Greenberg, S. (2004) Collaborative Physical User Interfaces. Report 2004-740-05, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada T2N 1N4, March.

Draft of an invited chapter to:
K. Okada, T. Hoshi, and T Inoue (Eds*) Communication and Collaboration Support Systems* (Advanced Information Technology Series) Information Processing Society of Japan.

Collaborative Physical User Interfaces

Saul Greenberg
Department of Computer Science, University of Calgary,
Calgary, Alberta, Canada T2N 1N4
saul@cpsc.ucalgary.ca

Abstract. Unlike the 'traditional' computer that is based on a screen, mouse and keyboard, physical interfaces for collaborative interaction are special purpose devices that can be situated in a real-world setting and are designed for particular collaborative contexts and uses. However this is a new design genre; developers do not yet know what these devices should do and what they should look like. In this chapter, I hint at the variety of design categories for collaborative physical interfaces, as suggested and illustrated by a collection of working prototypes created by researchers and students at the University of Calgary.

We will also see that two toolkits encouraged people to explore creative ideas in this new genre. Through *Phidgets*, people rapidly prototype physical user interfaces under computer control. Through the *Grouplab Collabrary*, people had an easy means to share data between interconnected devices and software. These combined tools became a media form that allowed researchers and students to create and develop new ideas concerning physical interfaces for collaborative interaction, or to vary already established ideas in interesting ways.

1. Introduction

1.1 Embodied Interaction

Consider how casual interactions happen in the real world. People notice others that are around, what they are doing, and what artefacts they are working with. They can easily determine if others are available. If they choose to do so, moving into communication is often a simple matter of approaching others: mutual and subtle signals are exchanged, and a conversation begins [16][19]. The real-world physical and social context often becomes the shared understanding grounding their talk, and surrounding resources are used to further their conversation [4]. This entire process is extremely light-weight, and as a consequence a meaningful and rich interaction can take place in even a few seconds. This is an example of *embodied interaction*, which leverages our physical presence in the real world and that is socially embedded within our real world practices and purposes [5].

In sharp contrast, our current generation of groupware makes computer collaboration almost entirely disjoint from real world activities. Making connections with others is heavyweight and awkward. People are only reachable if they are near their computer and, because callers are oblivious to what others are doing at the moment, distractions, interruptions and missed opportunities are the norm. Bringing in nearby real-world people, information and tools into the conversation - no matter how relevant - is excessively difficult. This dampens people's capacity to collaborate effectively. What has happened is that the computer is forcing people to collaborate on the computer's own terms, and as a result entirely ignores the deep social context and practice that defines human-human interaction [4][5].

The way to solve this problem is by designing for embodied interaction [5]. That is, we can construct collaboration tools that participate in the real world rather than stand apart from it, and that fit within particular environments and contexts of use. Embodied interaction means people act through technology that is inseparable from the real world, instead of using technology that is aloof from our physical and social environment.

1.2 Collaborative Physical Interfaces as a Vehicle for Embodied Interaction

My own particular interest in collaboration as embodied interaction is in the design of *collaborative physical interfaces*, defined as special purpose devices that can be naturally situated in a real-world setting and that are designed for particular collaborative contexts and uses. Unlike traditional computing devices (screen, mouse and keyboard) that isolate people from their real world context, collaborative physical interfaces encourage embodied interaction because they can be designed around several key features.

- *Tangible.* Devices lend themselves to tangible interface design, which Ishii defines as a general approach to human computer interaction that puts greater emphasis on physicality than traditional GUI design [1][13]. As a tangible device, digital information is presented in a physical form, and physical manipulation can be translated back to digital directions. They allow people to physically sense, grasp and manipulate computer information [13].
- *Ambient display* means that devices can unobtrusively portray non-critical information on the periphery of a person's attention [13][17]. The device adds information into the physical environment that competes with all other information; if well balanced, it is perceivable in the background of people's attention rather than the foreground. Examples include visuals (lights, motion), audio (mechanical noise, artificial signals), and touch (air flow, vibration).
- *Attentive in-depth interaction.* When people become interested in the information on the device, they can selectively attend to it through natural interaction. Moving closer to it may reveal information at greater fidelity. Exploring and interacting with the information in-depth may be a simple matter of touching or manipulating the device in natural ways *vs.* navigating through menus and other abstract controls.
- *Portability and placement.* A person can move small devices to locations that are appropriate to the current need, especially if it is un-tethered and movable (e.g., wireless and self-powered). For example, consider how one can adjust the salience of an ambient display. A person can relocate the display to balance their need and desire for that information. If very close by and in direct line of sight and hearing, it likely serves as an easily perceivable and reachable foreground device. If further away and not in direct line, it becomes an ambient display for peripheral perception.
- *Public and shareable.* Unlike traditional computers that are usually viewable and usable by a person seated directly in front of it, physical devices are easily shared by small groups [9]. When a device is located in a public room, all inhabitants can see its output (especially as an ambient display) and have equal opportunity to explore its information and to interact with it. Of course, the degree of sharing will depend upon where the device is in the room, and where it is in relation to each person. Devices can be moved to locations convenient to shared viewing and interaction e.g., atop a central table or bookshelf, or even passed around.
- *Environmental fit.* Devices can be designed to fit within their environmental surroundings i.e., it can take the form of furniture, artwork, aesthetic functional units, children's toys, kitchen appliances, and so on. This fits within Mark Weiser's notion of

ubiquitous computing, where computers become invisible in use because they blend into the environment.

- *Embodiment.* Devices can embody, personify and even add character to the information they carry. That is, instead of being perceived as a mere carrier and transmitter of information, they become that information. As one example, Kaminsky et. al. [14] describes how physical character-based toys (e.g., Microsoft's Actimates) can be leveraged as devices so that information and interaction is tuned to fit the personality of its character. In turn, this makes the embodied device recognizable, compelling and understandable. As another example, the device that represents a distant person can, to a certain extent, become that person. This will be discussed further in a later section.

There is now a great deal of interest in collaborative physical interfaces. Yet it is a relatively new design genre. Researchers and developers do not know what these new systems should do or what they should look like. While initial directions are suggested in the sub-disciplines of ubiquitous computing, pervasive computing, context-aware computing and tangible computing, only a few deal head-on with collaboration and physical user interfaces e.g., [1].

To understand what physical collaborative interfaces are, researchers should first look at the unfettered exploration fuelled by creativity to see all possibilities [8]. When this corpus is available for reflection, researchers can evaluate them against aesthetic, ergonomic, or utilitarian design criteria, and generalize ideas that work. While creativity is exploring the ways things could potentially be, design is choosing the ways things ought to be. This chapter is positioned at the beginning of this transition from creation to design.

In this chapter, I present and illustrate a variety of design categories for collaborative physical interfaces, as suggested by a collection of working prototypes created by researchers and students at the University of Calgary. My group had created several tools that made it very easy for people to prototype and implement such systems. With *Phidgets* [10], people rapidly build physical user interfaces under computer control. With the *Grouplab Collabrary* [2], people share data between interconnected devices and software. The combination of these tools became a media form that allowed researchers and students to create and develop new ideas in collaborative user interfaces, or to vary old ones in interesting ways [8].

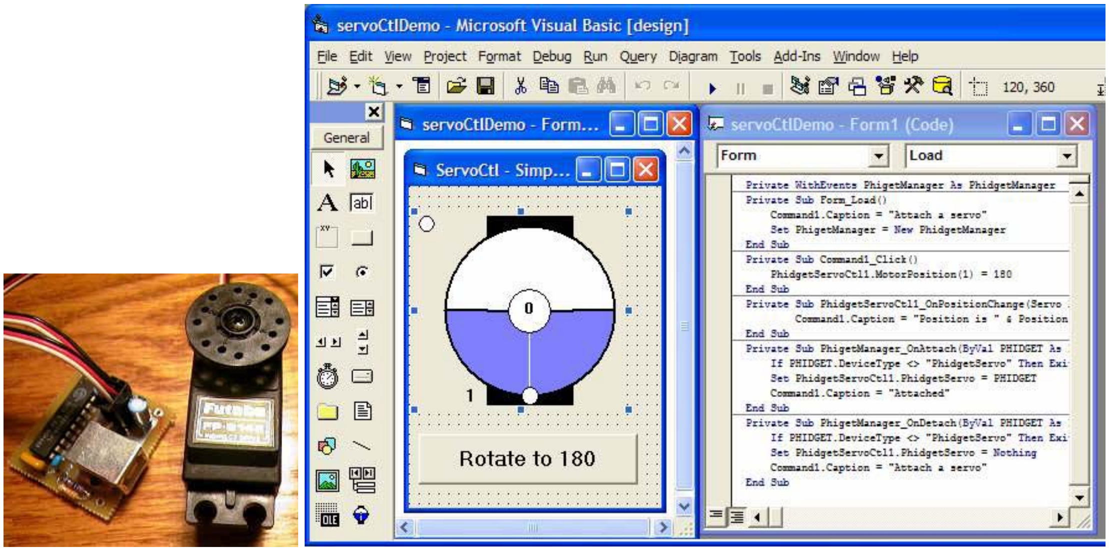
This chapter is written in two parts. To set the scene, I begin with a brief summary of Phidgets and the Collabrary toolkit. Afterwards, I will present design categories for collaborative physical user interfaces, and illustrate them by example. The implicit connection between the two parts is that we need to know the nature and technical affordances of these toolkits and how they encourage people to explore certain creative directions over others.

