Sensetable: A Wireless Object Tracking Platform for Tangible User Interfaces

James McMichael Patten
B.A. Interdisciplinary Studies
University of Virginia
May 1999

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences at the
Massachusetts Institute of Technology
June 2001

Author: **James Patten**
Program in Media Arts and Sciences
May 11,2001

Certified by: **Hiroshi Ishii**
Associate Professor of Media Arts and Sciences
Thesis Supervisor

Accepted by: **Stephen A.Benton**
Chair, Departmental Committee on Graduate Studies
Program in Media Arts and Sciences

Sensetable: A Wireless Object Tracking Platform for Tangible User Interfaces

James McMichael Patten

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences at the
Massachusetts Institute of Technology
June 2001

Abstract

Sensetable is a system that electromagnetically tracks the positions and orientations of multiple wireless objects on a tabletop display surface. The system offers two types of improvements over existing tracking approaches such as computer vision. First, the system tracks objects quickly and accurately without susceptibility to occlusion or changes in lighting conditions. Second, the tracked objects have state that can be modified by attaching physical dials and modifiers. The system can detect these changes in real-time. I present several new interaction techniques developed in the context of this system. Finally, I present several applications of the system, the most thoroughly developed of which is system dynamics simulation.

Thesis Supervisor: Hiroshi Ishii
Title: Associate Professor of Media Arts and Sciences

This work was supported by the Things That Think Consortium,
the Digital Life Consortium, and the Intel Corporation.

Sensetable: A Wireless Object Tracking Platform for Tangible User Interfaces

James McMichael Patten

Thesis reader
Robert Jacob
Associate Professor of Electrical Engineering
and Computer Science
Tufts University

Thesis reader
Jim Hines, Ph.D.
Senior Lecturer
MIT Sloan School of Management

Acknowledgements

I would like to thank the following people for helping make this work possible.

Jason Alonso
Mike Ananny
Angela Chang
Matt Conway
Dennis Cosgrove
Gordon Earlbacher
Jennica Falk
Phil Frei
Patrick Freuler
Tom Gardos
Paulo Goncalves
George Herman
Jim Hines
Ken Hinckley
Mary Murphy-Hoye
Moin Hussaini
Hiroshi Ishii
Rob Jacob
Nicole Justis
Jay Lee
Charlie Lertpattarapong
Benjamin Koo
Tom Malone
Cameron Marlow
Ali Mazalek
Katherine Moloney
Joe Panganiban
Gian Pangaro
Randy Pausch
Ben Piper
John Quimby
Ali Rahimi
Ben Recht
Casey Reas
Matt Reynolds
Jim Rice
Whittman Richards
Gustavo Santos
John Stankovic
Brygg Ullmer
Bill Verplank
John Viega
Paul Yarin

This thesis is dedicated to my mom, with thanks for her love and support.

Contents

1. Introduction

Tangible User Interfaces (TUIs) have attracted attention in the HCI community for their ability to take advantage of skills humans develop in the real world [18]. These interfaces often use groups of physical tokens to represent the digital state of a system. Users can interact with the system by manipulating these tokens. One goal of this approach is to provide a simpler and more intuitive mechanism for interacting with a computer by making aspects of the digital state of a computer system tangible. Two examples of TUIs are the musicBottles [17] and curlybot [12].

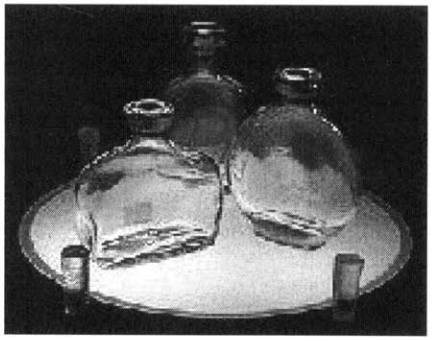

Figure 1-1: Professor Hiroshi Ishii's musicBottles project. Each bottle contains the sound of a musical instrument that is released when the bottle is uncorked.

A tabletop workspace with mechanisms for display and input is an appealing context for research in TUIs for several reasons. Such a space provides ample room to organize objects spatially, which can be an important part of thinking about the problem solving process [22]. Users can collaborate easily around such a space to solve problems using both hands. Finally, physical objects in this type of environment can be more than just input devices: they can become embodiments of digital information.

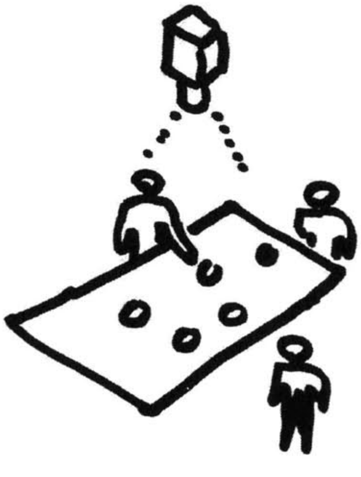

Figure 1-2: Collaborative use of an interactive tabletop workspace.

As a specific example, imagine that a group of executives in a semiconductor manufacturing company are sitting around a meeting table trying to develop a manufacturing plan for the next year. They need to decide which products the company should be making, and the amount of each product they should produce per month. Instead of doing the various calculations involved in the process on a wall-mounted whiteboard (a process which might take days or weeks to complete), the executives manipulate a series of physical objects on the meeting table itself. These objects represent the various parts of the company's supply chain: the factories, warehouses, suppliers etc. The objects each have dials and switches which the executives can use to adjust parameters corresponding to each object, as shown in figure 1-4. A computer embedded in the meeting table senses what the executives are doing to the objects. It detects when they are moved on the table, when their buttons are pressed, when their dials are turned, etc. These actions control parameters in a computer simulation of how the company works. A projector on the ceiling projects information onto the table about how the simulation is affected by these changes. Information about specific parts of the business appears on and around the corresponding physical models on the table. The executives experiment with ways of changing how their business works by manipulating the objects on the table. Through these experiments, they begin to develop an intuition for how certain specific changes in their business will affect the business as a whole. The tangible interface to the simulation on the meeting table provides a more intuitive, simpler way of controlling the simulation than GUI based approaches. This in turn allows the executives of the company to learn about the behavior of their company more quickly and more thoroughly.

