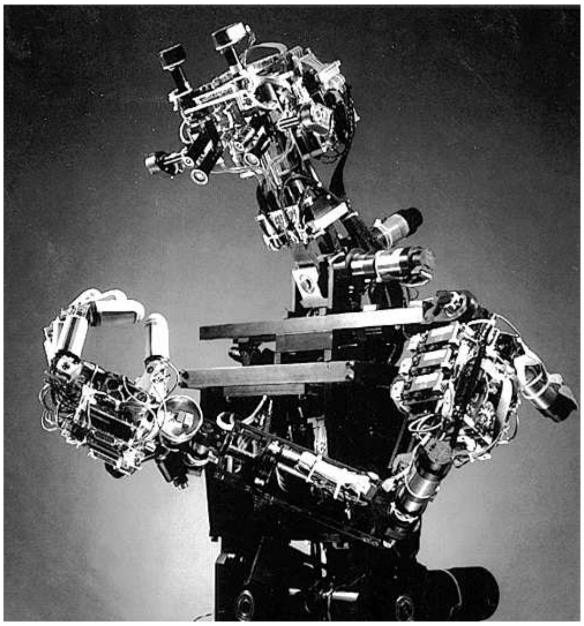

Figure 11: The Cog robot built at the MIT Artificial Intelligence Laboratory.

agents using these subsystems to have to know about any of their internal implementation details. Furthermore there is no need for any agents to know on which machines the lower level systems run, nor details of the communications paths to them. All that is hidden by the wrappers.

Figure 10 shows the flow of information between these agents during the interaction with the *Lab Guide Agent* described in section 1.2. When the person points at an item displayed in Netscape on the wall and says that they want to follow the link, the pointing operation is sent via the X-server agent pack to Netscape, which then notifies the Lab Guide agent, via the Netscape agent. The Lab Guide agent realizes that it has a relevant video clip by querying a database in natural language (since the content was referenced in English on a web page). It gets the SpeechOut agent to inform the person, and then tells the SpeechIn agent some context; it is expecting a "yes/no" response. Upon getting a "yes" response the LabGuide agent tells the VCR to play the tape at the appropriate coordinates.

4 Summary

Current day vision technology is sufficient for realtime user interfaces based on visual observation of people.

Vision and speech can augment each other and provide a more robust and natural interface than can either system alone.

We are not stuck forever with being dragged into the world of the computer in order to interact with it. Instead the tools are at hand to bring the computer out into our world, so that it is a participant in the normal forms of human interaction of which we partake.

No more paper tape. No more punched cards. No more WIMP interfaces.

5 The Future

Intelligent Rooms are one sort of interaction we can have with computers. But there are other possibilities as well. At the MIT Artificial Intelligence Lab we are also working on humanoid robots (Brooks & Stein 1994) that allow for face to face human-like interactions. The first such robot, Cog, is shown in figure 11. At IS Robotics,[4] in its Artificial Creatures division, work proceeds on mass-market toys with fully articulated faces, and complex emotional models for human interaction. Figure 12 illustrates a baby doll that acts very much like a small baby, and requires the care and attention of a child playing with it so that it is adequately fed, rested and stimulated. These lines of research lead to even more different forms of human computer interaction, very much based on the nonrational aspects that underly all our human to human interactions.

Acknowledgments

This research was funded by DARPA under contract number F30602-94-C-0204 administered through Rome Labs.

[4] A company founded by Rodney Brooks and Colin Angle as a spin-off of the MIT Artificial Intelligence Lab

Figure 12: A baby robot built at IS Robotics/-Artificial Creatures by Chikyung Won and Jay Francis, and programmed by Colin Angle and Rodney Brooks. The face has five articulations, it has a speech synthesizer, sound input, many sensors to understand how it is being played with, and a full emotional model.

References

Brooks, R. A. & Stein, L. A. (1994), 'Building Brains for Bodies', *Autonomous Robots* **1**, 7–25.

Coen, M. H. (1994), A Software Agent Construction System, *in* 'Proceedings of the 1994 Conference on Information and Knowledge Management Workshop on Intelligent Information Agents'.

Horswill, I. D. (1993), Specialization of Perceptual Processes, PhD thesis, Massachusetts Institute of Technology Artificial Intelligence Lab, Cambridge, Massachusetts.

Inoue, K. (1996), Trainable Vision-based Recognizer of Multi-person Activities, Master's thesis, Massachusetts Institute of Technology Artificial Intelligence Lab, Cambridge, Massachusetts.

Katz, B. (1990), Using English for Indexing and Retrieving, *in* P. Winston & S. Shellard, eds, 'Artificial Intelligence: Expanding Frontiers', MIT Press, Cambridge, Massachusetts.

Lettvin, J., Maturana, H., McCulloch, W. & Pitts, W. (1959), 'What the Frog's Eye Tells the Frog's Brain', *Proceedings of the Institute of Radio Engineers* **47**, 1940–1951.

Mellor, J. (1995), Enhanced Reality Visualization in a Surgical Environment, MIT-TR AITR-1544, Massachusetts Institute of Technology Artificial Intelligence Lab, Cambridge, Massachusetts.

Stauffer, C. P. (1997), Scene Reconstruction Using Accumulated Line-of-Sight, Master's thesis, Massachusetts Institute of Technology Artificial Intelligence Lab, Cambridge, Massachusetts.

Zue, V. (1994), Human Computer Interactions Using Language Based Technology, *in* 'IEEE International Symposium on Speech, Image Processing and Neural Networks', Hong Kong.