2. The Technology

This section briefly summarizes two technologies used to rapidly prototype collaborative physical user interfaces: Phidgets and the Grouplab Collabrary.

2.1 Phidgets

While creating specialized physical devices under computer control is an exciting new area in human computer interaction, everyday programmers face considerable hurdles if they wish to create even the simplest physical user interfaces. Most lack the necessary hardware training. Those willing to learn find themselves spending most of their time building and debugging circuit boards, firmware and low-level wire protocols rather than on their physical user


servoCtlDemo - Microsoft Visual Basic [design]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
120, 360
General
servoCtlDemo - Form...
ServoCtl - Simp...
0
1
Rotate to 180
servoCtlDemo - Form1 (Code)
Form
Load

```
Private WithEvents PhigetManager As PhidgetManager
Private Sub Form_Load()
    Command1.Caption = "Attach a servo"
    Set PhigetManager = New PhidgetManager
End Sub
Private Sub Command1_Click()
    PhidgetServoCtl1.MotorPosition(1) = 180
End Sub
Private Sub PhidgetServoCtl1_OnPositionChange(Servo
        Command1.Caption = "Position is " & Position
End Sub
Private Sub PhigetManager_OnAttach(ByVal PHIDGET As
    If PHIDGET.DeviceType <> "PhidgetServo" Then Exi
    Set PhidgetServoCtl1.PhidgetServo = PHIDGET
    Command1.Caption = "Attached"
End Sub
Private Sub PhigetManager_OnDetach(ByVal PHIDGET As
    If PHIDGET.DeviceType <> "PhidgetServo" Then Exi
    Set PhidgetServoCtl1.PhidgetServo = Nothing
    Command1.Caption = "Attach a servo"
End Sub
```

a. A Phidget Servo Motor **b.** Programming a Phidget Servo Motor in Visual Basic

Figure 1. The PhidgetServo

interface designs. The problem is that the software industry has not provided programmers with adequate building blocks for rapidly prototyping physical user interfaces. This leaves them in a position similar to early GUI researchers who had to build their widgets from scratch, or to early graphics researchers who had to build their 3D environments from the ground up. Given this onerous situation, it is no wonder that most research on physical user interfaces comes from well-known academic and industrial research laboratories.

Our solution was to develop a toolkit for rapid development of physical widgets, or Phidgets [10]. Our approach was to provide programmers with pre-packaged hardware devices. These include boards for controlling and gaining feedback from: servo motors (e.g., as pictured in Figure 1a), sensors, actuators, switches, LEDs, RFID tag readers, small screens, DC devices, power bars, and so on. All can be accessed by a very simple API, where software can even be 'dropped into' software applications via an interface builder. This familiar programming paradigm is directly analogous to how graphical user interface (GUI) widgets are programmed.

For example, imagine a conceptually trivial system, where a programmer wants to control the servo motor in Figure 1a both through a graphical control that allows fine-grain adjustment of the motor position, and by a button that would (say) rotate the motor to 180 degrees. Doing this requires the following steps.

1. Connect the *phidgetServo* (Figure 1a) to the computer via a USB cable (this can be done at any time).
2. Through a programming environment and interface builder (e.g., Visual Basic), drag and drop both a graphical button widget and a graphical phidgetServo widget from the tool palette (left side) into the window. This is how the interface on the left of Figure 1b was constructed.
3. Write a few lines of code that checks that the phidgetServo is actually connected to the computer and another few lines that controls the interaction between the button and the phidgetServo widget. For example, when the button is pressed, its callback just has to invoke `PhidgetServoCtl1.MotorPosition(1) = 180.` As seen in Figure 1 (right side), the entire program is quite short.
4. Run the program. The user can continuously rotate the actual motor to any position through the phidgetServo graphical widget, and can press the button to move the motor to 180 degrees.

```
/contacts                                ←stores per-contact data
     /1                                  ←unique ID for 1st contact
           /name    :- 'Saul'            ←friendly name
           /status  :- 'online'          ←availability status
           /photo   :- <jpeg image>      ←their picture
           /message :- 'Hi Mike'         ←chat message
     /2                                  ←unique ID for 2nd contact
           /name    :- 'Mike
           /status  :- 'busy'
           /photo   :- <jpeg image>
           /message :- 'Hi Saul. What is up?'
```

Figure 2. A shared dictionary for instant messaging

2.2 The Collabrary

Groupware developers have to deal with how data is distributed across the network, and how to process shared data. The problem is that groupware developers often end up devoting a great deal of their time experimenting, building and packaging various network services for groupware. This is because the services typically available to developers, mostly bare-bones TCP sockets, are too low level to help them rapidly prototype groupware applications. This is especially true if the data is complex, e.g., structures, objects, and multimedia.

Our solution was to develop the *Grouplab Collabrary* toolkit for rapidly creating groupware prototypes [2]. The goal of the Collabrary is to ease many of the mundane yet tricky aspects of groupware programming, particularly in how multimedia is captured and processed by a client and then distributed to other clients over a network. The Collabrary has two major parts. The first part includes components that ease multimedia capture, display and manipulation e.g., video, audio, images. The second part is a hierarchical shared dictionary component that eases how data is captured in a dictionary data structure shared between distributed groupware processes. It is this *shared dictionary* that encapsulates the network service layer. Unlike TCP programming, developers spend their time thinking about the data they wish to share rather than on networking primitives.

To explain, the Collabrary is a client/server architecture based on the concept of a *notification server* [18]. Essentially, clients (all groupware instances) can both publish and subscribe to data. If a client publishes some data, subscribers will be notified of this via an event mechanism, and can then retrieve the data. The Collabrary can act as a pure notification server [6] where no data state is maintained, i.e., once the published data is distributed to its subscribers, the server discards it. However, the Collabrary can also act as a data repository, where it stores all published data as hierarchical key/value pairs in a centralized dictionary structure, which can then be retrieved at any time by clients.

To illustrate via a trivial example, consider how Figure 2 implements an instant-messenger system by the way clients publish and subscribe to a shared dictionary data structure. The contact list is maintained under a key called `/contacts`. Different contacts are hierarchically listed as a unique ID number under this key, e.g., `/contacts/1` and `/contacts/2`. Specific information about particular contacts are child keys under this ID. For example, the first person would have their information under `/contacts/1/name`, `/contacts/1/status`, `/contacts/1/photo` and `/contacts/1/message/`.

To make this all work, new clients connecting to the shared dictionary publish information about its user, e.g., the user's name, initial availability status and photo. The client updates this information over time by republishing new values if any are changed. All clients also subscribe to specific keys, perhaps by pattern-matching: when any data is added, altered or deleted, clients are immediately notified of the change and can use information about it to regenerate the interface. For example, to handle the appearance and disappearance of new contacts, the client can subscribe to `/contacts/*/name`. When it receives a notification that a name has been added, changed or deleted, it can update the contact list by invoking `ForEach n in /contacts/*/name, Print n`

To implement the chat dialog, clients would subscribe to the `/contacts/*/message` key. When a user types text, the client publishes that text into the `/message` key. All other clients would receive a notification that a message has been added or changed, and would retrieve its author's name and the message contents and insert it in the chat dialog.

2.3 Phidgets and the Collabrary as a distributed model-view-controller system

The Collabrary and Phidgets combine to create a powerful platform for collaborative physical interfaces. In essence, the Collabrary shared dictionary serves as a distributed model-view-controller system [12]. The *model* is the abstract data contained in the dictionary. The *view* is the part of a phidget that displays changes in the data. The *controller* is the part of the phidget that collects input from the user or from the environment, e.g., through sensors, RFID tags and actuators, and publish that data into the model in an appropriate form. Similarly, other associated software can react to the software (as a view) or add relevant data to the model (as a controller).