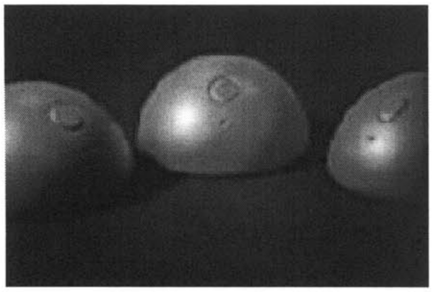

Figure 1-3: Phil Frei's curlybot project. This educational toy records and plays back motion through the same physical object.

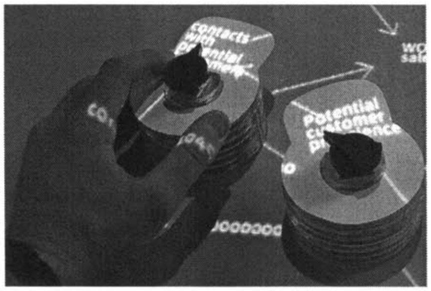

Figure 1-4: A user modifies a parameter in a system dynamics simulation using an object tracked by Sensetable.

The notion of an interactive display surface that is able to sense the positions of objects on top of it has been discussed in the HCI literature for many years [9,36,42]. However, the typical approaches to this object-tracking problem each have some limitations. Computer-vision-based approaches can have problems with robustness due to the need for controlled lighting conditions. [39] Tracking latency can also be an issue when objects are moved around in the sensing space. Magnetic tracker based approaches, such as those made by Polhemus and Ascension require that wires be attached to the objects being tracked [32].

To support our research in interactive tabletop surfaces, I decided to develop a new platform, called Sensetable, which aimed to improve upon existing methods in two ways. First, I wanted the platform to provide accurate, low-latency wireless tracking of 6-10 objects on a flat surface. Second, in order to explore new interaction techniques I wanted to allow users to modify the tracked objects (using dials or "modifier" tokens as shown in figure 1-5), and to map these physical changes to changes in the application running on the platform. All of the technologies I investigated for this platform employed some form of electromagnetic sensing to determine the positions of objects.

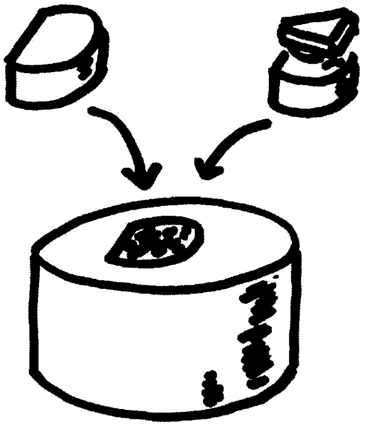

Figure 1-5: A socket on top of a Sensetable puck, into which one can place dials and modifiers.

After considering several alternatives, I decided to implement the first prototype by extending commercially available sensing tablet technology. After completing the first prototype, I began developing applications and exploring interaction techniques using the system. After observing the strengths and weaknesses of the first implementation, I began developing two more hardware implementations to experiment with overcoming the weaknesses of the first platform in different ways. In the next chapter I describe previous research related to the Sensetable project. In the third chapter, I describe the implementation of the three Sensetable prototypes. I continue by presenting the interaction techniques and applications I have developed on top of Sensetable. Finally, I present some conclusions and plans for future work.

2. Related Work

In this chapter I discuss some supporting research related to the Sensetable platform. This research includes several related projects and technologies involving interactive surfaces, as well as some experiments about how humans use various types of physical interfaces to computers. Finally, I discuss some principles of tangible user interface design as they relate to the Sensetable project.

2.1 Related Experiments and Psychological Theory

Some work has been done to understand different ways that spatial arrangements of objects can be used to help us think. Work by Kirsh [22] explores a variety of ways that people use the space around them while solving problems. Kirsh divides actions taken in a problem solving process into "epistemic" and "pragmatic" actions. Epistemic actions are those which help one think about what action to take to solve a problem. Pragmatic actions are those which are taken to actually solve the problem. For example, if one wanted to listen to some music, one might flip through a catalog of CDs to determine which one to play. This would be epistemic action. Once one had decided upon a CD to play, one would then take that CD, and put it into the player, and then press the play button. This would be pragmatic action. Epistemic actions are a component of a problem solving strategy called a complementary strategy. Kirsh defines a complementary strategy as "any organizing activity which recruits external elements to reduce cognitive loads." [21] An example complementary strategy is grouping coins into denominations while counting them to increase the speed and accuracy of the counting process. Kirsh's work shows that complementary strategies can lead to performance gains even in tasks which do not inherently require the environment to be changed in any way.

A significant part of Kirsh's work deals with organizing objects in space to help one complete a task. He explains several ways in which organizing things spatially can help people increase their performance on a task. Spatial arrangements can simplify choice, simplify perception, or simplify mental computation. An example of simplifying choice is sorting a list of papers in an "in box" in order of priority. When dealing with these papers, one can simply take the one off the top and deal with it, without having to carefully consider the ordering of priorities after dealing with each item in turn. An example of simplifying perception is sorting pieces of a jigsaw puzzle into similar categories based on whether they are an edge piece, a piece of a certain color, etc. It is easier to visually perceive the differences between similar pieces when they are close to each other, rather than being among a group of dissimilar pieces. An example of simplifying mental computation is sorting items into different categories based on attributes which are not immediately apparent through visual perception. One might sort a group of books into fiction and non-fiction categories. Once one had grouped them, one would not need to remember whether each book was fiction or non-fiction [21].

This work has interesting implications for the Sensetable project. If complimentary strategies help people solve problems faster, and one common complimentary strategy is organizing this spatially, than an interface which lets people quickly and easily organize things spatially (such as Sensetable) should help them solve problems faster. As well, Kirsh's work suggests that in a system like Sensetable, there should be ways to manipulate the physical objects which are not interpreted by the computer. The user can employ these uninterpreted degrees of freedom in a complementary strategy during the problem solving process.