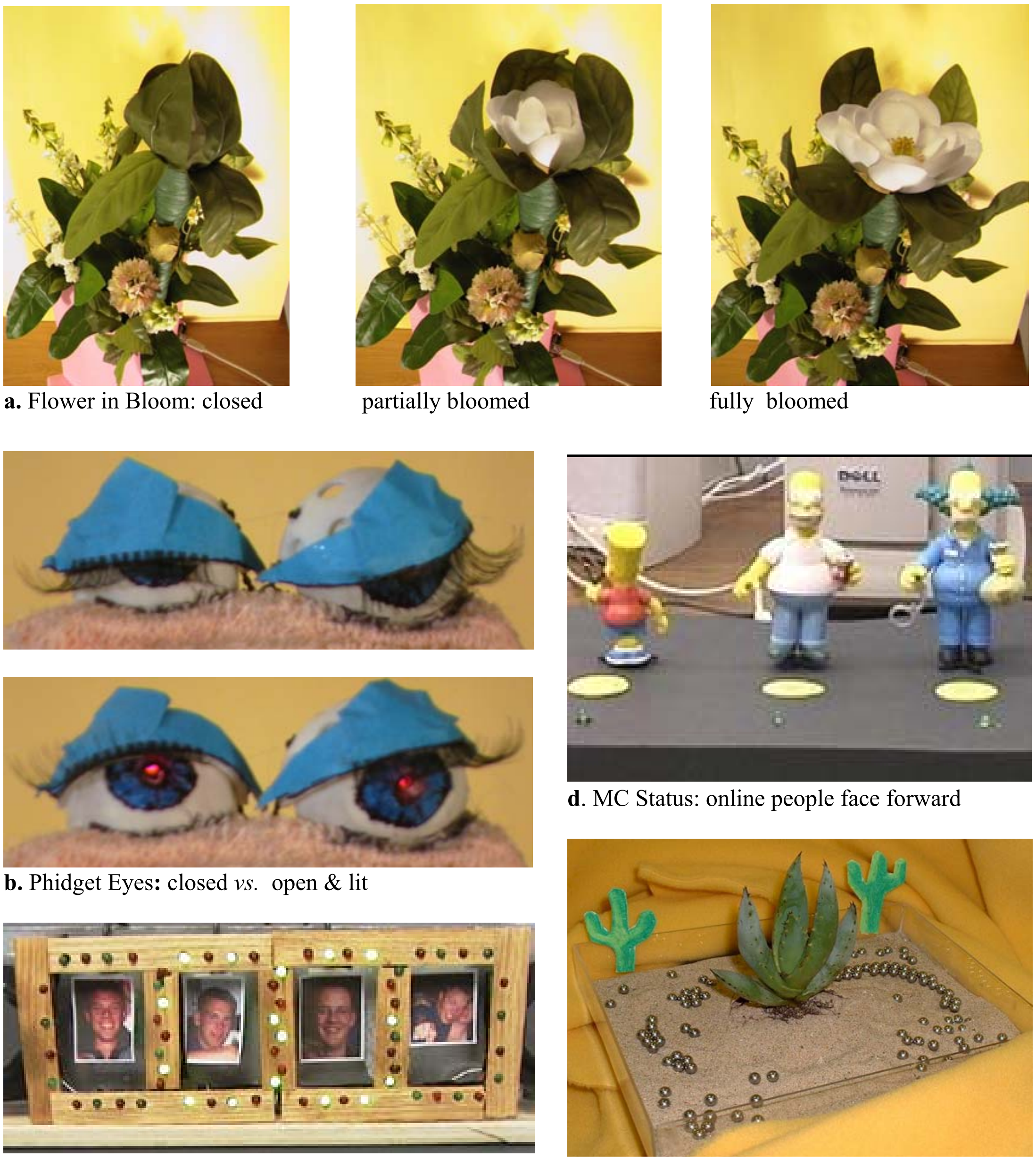
3. Design Categories

These toolkits opened up a new creative design space where people could quickly craft physical devices, connect them to each other, and have them run under computer control. As hoped, the advent of Phidgets and the Collabrary heralded an explosion of physical user interface designs by both researchers and students at the University of Calgary. Some designs were part of serious projects pursued within a thesis or research context. Others were curiosity-driven explorations, where graduate students in particular just wanted to try out the technology as a means to explore ideas. I also provided the technology to undergraduate students in an advanced human computer interaction design course, where I gave them a short open-ended assignment to see what they could do with it (they were free to develop any project they wished as long as it used Phidgets). The results were remarkable. While a few researchers and students replicated examples of physical user interfaces reported by other researchers, most produced their own innovative systems.

Many of these systems involved collaborative physical user interfaces. Yet they vary considerably in design, in scope, and in how they exploit the several properties of physical devices mentioned in the introduction. They also vary greatly in how they would address embodied interaction. Using these as inspiration, I present design categories for collaborative physical user interfaces, where I illustrate them by examples taken from the various projects. While I will not describe how they are implemented, this should be reasonable to deduce given the description of Phidgets and the Collabrary in the previous section.

3.1 People status indicators

People maintain awareness by naturally tracking the state of many things in the real world. This awareness also serves as a fundamental means for tracking collaborative events, and for starting and maintaining casual interaction.



a. Flower in Bloom: closed

partially bloomed

fully bloomed

b. Phidget Eyes**:** closed *vs.* open & lit

c. Messenger Frame: online people are lit up.

d. MC Status: online people face forward

e. Magnetic Desert

Figure 3. People Status Indicators

When people are not physically present, physical devices can become status indicators that display information to a person about one's distant colleagues. This can include their colleague's presence and availability, and their interest in conversation. They can be effective vehicles for embodied interaction because they exploit several of the features listed in Section 1.2. First, they become an ambient display. In contrast to screen-based solutions that demand foreground attention (e.g., instant messaging contact lists), people can see these devices at the periphery of their attention. Second, the device portability and placement means that people can also relocate the device in their physical environment to match their current level of interest in that distant person. Third, because devices are public and shareable, that information is a resource available to others in the room. Fourth, they can also embody the activities of the remote person, either concretely (e.g., as a picture) or abstractly.

Status of a distant person is easily done by mapping people's implicit or explicit activity status onto discrete, continuous or even abstract displays. This explains why many projects were some type of status indicator. For example, *Flower in Bloom* by Susannah McPhail is a floral arrangement made out of artificial flowers (Figure 3a). The central large flower blooms under program control from a continuum ranging from closed to full bloom. As seen in Figure 3a's various snapshots of the flower over time, this offers a single continuous variable for showing information ranging from an absent person (closed) to present but busy (half bloom) to present and available (full bloom). Another example is *PhidgetEyes* by Debbie Mazurek, whose eyes can open and close to any position and whose pupils light up (Figure 3b). This gives two variables to map information. For example, the eye state can map presence collected from a person's implicit activities (similar to the Flower in Bloom), while the lit-up eyes can signal that the other person is explicitly interested in communicating.

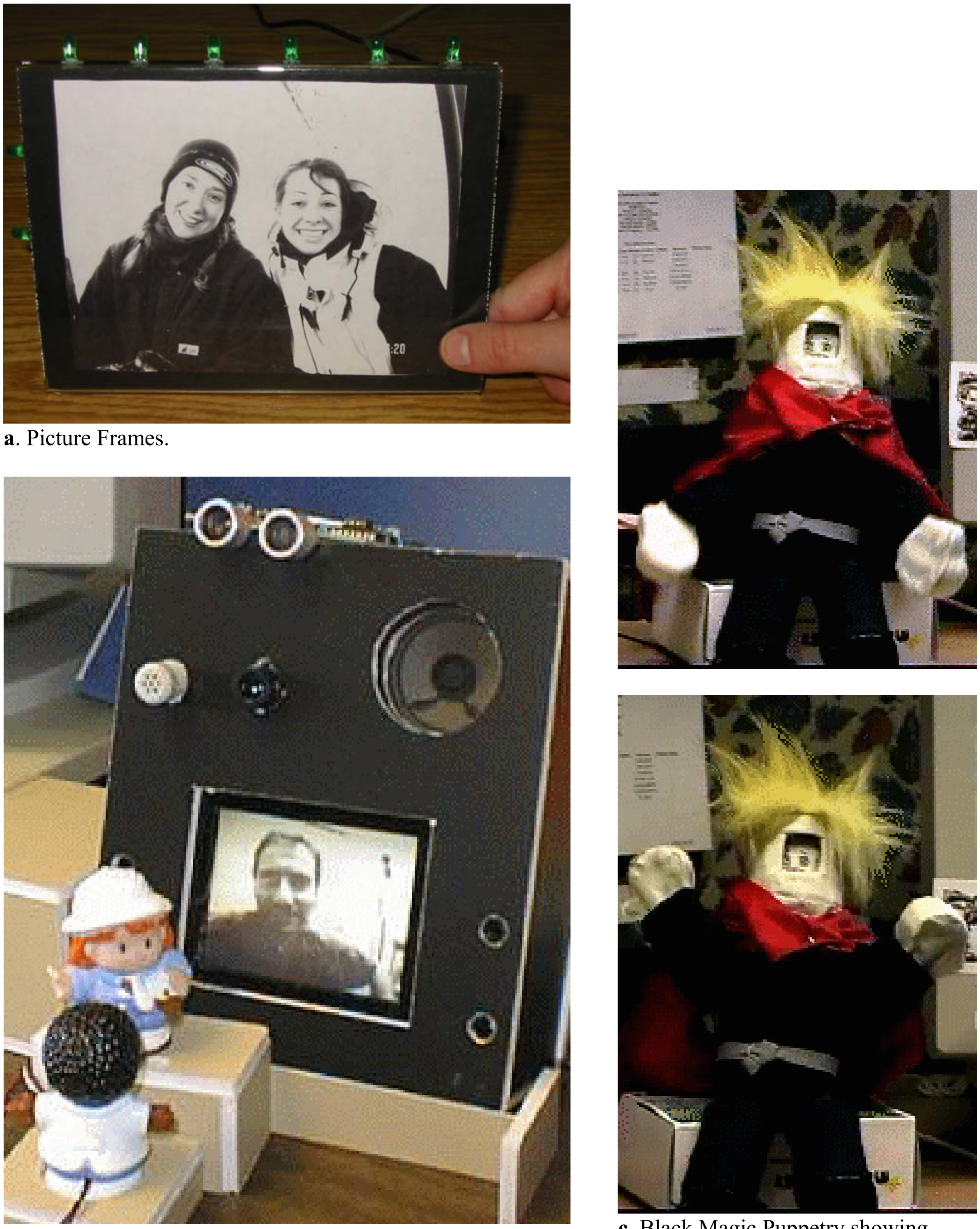
Status of several distant people is a variation of the above, where several people's status is mapped onto an array of devices. *Messenger Frame* by Mike Hornby-Smith (Figure 3c) presents a contact list as several photos. A particular contact's photo is lit up and a sound cue generated in an embedded speaker as that contact appears online. In *MC Status* by Christian Leith (Figure 3d) several contacts are represented by individual figurines. Offline figurines face the wall, online figurines face forward, and in-between states are represented by partial forward rotation.

Status of activity within a place gives an indication activity within a room. For example, *Magnetic Desert* by Kari Basaraba is an ambient display that moves metal bearings around a dish at a rate that varies with the amount of motion detected in the room (Figure 3e). Any motion indicates presence, and the degree of motion suggests the degree of activity within the room. The *Coffee Room Monitor* by Chris Willott is similar (not pictured), but in this case it shows which people are physically present in a coffee room. Pegs on the board are assigned to various people, and these pegs are used to hang people's physical coffee cups. Pegs are instrumented so that the system can detect if a peg is holding a cup or not i.e., if the person has taken the cup to drink coffee. This activity is displayed on a separate physical device that represents each person as a figurine.

3.2 Communication Channels

Given that status indicators show the presence of others, it is only natural that people can respond to that information by opening a communication channel through it. Through attentive in-depth interaction (Section 1.2) people can act on the information they have about another's status, either by finding more details or by actually opening a channel. They usually do this via the tangible interaction properties of the device e.g., by approaching it (which is sensed) or by touching it.

Mutual attraction. A device can show that one person is thinking about another, and that this thought is reciprocated. This can be used to initiate an actual exchange, or just to tell another person that they are in one's thoughts. For example, Kathryn Elliot used the Tangible Media Group's Lumitouch frame idea [3] in her implementation of *Picture Frames.* These are two or more picture frames (Figure 4a) that interact with one another as an ambient display. Co-workers, friends or loved ones separated by distance use the Picture Frame as a way to make others aware of their presence in a non-intrusive manner and/or to communicate emotional content. Exploiting the tangible property of the device, touching one frame causes lights to blink on its partners, and the remote person can respond in turn. Furthermore, a digital frame (not shown) simulates the behaviour of the physical one, and also displays the name of the person who sent the touch. Any 5"x7" picture may be inserted into the Picture Frame, which transforms the device into a fairly concrete embodiment of the remote person.



a. Picture Frames.

b. Active Hydra

c. Black Magic Puppetry showing different hand gestures

Figure 4. Communication channels

Instant Messenger embodiments. Physical devices can also serve as embodiments of people already listed on Instant Messenger systems (IM), where they show status information collected from an instant messenger contact list and open up the IM channel as needed. The *Messenger Frame* and *MC Status* systems mentioned above and shown in Figures 3c & d are also examples of IM embodiments. If one touches a picture in the Messenger Frame or the area in front of the figurine in *MC Status*, the appropriate chat window will appear on a nearby computer. A different form of an Instant Messenger embodiment is *Black Magic Puppetry* by Rosemary Sanchez, which displays emoticons in a physical form rather than text messages. It is realized as a voodoo doll, where the arms and facial expressions change to convey the mood of the conversation as represented by the emoticons (Figure 4c).

Digital but physical surrogates are tangible representations of remote people, positioned within the environment and under digital control. They embody the remote person within the device not only by the awareness information it presents, but by encapsulating the communication channel within it as well. Natural interactions with the surrogates control the communication. For example, the *Active Hydra* surrogate (Figure 4b) created by Hideaki Kuzuoka and myself [11] embodies a single remote person by showing a video and audio connection to that person within a device, and that opens the communication channel as a function of proximity. Similar to *MC Status* (Figure 3d), the figurine in front of the *Active Hydra* in Figure 4b also shows the availability of the remote person by the direction it is facing.

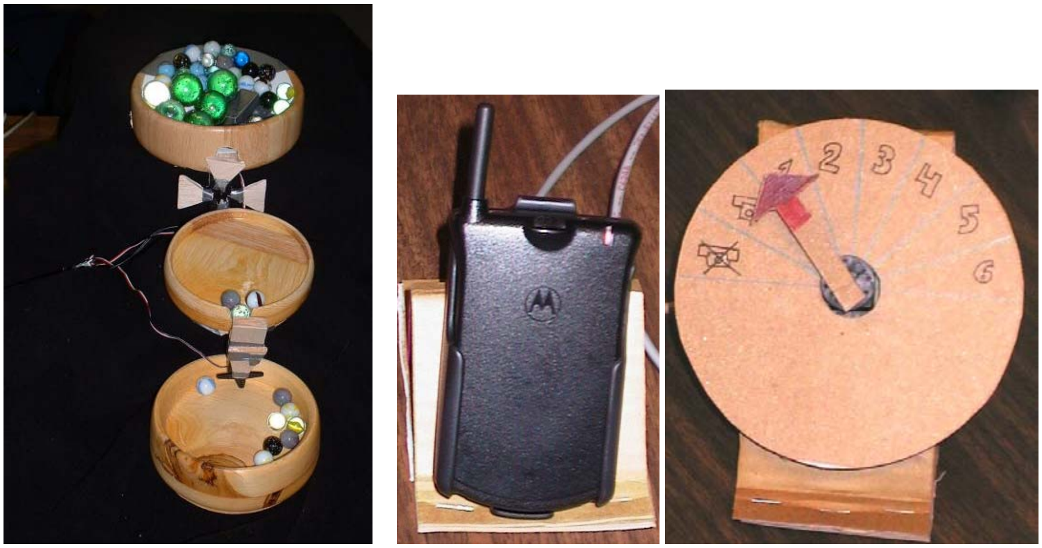
Tangible Communication. In real life, communication is not only by sound and visuals, but by touch as well, especially between intimates. While our students did not build any systems within this category, several examples appear in the literature that exploits the feature of tangibility. *Hand Jive* is a pair of devices designed for play, each with two movable but connected balls [7]. Moving a ball on one device causes its partner ball to move on the other; people play together by developing patterns of movement and rhythms. Similarly, inTouch [1] is a haptic device that gives the illusion that two distant-separated people are manipulating the same physical device. It comprises two devices, each consisting of three cylindrical rollers mounted on a base. When a person rotates one of the rollers, the corresponding roller on the remote object rotates in the same way. Through force feedback, the two partners can feel, stop, or counter each other's motions. Of course, one does not have to touch the devices to get information from it. Each acts as an ambient display, where motion (triggered by a remote person's manipulation) acts as an invitation to the local person to join in, which one can easily do simply by moving towards it and initiating interaction.