Zhang presents a study which shows that the nature of the objects used in problem solving tasks can dramatically affect how people think about the tasks and how long the tasks take to solve [46]. He compares the time required to solve two variants of the "Towers of Hanoi" puzzle. The variants have the same rules as the standard puzzle. However, one uses oranges of varying sizes instead of the rings in the standard puzzle; the other uses coffee cups. Zhang found that the puzzle involving oranges took more than twice as long as the coffee cups puzzle to complete, with six times as many errors [45].

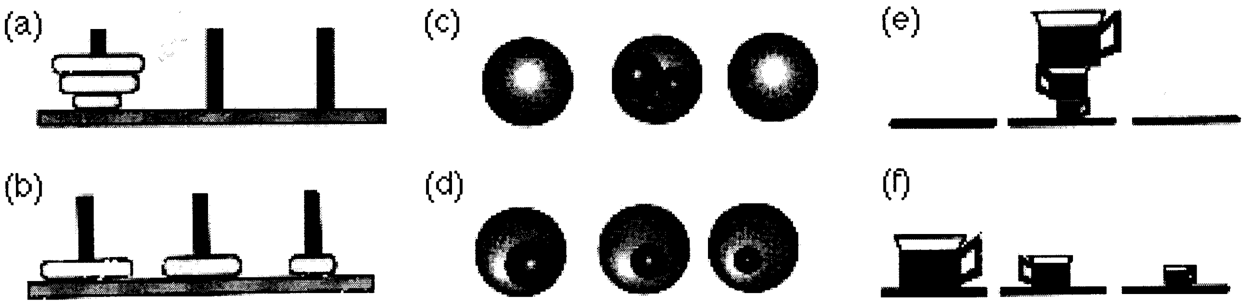

Figure 2-1: Zhang's variants of the Towers of Hanoi puzzle

Zhang's work emphasizes the impact that physical affordances can have on a problem solving task. In the context of the Sensetable project, this work suggests that the physical affordances of the objects on the Sensetable surface are very important, and that different physical forms could be applied to different problem solving tasks to make the system easier to use.

A variety of researchers have recognized that the ability to use two hands while interacting with an interface can lead to significant performance improvements. This holds true when the two hands are completing unrelated tasks, as well as when they are acting cooperatively. [4, 10, 15] This work suggests that allowing for two handed interaction should aid the process of manipulating objects in a problem solving task. Thus, Sensetable should provide for easy two-handed interaction.

In addition to this work about solving problems using spatial information, there is also a variety of work on how people remember and use spatial information about their environment. Malone asked ten office workers to locate items in their offices in order to understand the different strategies people use for filing and retrieving information [26]. While his results suggested that office workers, particularly those with neat offices, were good at finding documents within them, more formal work on this question has suggested that it can be difficult to rely on location information alone for recall [8, 24, 28]. Dumais and Jones found that retrieving documents by name was more effective than using spatial information for retrieval [8]. Lansdale argues that memory of location can be quite poor in cases where documents are not organized according to some logical structure. In cases where a structure is imposed, subjects can use it to help determine the location of documents, and thus their performance at recalling location improves [24].

On the other hand, Mandler et al. have compared the performance of subjects at recalling object location when they are intentionally trying to remember location and when they are not. They found only a small decrease in recall performance when subjects were not told to remember object location. From this they concluded that much object location information is encoded automatically [27]. However, Naveh-Benjamin responds that location information is in fact not encoded automatically when subjects are observing a spatial configuration rather than modifying it themselves [28].

Despite the disagreement in the literature about the utility of spatial information, recent work by Robertson et al. on the Data Mountain system suggests that spatial memory can be used to reliably improve performance in a task involving the retrieval of web documents represented by icons on the screen [34]. In the Data Mountain system, users employ a mouse to place web pages on the side of a "mountain" displayed on the computer screen in 3D. Robertson et al. found that when users were presented with a title, summary and thumbnail image of a document, they could retrieve it more quickly and with fewer errors with the Data Mountain system than with the Internet Explorer^{TM} Favorites mechanism.

2.1.1 An Experiment on the Use of Space

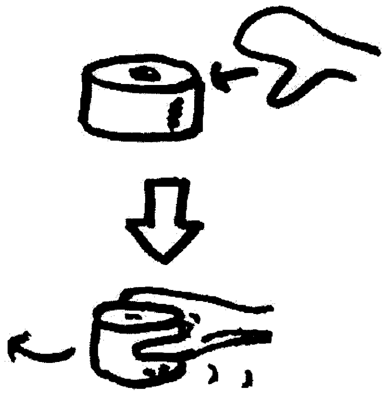

Figure 2-2. A two stage input model for TUIs. First, one acquires the physical object, and then manipulates it as desired.

Given this research on how people use spatial information to help them remember things and solve problems, before beginning the Sensetable project I decided to explore the differences in how people use graphical and tangible user interfaces to organize things while solving problems. A variety of research suggests that TUIs provide both quantitative and qualitative benefits over GUIs for some applications [9,15,39]. However, little work has been done to explore how people use space to solve problems in GUIs and TUIs.

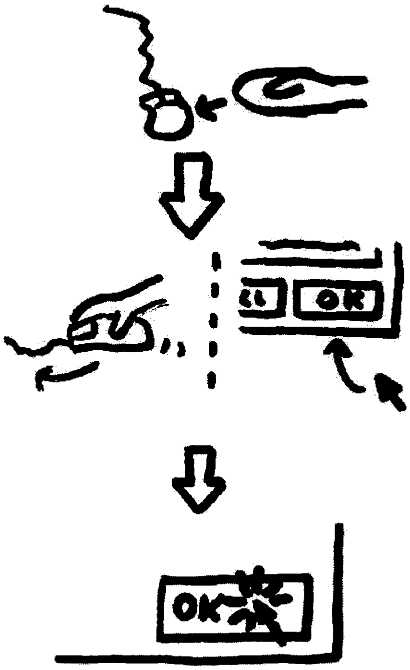

Figure 2-3: The three stage input model for GUIs. First, one acquires the mouse. Second, one moves the mouse cursor to the graphical item of interest. Finally, one manipulates it as desired.