3.3 Notifications of Asynchronous Messages

Much communication these days arrives asynchronously: phone messages, email, and a host of other items. Yet current computer technology expects people to either continuously review incoming items to see if there is anything new (e.g., such as opening up one's inbox), or to notice notifications that appear on the display (e.g., the current appearance of a mail icon). Another option is to embed notifications within a device which can be situated in the physical environment as an ambient display, so that people notice these notifications and can optionally retrieve the information through that device by in-depth interaction.

Momentary notification of incoming messages. The device can indicate its notification as the message arrives, but will not persist in showing that notification. If the person happens to see or hear that notification they can react to it, but otherwise it is missed. A whimsical example is the *Nerf Email Notifier* by student Carman Neustaedter (not pictured). Taking the form of a physical mailbox, it notifies its owner about incoming mail by rotating the mailbox towards the person and shooting a nerf disc out of the mailbox opening (it does this by computer control of a nerf gun, a child's toy that fires very soft foam discs).

Persistent notification of incoming messages. A different approach is to show notifications persistently over time, so that people can notice that messages have arrived after the fact. Another email-based example is Marble Mail by Shannon Goodman (Figure 5a), inspired by Durrell Bishop's Marble Answering Machine project at the Royal College of Arts. It is a physical representation of the state of a person's email box. The top bowl acts as a storage area representing potential email. As email arrives, a marble drops to the middle bowl containing unopened messages. As email is read, a marble drops into the last bowl containing



a. Marble Mail

b. Missed calls, showing the cell phone in a cradle and the display of missed calls

Figure 5. Notifications of Asynchronous messages

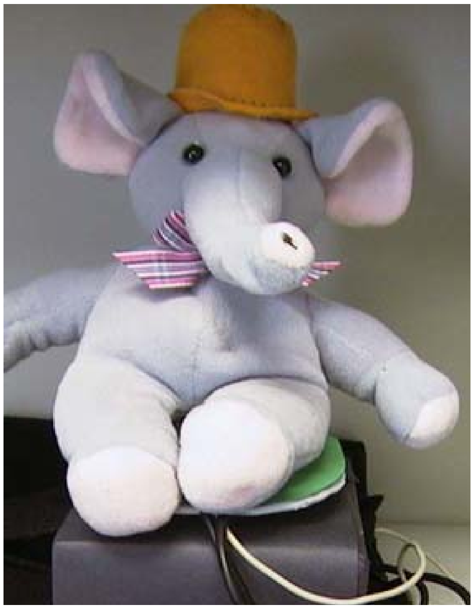


Figure 5c . Ele-Phidget

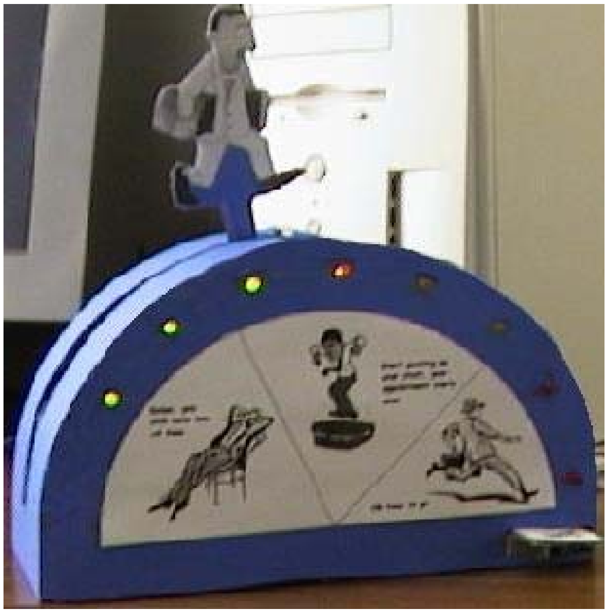


Figure 6. Notification of meetings: Appointment assistant

read messages. The dropping marble serves as the momentary notification, while the number of marbles in a bowl shows as the persistent notification. Raul Nemes' *Missed Calls* works on cell phone messages rather than email (Figure 5b). The device tell him how many calls he missed on his cell phone when he did not carry it with him i.e., when the phone was on a special cradle. By monitoring signals detected by a sensor embedded in this cradle, the device could detect when the phone was placed in or out of the cradle, and when the phone vibrated as it rang. A physical dial (which could be positioned anywhere) displayed whether the phone was in the cradle or not, and how many times a call had come in without being answered. When a person took the phone off the cradle, it would automatically reset itself. A final and more complex example is *Ele-phidget* by Shivaughan Warwaruk (Figure 5c). A plush toy elephant acts as a persistent message notification for an audio chat program. It normally faces the back wall, but when a message arrives, the elephant turns around and faces the front.

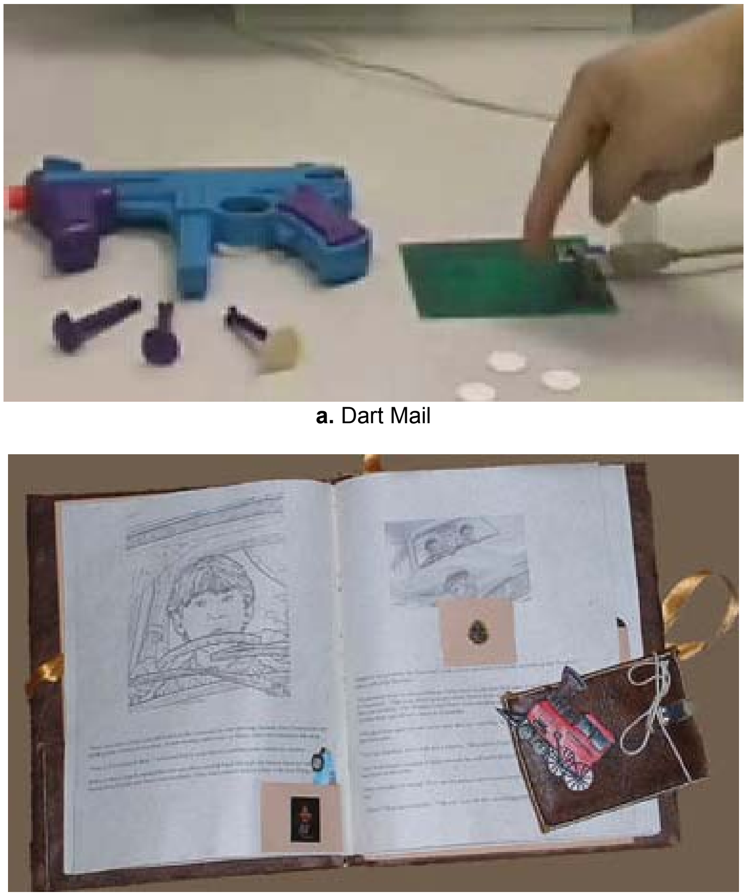
However, it also acts as a physical but digital surrogate; when the person pushes the elephant's stomach the message is played back. After all messages are heard, the elephant turns away. To record a message, the person squeezes the elephant’s head and speaks into the elephant's trunk. A second squeeze sends the message.

3.4 Notification of Meeting Activities

Continuing on the theme of notifications, devices can also serve as an ambient display of information about upcoming meeting activities. For example, *the Appointment Assistant* by Zaid Alibhai (Figure 6) is an ambient appointment display that interacts with a user’s on-line calendar to remind them of upcoming appointments. As an appointment approaches, the figure on the top of the display physically moves along the scale and LEDs light up to further indicate the relative time remaining before the next appointment.

3.5 Information exchange

A powerful use for devices is for them to act as a means for exchanging electronic information in a physical form. Because one person is able to physically give a device to another person, this exploits their properties of portability, tangibility, and shareability. Of course, the actual exchange of media such as floppy discs, CD-ROMs achieves this purpose for static



a. Dart Mail

b. Interactive Storybook

Figure 7. Information exchancge

information. However another mechanism is to have the exchange medium serve as a handle to information so it can be retrieved from the computer. That is, instead of exchanging cryptic and hard-to-remember location strings and access codes (such as URLs, file locations, passwords), one exchanges a tangible representation that holds all this information and that can retrieve it automatically.

Students Anthony Tang and Eric Pattison developed this theme through their quirky *Dart Mail.* The system comprises a toy gun, several shootable rubber darts containing RFID tags, and RFID tag readers (Figure 7 a). A person selects a document by first dragging and dropping it into a GUI interface on a traditional computer display. He then links the document to a particular dart by waving the dart over the RFID tag reader. He then shoots the dart at the intended recipient, who can then retrieve the document at any time on their computer by waving the dart over their own tag reader. Of course, the dart gun is just a playful representation of this system; Tang and Pattison also embedded tags into other physical form factors that can be exchanged between people e.g., cards, key fobs, and so on.

Another example using RFID technology as an exchange medium is the *Interactive Storybook* by student Christina Escabillas (Figure 7b). People can retrieve physical objects crafted to fit the storyline from small envelopes within the book (objects actually contain the tags) and wave them over a 'magical surface' (a tag reader) that invites the reader to interact with information they represent. This causes a display to appear on a traditional computer, where the displayed items relate to the part of the story containing the object e.g., information or scenes related to the current page that they are reading. While Escabillas' example is based on a Harry Potter story, we can easily imagine other examples where a writer can construct reports using similar ideas, which are then disseminated to colleagues.

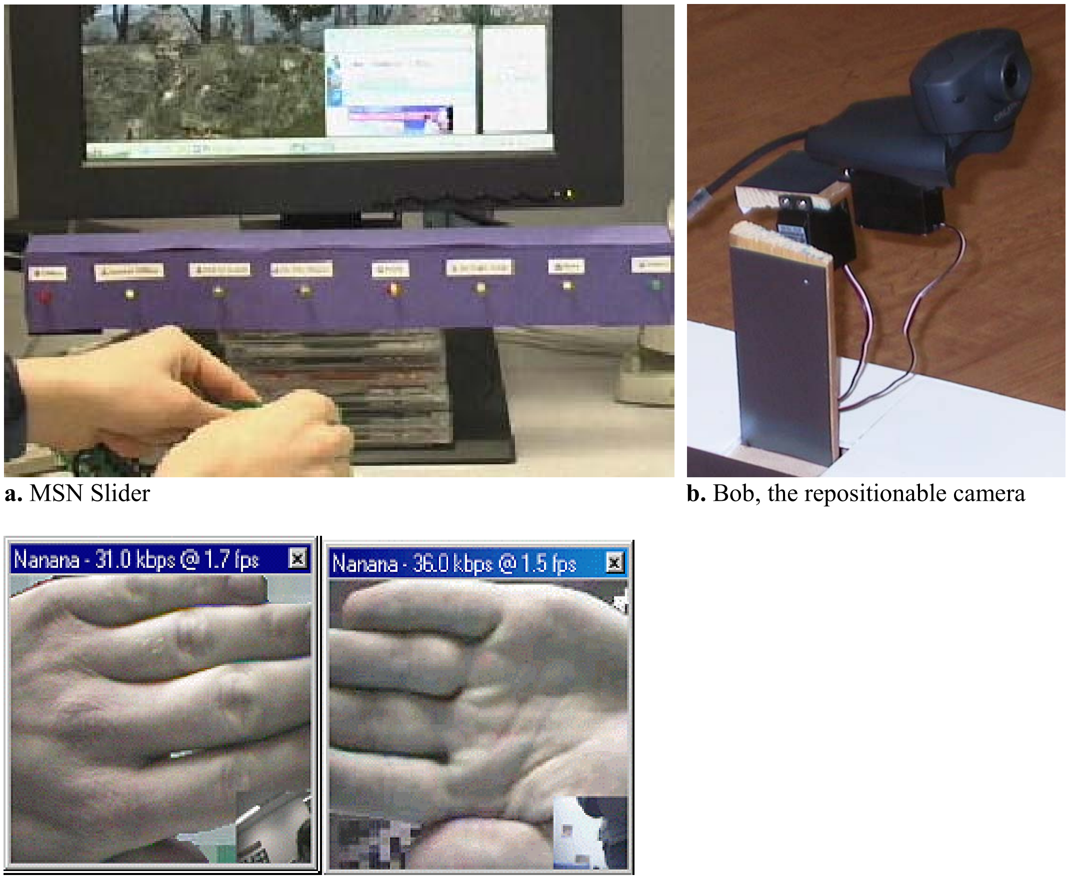
3.6 Privacy control

Privacy is a huge concern in many collaborative systems. Physical devices can help alleviate privacy concerns in several ways. They can show privacy status (as an ambient display); they can be used to directly adjust privacy factors (as a tangible device) and they can show appropriate feedback of its state depending on how and where information capture devices appear within the physical surroundings (placement and environmental fit) [15].

Privacy actuators are physical controls that let people adjust their desired privacy levels. A simple example is *MSN Slider* by Michael Rounding, a physical slider that lets a person quickly set their on-line status on MSN Instant Messenger (Figure 8a). Moving the slider changes the state from online, to busy, away, offline, and so on. A physical display made up of labelled lights provides immediate feedback on the person's on-line status.

Perceptible information gatherers are devices that, while capturing information from the environment, also reveal that they are doing so. This feedback helps people understand the what, where and when of personal information gathering. One example is *Bob, the repositionable camera* by Alan Flanders (Figure 8b). This is a web camera that tracks people by moving itself to keep the person's face in the center of its view (under the covers, it uses servo motors to move the camera in both the X and Y axis and uses Bradski's face tracking algorithm). Because camera motion is highly visible and audible (because of the sound of the motors) people can immediately perceive that they are being tracked.

Information blockers combine the above categories by not only showing that information is being gathered, but also by allowing people to block it by either implicit or explicit action. Several examples illustrate this, again by using a camera. In the *Active Hydra* mentioned previously, the presence or absence of the audio and the quality of the video portrayed within the surrogate is controlled implicitly by people's position relative to the surrogate. As in real life, both people must be close to the other person's surrogate if they are