One difference between TUIs and GUIs is the ability of users to place a physical object or group of objects in a certain state faster than analogous operations can be performed on the screen [9]. For example, users can sort a collection of physical objects with their hands faster than they can sort a collection of icons on the screen. Several factors seem important here, including the ability to move physical objects with both hands, the ability to move more than one object with each hand, and the instant haptic feedback from physical objects that lets you know you have indeed grasped them. The models for GUI and TUI input also illustrate another key difference. The three state model for graphical input [5], shown in figure 2-3, divides the process of manipulating an object on the screen into these steps: First, one must grasp the physical input device, such as a mouse. Next one must use this device to acquire the graphical object to be manipulated. Finally, one can manipulate the graphical object as desired. In the physical world, a two state model is more appropriate [9], as shown in figure 2-2. One simply acquires the physical object to be used, and then manipulates it as desired.

However, I believe the differences between GUIs and TUIs go much further than issues of speed. Because the nature of interaction with TUIs is fundamentally different from that with GUIs, I think that their roles in epistemic action may differ. Understanding this potential difference is important for two reasons. First, it may help us develop a better understanding of which applications are best suited for specific TUI platforms such as Sensetable. Second, a thorough knowledge of how space is used differently in GUIs and TUIs may suggest design considerations for TUIs of which we are currently unaware.

To explore the differences between GUIs and TUIs in terms of epistemic action, I conducted an experiment in which I asked subjects to read a group of news summaries and think about how the summaries related to each other. For this task, some subjects used a TUI while others used a GUI. I designed the two interfaces to be as similar as possible, the GUI using on-screen icons to represent the summaries, the TUI using wooden blocks. To isolate the effects of spatial memory in the experiment, I made the tokens visually identical. The subjects accessed the summary associated with each block or icon by placing the token into a reader. While reading, most subjects moved the tokens around to help them think about how the summaries were related to each other. After the subjects finished reading, I interviewed subjects about their spatial layout strategies and measured their ability to remember the token with which each news summary was associated.

I observed the following:

- Only TUI subjects used layout strategies which involved positioning tokens based on location within the space as a whole, rather than positioning relative to other tokens in the space. I call this strategy reference frame based positioning.

- Subjects who incorporated this reference frame based positioning scheme in their placement strategy were able to recall the associations between tokens and articles better than others.

- TUI subjects performed better at the recall task than the GUI subjects, remembering the locations of an average of 5 blocks, compared with 3.5 for the GUI case.

2.1.1.1 Description of Experimental Task

Subjects were asked to put themselves in the position of a newspaper editor who had to read ten short news summaries. Each summary was a 100 to 150 word excerpt from a top story in a mainstream online newspaper. They were told to take as much time as necessary to read all ten, and to look at each summary as many times as they wished. They also were told to expect a series of questions about how the summaries could be used in a newspaper afterward. I stated that subjects might want to consider how the summaries were related to each other, what the implications of each summary would be, and which readers would be interested in each summary, emphasizing that there were no correct answers. As I was interested in understanding how subjects' organizational schemes would develop and evolve over the course of the experiment, I was careful not to suggest any particular classification scheme for the summaries.

The subjects were divided into two groups: half of the subjects used a TUI to access the series of news summaries; the other half a GUI. The TUI consisted of a group of visually identical wooden blocks. When a block was placed in a reader device attached to the bottom of a computer monitor, the summary corresponding to that block appeared on the screen directly above it, as in figure 2-4. The GUI subjects accessed the same news articles by dragging and dropping an icon into a reader area displayed on the screen. When an icon was placed inside of this reader as shown in figure 2-6, the summary corresponding to that icon was displayed next to the reader.

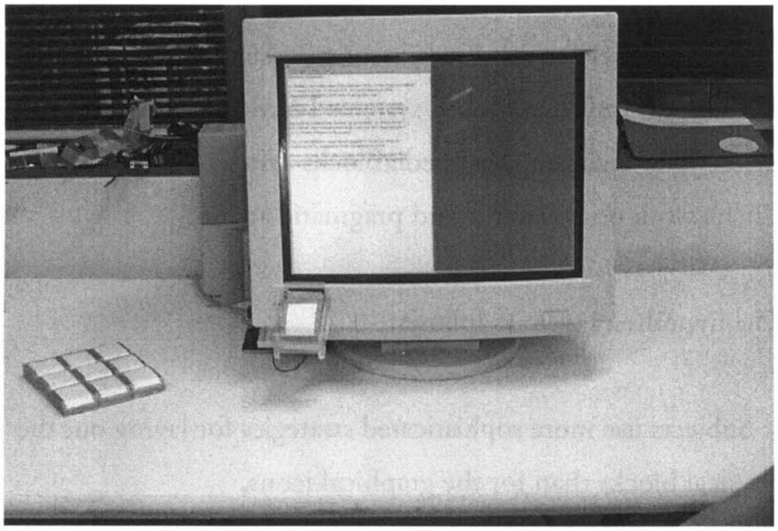

Figure 2-4: A block is in the reader, while the other nine are in their initial positions.

While the subjects were reading the summaries, I observed where they placed the blocks on the desk or the icons on the screen. Immediately after a subject indicated that he or she was finished, he or she was asked to indicate which icon or block corresponded to each summary. The subject was prompted with the title of each summary in random order. The purpose of this task was to measure how well the layout strategy each subject used helped him or her remember with which summary each token was associated. After this task was complete, the subject was interviewed about how he or she organized the blocks or icons during the task. All subjects were asked about organizational strategies using the same set of scripted questions. The organizational strategies described in the "results" section come from the subjects' reports about the strategies they employed. The final configuration of the blocks or icons was also recorded.