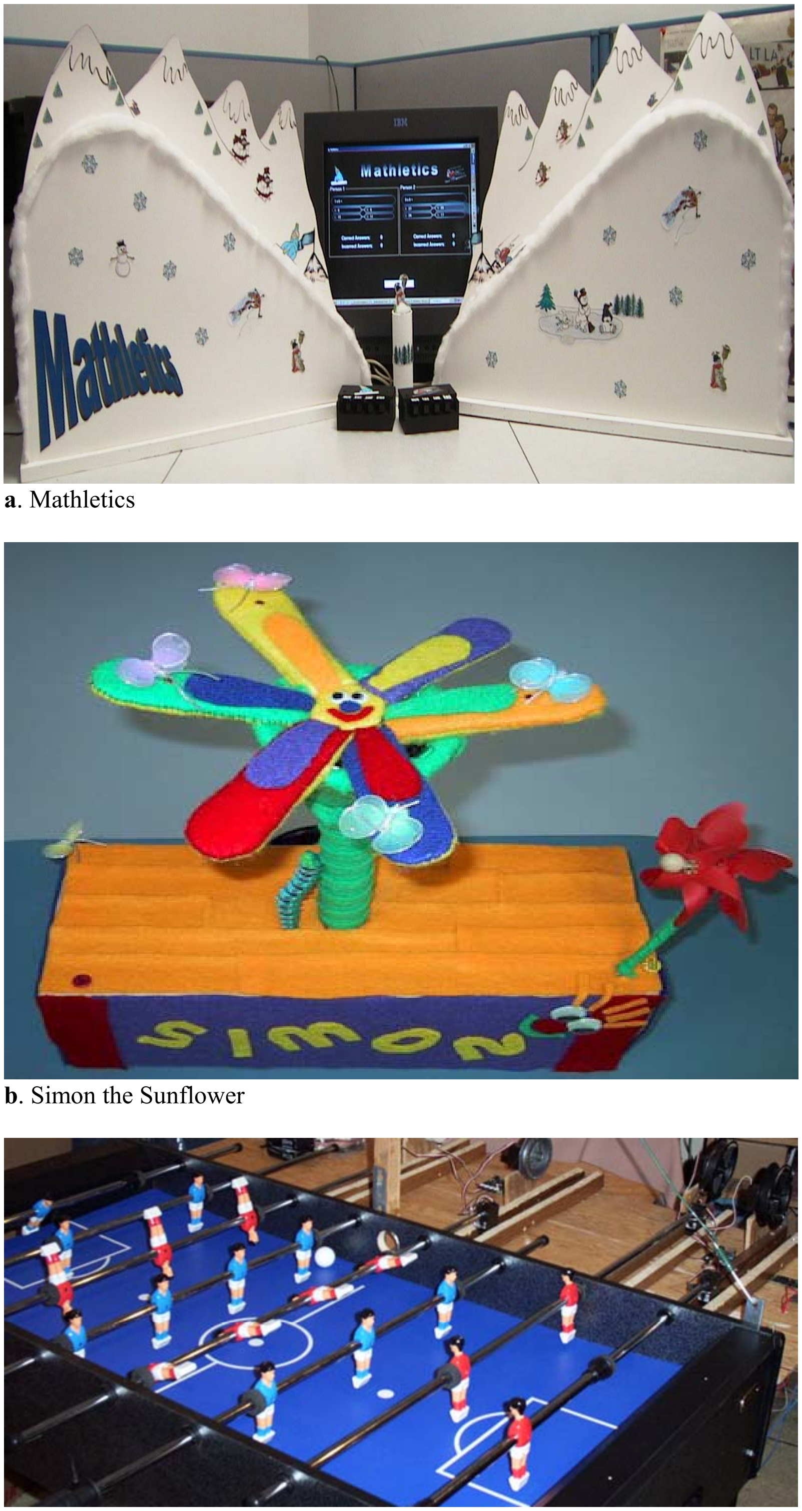
a. MSN Slider

b. Bob, the repositionable camera

c. Nanana and how both parties see blocked images

Figure 8. Privacy control

to see and here them in full fidelity. As one or both move away, the audio channel closes (so they can no longer hear what is going on in the other site). Even further away, and the video fidelity decreases by only periodically updating it. The Active Hydra also contains a second figurine (the one at the very front of Figure 4b) that can be used to explicitly adjust ones interest in them, which in turn adjusts what others can seen. If the person moves the figurine to face the camera, full interest (and thus full video/audio) is indicated. If moved to face away, then only partial interest is indicated, and the audio/video connection is throttled. If the figurine is laid down on its side, then the connection is blocked. Michael Boyle's *Nanana* is similar to the Active Hydra in that it too adjusts video fidelity as a function of proximity, except this time through progressive digital blurring of the video image. Nanana also allows a person to easily block the video through a manual gesture. A person just covers the camera with their hand (a simple algorithm detects when the image goes black for a few frames), which toggles the blocking mode. When one person blocks, she sees the image of the back of a hand superimposed over the remote party's video image; the remote party instead sees the image of the palm of a hand (Figure 8c). Both parties may block at any time—indeed even at the same time—but each is responsible for removing the block. This approach closely resembles one's tendency to cover the camera in "dire" circumstances.



a. Mathletics

b. Simon the Sunflower

c. FoosWars

Figure 9. Collaborative Games

3.7 Collaborative Games

People naturally collaborate over physical games, and a wide repertoire of such games now exists in every home: puzzle games, board games, construction games, and so on. Unlike computer displays, games based on physical devices are easily shared by small groups. People can easily see and interact with them. They can be moved to a place convenient to shared interaction e.g., atop a table, or even on the floor, and repositioned so they are convenient for a particular person e.g., as in turn-taking. They encourage play by inviting attentive in-depth interaction.

Turntaking collocated games are those where collocated people share the game device by taking turns. Seeing each other as they play adds to the game competition. For example, *Simon the Sun Flower*, developed by student Nancy Lopez, recreates Hasbro's Simon game for children as a flower with glow-worms located on its petals (Figure 9a). The flower creates a visual and audio pattern by playing a sequence of sounds while lighting up glow worms on particular petals (via LEDs). The child whose turn it is repeats the pattern by squeezing the bugs on the correct leaves (each contains a pressure sensor); if they make a mistake, the sound changes and it is the next child's turn.

Simultaneous collocated games are those where collocated people share the game device at the same time. Each person has their own input controls, and they can see how their input interacts with the other person's actions. For example, student Russell Kruger created *Mathletics,* a children's game intended to provide incentive for primary grade students to learn their multiplication tables. As shown in Figure 9b, it contains two ski hills and a computer display. Multiple-choice questions appear on the screen, and both students child independently enters the answer to their questions using a specially built controller. If they answer a question correctly, their figure progresses down the mountain. The first child to correctly answer 10 questions wins, at which time their figure reaches the bottom of the mountain, and the mountain light up.

Simultaneous distributed games are those where distributed people share a physical game device at the same time. One person plays on the physical game, the other accesses it remotely. For example, *FoosWars* by Mike Larke and Mike Clark (Figure 9c) is a soccer table game (also called foosball) re-instrumented for a distributed player. One person plays on the physical table while the other plays over the web (motors and pulleys activate the board, as seen in the upper part of Figure 9c). The remote player has a live aerial view of the table captured via a web camera located above the table, and directly manipulates his or her players through use of physical sliders.

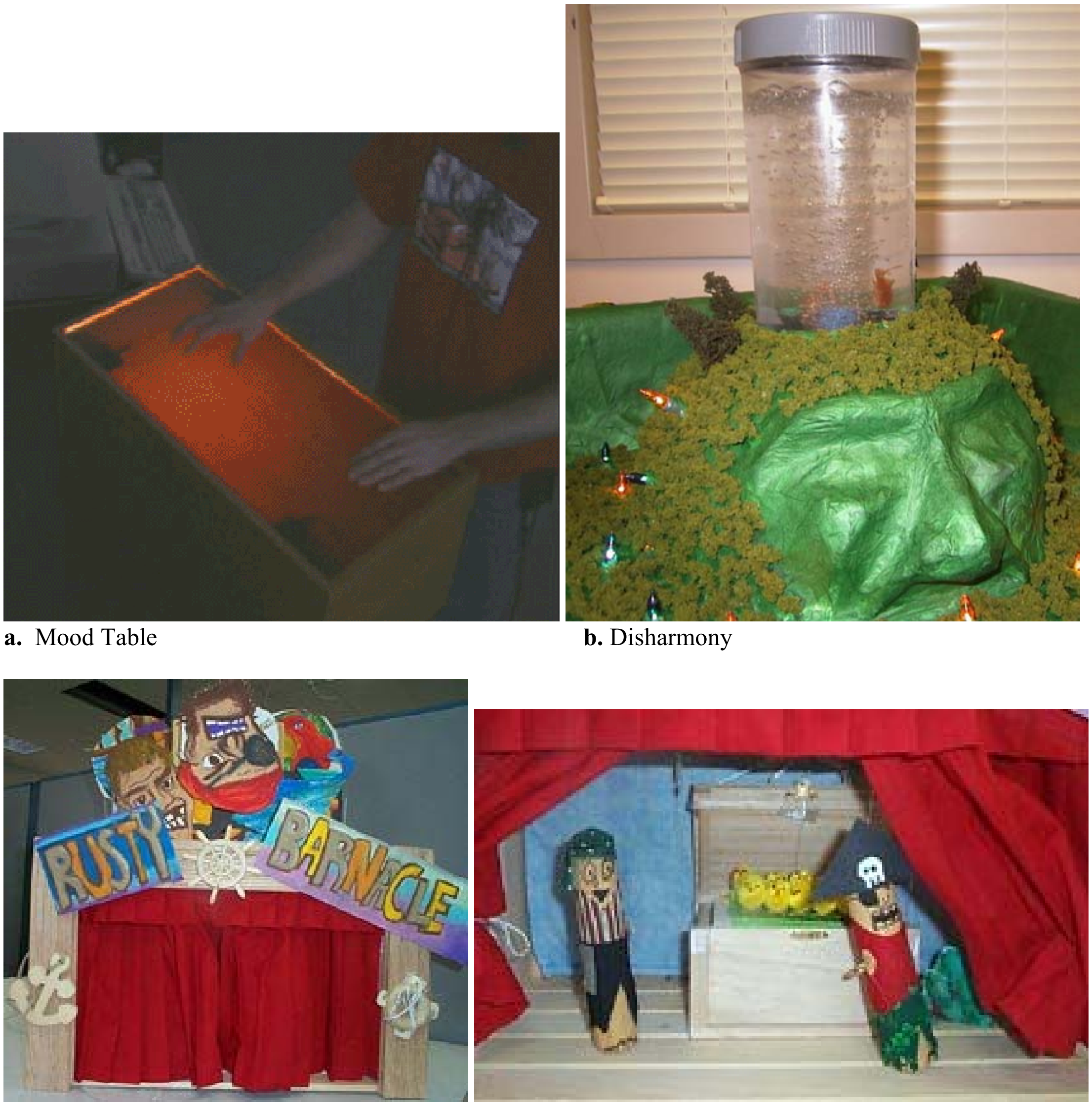
Of course, the above suggests other games categories. These can combine asynchronous games, where people can interact with it at different times with the ways that its players can be collocated and geographically distant.