2.1.1.2 Experimental Hypotheses

The hypotheses for this experiment were suggested by the various physical token-based systems I have explored in the Media Lab, including the mediaBlocks system [37], and by Kirsh's work on epistemic and pragmatic action.

The hypotheses were as follows:

1. Subjects use more sophisticated strategies for laying out the physical blocks than for the graphical icons.

2. Subjects using the physical objects more accurately remember which token each summary is associated with than those who use graphical icons.

2.1.1.3 Subjects

Thirty-six subjects (18 males, 18 females) were paid $10 each to participate in the experiment. The subjects ranged from 18 to 49 (mean 26.7) years old, and reported using a computer between 2 and 40 (mean 21.9) hours per week. Despite this variation in weekly computer usage time, subjects reported using them for quite similar tasks, including electronic mail, word processing and accessing websites.

2.1.1.4 Experimental Procedure and Design

In the TUI case, ten 2" x 2" x 0.75" wooden blocks were used to represent the news articles. Each block had a piece of paper on top which was used to cover up markings on the top of some blocks, to make them appear as visually similar as possible. Each block contained a digital identification tag and two strips of fuzzy conductive material on the bottom, as used in the mediaBlocks system [37]. The content of a block was accessed by inserting it into a reader device, which was attached with Velcro to the bottom left corner of a 21" computer screen.

The reader was designed so that the weight of the blocks would be enough to ensure electrical contact was made as the blocks were placed in the device. It could only accommodate one block at a time. The reader device only allowed wooden blocks to be placed into it if the diagonal face of the block was facing toward the subject. This ensured that proper electrical contact would be made with the block.

I demonstrated the use of these blocks to the subject, and then asked the subject to try using them. All subjects were able to use the blocks correctly on the first try, and reported no difficulty in understanding how to use them. When a block was placed into the reader, conductive strips inside of the reader connected with those on the block so the digital identification of the block could be read. Based on this identification number, the news summary corresponding to that block was displayed on the left half of the screen, directly above the reader device. The right half of the screen was not used in the TUI.

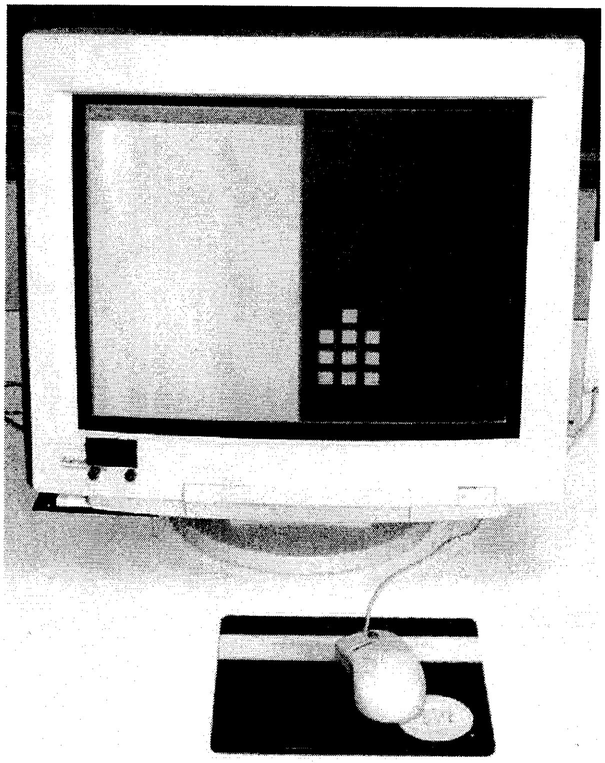

Figure 2-5: The GUI task with icons in their starting positions.

The task began with the blocks grouped to the left of the display as shown in figure 2-4. No items were on the desk except the monitor and the blocks. Subjects were told that they could leave blocks in any location on the desk when they were not in use.

In the GUI case 10 visually identical 45x45 pixel icons were used to represent the news summaries. These icons were constrained to the right half of the screen in an area measuring 640x1024 pixels, while the summaries themselves occupied the left half of the screen. The screen was divided in this manner to prevent the text of the news summaries on the screen from occluding any of the icons. The content of these icons was viewed by dragging the icons into a graphical reader area at the top of the screen. As in the TUI case, subjects were told that they could leave icons in any location when they were not in use.

Software was used to constrain the icons so that only one icon could be placed in the reader area at a time, to maintain consistency with the physical case. Users could not double-click on the icons to open the news summaries as one can in many common GUIs. I wanted to understand how users would choose to arrange the icons if they had to develop some sort of strategy for doing so. Allowing users to double-click to open them would have made it possible to view each article without moving the corresponding icon. I suspect that in this case subjects would have done quite poorly at recalling which icon corresponded to each summary, as a similar experiment revealed quite poor recall rates [28]. Instead, I relied on the drag-and-drop metaphor which is commonly used in today's GUIs, and which also maintained consistency with the TUI condition of the experiment.

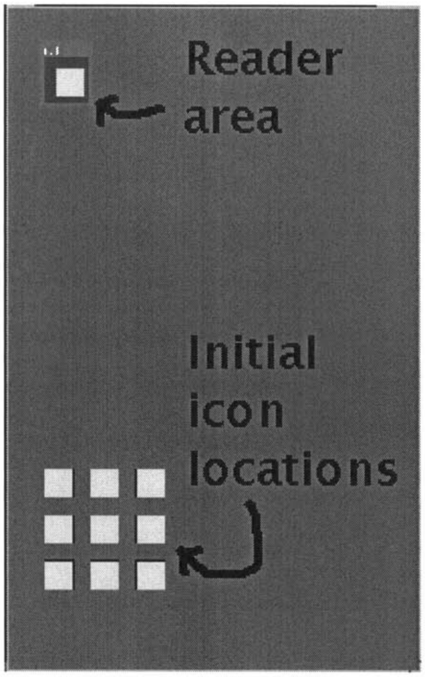

Figure 2-6: The news summary associated with an icon is displayed when the icon is moved to the reader area.

Subjects participating in the GUI case were shown how to use the interface, and then were asked to try it themselves. Only one subject had difficulty using the interface at first, and after I explained that the left mouse button rather than the middle one had to be used to drag the icons, this subject did not have difficulty.

2.1.1.5 Experimental Design Considerations

Both GUIs and TUIs have a variety of characteristics that come "for free" which would greatly improve performance in tasks such as this one. For example, the icons on the screen could be annotated with short text labels which describe the summaries. The icons themselves could contain an image relevant to the summary. Summaries could be structured hierarchically in "folders" on the screen. In the TUI case, users could draw annotations with erasable pens on the tops of objects used to represent data. The three dimensional nature of the objects could be used in a variety of ways, such as stacking the objects on top of each other or storing them in different locations in the physical environment. In addition, graphical information about the physical objects in a TUI could be projected either from above [39] or below [36] the surfaces upon which they rest. In this experiment, I tried to take out as many of these factors as possible to focus on the issues of space so that I could begin to understand the differences between GUIs and TUIs in this regard. I insured that the objects a subject used, whether physical or graphical, looked as similar as possible, and that subjects had the same amount of space to work with while rearranging the objects in proportion to the size of the objects themselves.

Because the experiment involved a surprise spatial recall task, I used a between-subjects design. After performing one condition of the experiment, subjects learned that the experiment was focusing on their spatial organization strategies rather than their approaches to newspaper editing. Pilot experiments suggested that subjects did not focus on the task of organizing the articles for a newspaper when they knew that a spatial recall task would follow. Rather they focused on memorizing the article locations according to some mnemonic. For example, one pilot subject alphabetized the stories based on their titles, treating the task as a memory task rather than an organization task. I was more interested in organizational strategies based on the content of the articles than simple strategies such as alphabetization. I expected that a strategy based on the content of the articles would have to evolve over time as the subject read more of the articles, where a strategy such as alphabetization would not. I felt that the process of adapting strategies during the experiment was important to explore, because strategies might evolve differently in the TUI than in the GUI.

2.1.1.6 Limitations of the Experiment

While I controlled for a variety of factors between the TUI and GUI conditions of the experiment, this did not include the extra rotational dimensions available in the physical interface. The wooden blocks were shaped such that the front and back were easily distinguished, so users would insert them correctly into the block reader. While it was possible for a subject to use the rotation of the blocks on the desk to encode information about them, I anticipated that subjects would tend to keep the front of the blocks facing toward them, so that they could be inserted quickly and easily into the reader. In practice, no subjects reported using the rotation of the blocks to encode any information.

In addition, I did not control for the organizational strategies that subjects were familiar with, or chose to use in the experiment. In one sense this was desirable because it helped us to understand what types of strategies subjects were inclined to use given the skills at their disposal. However, this decision also contributed to within-group variability, because the organizational strategies subjects used seemed to be an important factor in recall performance. While this limitation would not have been an issue in a within-subjects design, I believe that when coupled with the surprise recall task, a within-subjects design could have introduced more severe limitations. As discussed in the "Design Considerations" section, pilot subjects changed organization strategies when expecting a recall task. I was concerned that this change of strategies between the two trials would add noise to the data.

2.1.1.7 Results of the Experiment

Some TUI subjects employed spatial encoding techniques which relied on the position of the blocks within an external reference frame, while GUI subjects did not. TUI subjects who used this reference frame based positioning strategy did better on the recall task than those TUI subjects who did not. As well, TUI subjects performed better than GUI subjects at the recall task overall. I discuss the findings in detail below.

Spatial Arrangement Strategies

After the memory recall tasks, I asked subjects to describe their spatial layout strategies. Three GUI subjects reported that they adopted a layout strategy after reading only one or two stories, but later their arrangements of icons became less and less consistent as they found that some of the remaining stories did not fit well into the organization scheme they had devised. Because they did not adopt a new classification scheme after finding that their initial one was not sufficient, when they were done reading the articles they found the organizational structure of little assistance when remembering which story each icon contained.

In contrast, some TUI subjects appeared to frequently adopt new organizational schemes, or adjust old ones, in order to accommodate new stories. TUI subjects would often re-read the first three or four stories and rearrange them on the desk before reading the remaining stories for the first time. Other TUI subjects would read all of the articles once first, and then rearrange them on the desk by quickly checking the title of each one in the reader, and then moving it to an appropriate location on the desk.

Interviews of subjects revealed that three basic types of spatial encoding mechanisms were used, though at times they were used in concert with each other. These strategies were:

Grouping – Subjects would place summaries with some property in common together in the space. e. g. Summaries only of interest to local audiences, or summaries about violence.

Ordering – Subjects would rank summaries or groups of summaries along an axis, such as how the summaries made the United States look in the eyes of other countries.

Reference frame based positioning – Subjects would place an object by itself in the space, in a location which meant something specific to that object, regardless of the spatial arrangements being used for other objects. For example, one TUI subject placed a summary about fires in the western United States far to the left of other summaries to represent that it dealt with the western part of the country. Another TUI subject reported placing an article about heart problems on the desk directly in front of his heart and placing a summary about arms sales directly in front of his arm, taking advantage of the dual meaning of the word "arms."

Subjects in both conditions of the experiment employed grouping and ordering strategies. The results are summarized in figures 2-9 and 2-10. Eight GUI subjects used a grouping strategy. Seven of these eight used grouping exclusively, while the other one also sorted two of the groups' contents by importance from left to right. In contrast, ten TUI subjects used grouping, but seven of these ten employed it in combination with another strategy. All five subjects who used reference frame based positioning also used grouping.