3.8 Interactive Art

While art has historically been something that people look at passively, many museums are now interested in interactive art installations that encourage interactions between exhibit attendees. As embodied devices, these art installations are opportunistic, where people serendipitously meet other attendees, observe how they are interacting, and then join in.

One example is the *Mood Table*, developed by student Eric Pattison (Figure 10a). Several people can press the surface of the Mood Table with their hands at the same time, which reacts to these forces by displaying different light patterns on its surface and by reciting poetic phrases. Of course, both the light patterns and sounds are visible and audible to all in the room, encouraging others to observe and to join in.

A similar example is *Disharmony* by Mike Polowick (Figure 10b). This abstract art piece is a conglomeration of loosely-related themes designed for provoking thought in the viewers. All parts were crafted with deliberate intention, but there is no specific meaning; any interpretation is correct. Depending on how people move a pieces on a chess board (not



a. Mood Table

b. Disharmony

c. Rusty Barnicle, showing the stage and a scene with computer-controlled puppets

Figure 10. Interactive Art

shown), bubbles disturb real fish in their fish bowl, lights blink, a disk spins, and so on.

A quite different example are devices used to create a live show. An example is *The Rusty Barnicle* by Kevin Foster, a computerized puppet show and theatre (Figure 10c). The operator can use controls to work the curtains, a chest of gold, and two characters of the puppet show. The story is a pirate tale of adventure in the Caribbean, and of course the operator can pace the tale in response to the audience reaction.

Finally, devices can be used to encourage audience participation and exploration in an interactive performance. In one of four rooms comprising a theatre space, Kevin Foster along with other performing artists positioned three human-sized clear boxes in a room, each containing an actor and a special light at its top. Lights were controlled by various sensors e.g., a light sensing device hidden in a large pipe, a force sensor embedded in a big and oddly shaped button, and a slider actuator made into an peculiar-looking factory switch. All lights were off when the audience entered the room, and audience members had to figure out how to turn on the lights in each box so that the actors could interact. Depending on which actors were lit, they would act out different scenes. Audience discovery of these controls was deliberately made challenging. For example, to activate a particular actor the audience had to discover that they had to take a light dangling in the middle of the room and place it into the pipe with the light sensor.

4. Summary

This chapter only hints at the possibilities of categories for collaborative physical interfaces. There are, of course, many other research groups now working on these types of devices. For example, Hiroshi Ishii's Tangible Media Group was arguably the first to explore devices as a serious research area. Since then, many other examples have appeared in the ACM CHI, ACM UIST and ACM CSCW and UBICOMP literature as well as a host of other conferences and journals. Researchers are trying to understand the characteristics of these devices. Examples include heuristics for ambient displays [17], the role of digital surrogates [11], privacy devices [15], and character-based devices [14].

Physical collaborative user interfaces are now in a stage of invention, where the original breakthrough idea has been replaced by many creative replications and variations [8], all made possible by easy to program prototyping toolkits [2][10]. This chapter is just a first attempt to reflect and generalize the ideas seen within our laboratory, and echoes other efforts at generalizing the capabilities of physical devices [13][14]. As toolkits with different capabilities for collaborative physical interfaces appear and as more ideas are developed, new categories and opportunities for these devices will certainly emerge.

I believe researchers in this area have just scratched the surface in their explorations of the design space for collaborative physical devices. We are only just beginning to understand how the surface features of these devices afford embodied interaction: their tangibility, their ability to work as an ambient public display, their ability to be easily shared, the opportunities they present for in-depth interaction, how they can act as surrogates for distant people and information, and the way their designs can be crafted to fit the physical environments. Of course, embodied interaction is much more than that, for the best designs will derive from an understanding of people's actual social practices within their environment. When we know that, we can still use these categories to design devices that best fit within a person's particular circumstances.

5. Availability of system videos and toolkit software.

Almost all systems described in this paper are viewable as videos available at:

- www.cpsc.ucalgary.ca/grouplab/phidgets/gallery/ or
- www.cpsc.ucalgary.ca/grouplab/papers/videos/.

Phidgets have been commercialized, and are available at www.phidgets.com. The collabrary is available at www.cpsc.ucalgary.ca/grouplab/collabrary/ although newer versions are now under development. Visit www.cpsc.ucalgary.ca/grouplab/ for updates.

6. Acknowledgements

The various projects would not have been possible were it not for the terrific toolkits built by Michael Boyle (the Collabrary) and Chester Fitchett (Phidgets). I am of course indebted to my many inspired students who gave me permission to use their projects in publications. Michael Boyle and Carman Neustaedter also provided useful feedback on earlier drafts of this chapter.

References

[1] Brave, S., Ishii, H. and Dahley, A. (1998) Tangible Interfaces for Remote Collaboration and Communication. Proc. ACM CSCW'98, 169-178, AMC Press.

[2] Boyle, M. and Greenberg, S. (2002) GroupLab Collabrary: A Toolkit for Multimedia Groupware. In J. Patterson (Ed.) ACM CSCW 2002 Workshop on Network Services for Groupware, November.

[3] Chang, A., Resner, B., Koerner, B., Wang, X and Ishii, H. (2001) LumiTouch: An Emotional Communication Device. ACM CHI Conference Companion, ACM Press.

[4] Clark, H., (1996). Using language. Cambridge University Press, Cambridge.

[5] Dourish, P. (2001) Where the action is: The foundations of embodied interaction. MIT Press.

[6] Fitzpatrick, G., Mansfield, T., Kaplan, S. Arnold, D., Phelps, T. and Segall, B. (1999) Augmenting the Workaday World with Elvin. Proc 6th European Conf Computer-Supported Cooperative Work (ECSCW'99), p431-435.

[7] Fogg, B., Culter, L., Arnold, P. & Eisbach, C. (1998) HandJive: A device for interpersonal haptic entertainment. ACM CHI'98, 57-64, ACM Press.

[8] Greenberg, S. (2003) Enhancing Creativity with Groupware Toolkits. Proceedings of the CRIWG ' 2003 9th International Workshop on Groupware (Sept 28 - Oct 2, Autrans, France), LNCS vol. 2806, 1-9, Springer-Verlag.

[9] Greenberg, S. (1999). Designing Computers As Public Artifacts. International Journal of Design Computing: Special Issue on Design Computing on the Net (DCNet'99), November 30 - December 3, University of Sydney.

[10] Greenberg, S. & Fitchett, C. (2001) Phidgets: Easy Development of Physical Interfaces through Physical Widgets. Proc ACM UIST Symp. User Interface Software and Technology, 209-218, ACM Press.

[11] Greenberg, S. and Kuzuoka, H. (2000). Using Digital but Physical Surrogates to Mediate Awareness, Communication and Privacy in Media Spaces. Personal Technologies, 4 (1), January, Elsevier.

[12] Greenberg, S. and Roseman, M. (1999). Groupware Toolkits for Synchronous Work. In M. Beaudouin-Lafon, editor, Computer-Supported Cooperative Work (Trends in Software 7), Chapter 6, p135-168, John Wiley & Sons Ltd..

[13] Ishii, H. and Ullmer, B. (1997) Tangible Bits: Towards Seamless Interfaces Between People, Bits and Atoms. ACM CHI'97, 234-241, ACM Press.

[14] Kaminsky, M., Dourish, P., Edwards, W.K., LaMarca, A., Salisbury, M. and Smith, I. (1999) SweetPea: Software Tools for Programmable Embodied Agents. ACM CHI'99, 144-151, ACM Press.

[15] Neustaedter, C. and Greenberg, S. The Design of a Context-Aware Home Media Space. Proceedings of UBICOMP 2003 Fifth International Conference on Ubiquitous Computing. 297-314, LNCS Vol 2864, Springer-Verlag.

[16] Kraut, R., Egido, C. & Galegher, J. (1988) Patterns of contact and communication in scientific collaboration. Proc ACM CSCW Computer Supported Cooperative Work, 1-12, ACM Press.

[17] Mankoff, J., Dey, A., Hsieh, G., Kientz, J., Lederer, S. and Ames, M. (2003) Heuristic Evaluation of Ambient Displays. ACM CHI'2003, 169-176. ACM Press.

[18] Patterson, J., Day, M. and Kucan, J. (1996) Notification Servers for Synchronous Groupware. Proc ACM CSCW'96, 122-129, ACM Press.

[19] Whittaker, S., Frohlich, D., & Daly-Jones, O. (1994). Informal workplace communication: What is it like and how might we support it? Proc. ACM CHI Human Factors in Computing Systems, 131-137, ACM Press.