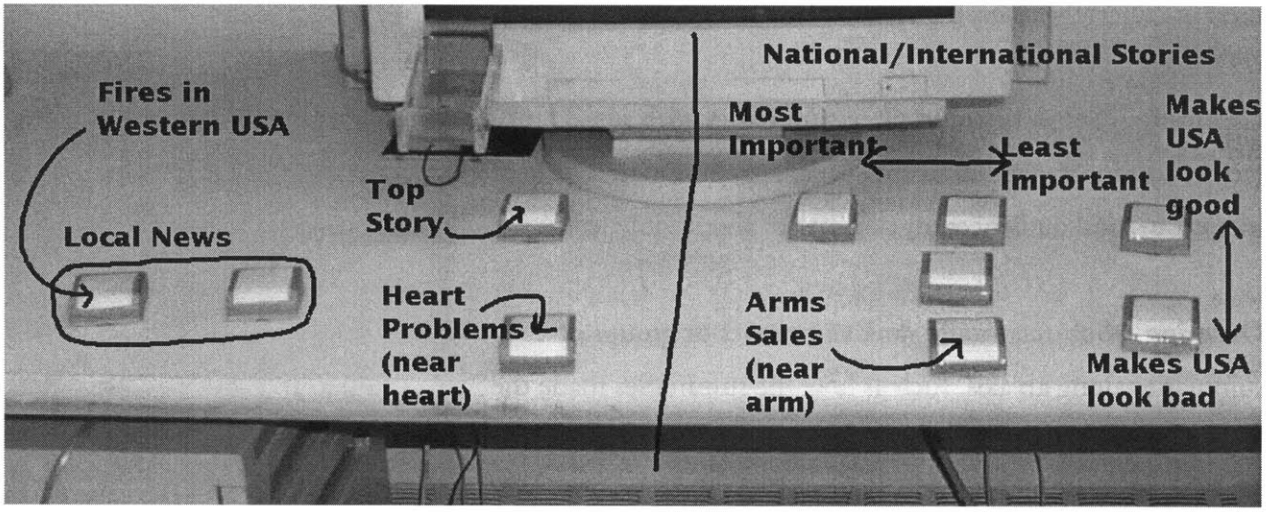

Figure 2-7: Example final position of the blocks after the TUI task. Note the use of grouping, ordering, and reference frame based positioning.

Subjects grouped the summaries into categories such as "front page" "world news" and "local news" or "politics," "human interest" and "other." Subjects used ordering schemes based on various parameters including how interesting the summaries were, or the number of people they affected. Figure 6 shows a typical final layout of icons for a GUI subject. None of the subjects in the GUI case used a layout strategy which included reference frame based positioning. However, five TUI subjects did use such a strategy. This reference frame based strategy seemed to help subjects improve recall rates as well. The mean recall rate of subjects who incorporated this strategy was 8.2 (std. dev. 2.05) which is in contrast to the mean recall rate of 3.8 (std. dev. 2.05) for TUI subjects which did not use reference frame based positioning. Note that this mean is quite similar to the overall mean for GUI subjects. Figure 5 shows the final position of the blocks for a subject who used this reference frame based positioning strategy. The high standard deviation in the TUI data is due to the difference in performance between subjects who employed reference frame based placement strategies and those who did not. The correlation between the use of a reference frame based positioning scheme and performance in the recall task for the eighteen TUI subjects suggests that a reference frame based positioning strategy is an effective method for representing information using spatial layout in TUIs.

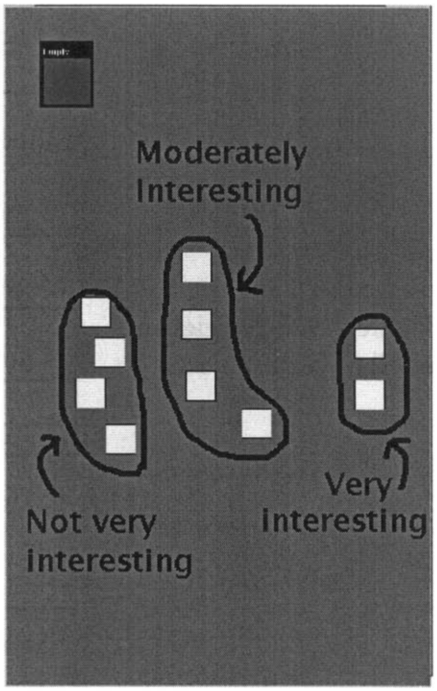

Figure 2-8: Positions of icons after the task. This subject only used a grouping strategy, though some GUI subjects also employed ordering approaches.

In both the TUI and GUI conditions, there were some subjects who encoded little or no information into the spatial arrangement of the tokens. Three TUI subjects and three GUI subjects placed each token very near where it was before they began reading it, in essence not using any spatial organization strategy at all. In addition, three TUI and three GUI subjects simply kept the tokens they had already read separate from those they had not. Finally, two GUI subjects and one TUI subject sorted the icons according to the order in which they had read them.

Figure 2-9: Strategies and recall of GUI subjects

Strategy	**Num. Subjects**	**Recall Rate**
Little/ no organization	8	3.38
3 groups, no ordering	6	4.16
3 groups, ordering within 2	1	3
4 groups, no ordering	1	2
Only ordering	1	3

Figure 2-10: Strategies and recall of TUI subjects

Strategy	**Num. Subjects**	**Recall Rate**
Little/ no organization	7	4.14
3 groups, no ordering	1	1
3 groups, ordering within all	1	8
4 groups, no ordering	2	3.50
Only ordering	1	4
4 groups, ordering within 1	1	0
Reference frame based positioning along with 1-4 groups	5	8.20

When asked about the layout of the objects, subjects who employed little spatial organization gave several explanations. One TUI subject said that "accessing the stories from the blocks was so easy that I felt no compelling need to organize them." A GUI subject said she was "storing them more mentally than spatially." Finally, a TUI subject mentioned that he was expecting to be quizzed on the details of the news summaries, so he had focused on memorizing them rather than on thinking about how the summaries might be used in a newspaper

Subjects in the TUI case remembered the locations of an average of 5.0 blocks (std. Dev. 2.85). With an outlier removed as discussed below, subjects in the GUI case remembered the locations of 3.47 blocks on average (std. Dev 1.23). Figure 4 shows this result. The bars represent standard error.

On the GUI portion of the experiment, one subject correctly recalled eight of the news story locations, placing him 2.68 standard deviations above the mean for GUI subjects. This is above the critical value of 2.50 (5% confidence interval) for a single outlier in a normally distributed sample of 18, as discussed in [1]. In a telephone conversation with me 11 days after participating in the experiment, this subject was able to correctly recall the organization strategy he used in the task, complete with the location of the groups of icons on the screen and the stories associated with each group. Because of this subject's demonstration of this superb memory ability and his large deviation from the mean GUI score, I separated this datapoint in the remainder of the statistical analysis. This subject's organizational strategy involved grouping the stories into four categories. He did not report using any techniques

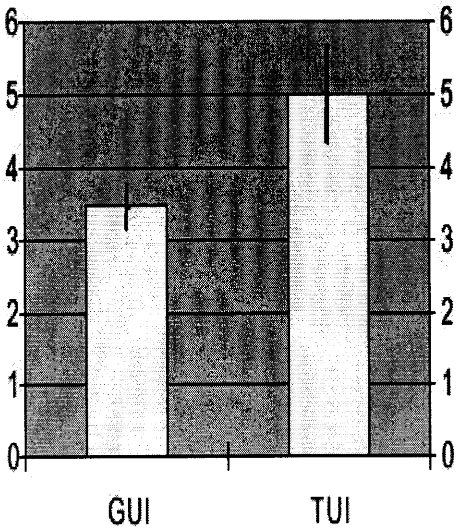

Figure 2-11: GUI and TUI object recall rates

different from the usual GUI grouping strategies described below.

A one-way ANOVA indicated that the difference in performance between GUI and TUI subjects was statistically significant ($p < 0.05$, $F(1,34) = 4.16$).

2.1.1.8 Discussion of the Experiment

I observed that some TUI subjects employed reference frame based positioning effectively in the experiment. I also observed that TUI subjects performed better than GUI subjects at the recall task. This section contains some possible causes and implications of these results.

In the Results section, I reported that TUI subjects seemed more likely to change an organizational strategy to fit new stories as they read. One possible explanation for this difference is that it is easier to move tokens around in a TUI than in a GUI. With a TUI, subjects can manipulate objects with both hands at the same time. They can also slide groups of objects on the desk with one hand. As well, TUI users get instant, haptic feedback when they touch a physical token.

The models for GUI and TUI input suggest another key difference in usability. In the three-state model for graphical input [5], one must first grasp the physical input device, such as a mouse. Next, one must use this device to acquire the graphical object to be manipulated. Finally, one can manipulate the graphical object as desired. In the physical world, a two-state model is more appropriate: one simply acquires the physical object to be used, and then manipulates it [9]. The extra step required for this task in a GUI suggest that more time and mental effort is typically required to perform this task.

The separation between the mouse and the GUI screen may also make interaction with a GUI more difficult. When a user moves an icon on the screen with a mouse, the mouse itself moves in a horizontal plane, while the cursor moves in the vertical plane of the screen. MacKenzie and Iberall have pointed out that when the visual map and the proprioceptive map are not aligned, performance in object manipulation tasks can degrade [25].

Another issue that may complicate the process of manipulating objects in a GUI is the act of picking the mouse up off of the mouse pad. With most mice, the mouse pointer is only moved when the mouse is in contact with the surface beneath it. This means that just because the mouse pointer is at one side of the screen, the mouse itself and the hand guiding it are not necessarily at the corresponding corner of the mouse pad. Because the positions of the mouse cursor and the mouse itself are seldom correlated, the user cannot employ the position of the physical mouse relative to his or her body to help remember the positions of things on the screen.

These differences in interaction qualities between GUIs and TUIs may make users more likely to involve TUIs than GUIs in epistemic action. Epistemic action is a way to help offload thinking and memory tasks from the mind to the external world. In order for epistemic action to be worthwhile in a problem solving task, one must save more mental effort by encoding information in the physical world than one expends in the encoding process. Thus, the easier it is to manipulate objects in a problem solving task, the more frequently it will make sense to encode information in those objects to simplify the problem.

Another reason why TUI subjects may perform better at the recall task than GUI subjects is that people may be better at remembering where they have placed physical things than graphical icons, regardless of the organizational structure that they place them into. One aspect of this may be motor memory. While motor memory may be used to one's advantage in a TUI, the motions required to manipulate an object in a GUI change each time the user picks up the mouse and recenters it on the mouse pad, so memory of past actions seems less useful.

Another issue to consider is that one must pay explicit attention to the locations of nearby objects when moving things in the physical world. Thinking about avoiding other objects while placing an object in the physical environment may help the user remember location better, because more attention must be directed to the locations of nearby objects [Whittman Richards, personal communication]. To move an object on a desk, one must either lift the object off of the desk or slide it carefully around other objects to avoid disturbing their positions. In most cases, GUIs do not exhibit this behavior.

The use of reference frame based positioning in the TUI case seems to be important as well for developing a coherent spatial arrangement of the blocks. There are several reasons why this placement strategy may be more appropriate for TUIs than for GUIs. The first issue is that the visual and physical properties of objects are much more varied in the context of TUIs than in GUIs. Even in this experiment, in which I removed extraneous objects from the desk area which conceivably could have been used in a spatial organization scheme, one subject used the context clues provided by the computer monitor, by placing a block near its base to help him remember to put the corresponding story in the front page of his newspaper.

The human body can be a useful reference frame for TUIs as well. When a user places an object to his or her left in physical space, from the user's perspective this object is in a very different position from an object in front of the user. The center and right side of a computer screen are close together in comparison. With a standard desktop monitor, icons spread about the screen are all still in front of the user. This makes it difficult to use the position of the objects relative to the body to differentiate between them.

Because using spatial information seems to be easier in TUIs than in GUIs, TUIs may afford Kirsh's epistemic action to a greater degree than do GUIs. This conclusion is supported by the decisions of several of the GUI subjects in the experiment to abandon or not develop their spatial organization strategies when their original strategy did not appear satisfactory. In short, TUIs may make it easier for us to think about some problem solving tasks in ways that GUIs do not.

The differences between TUIs and GUIs observed in this experiment suggest some design considerations for TUIs. First, it can be useful for an interface to provide ways for the user to move and organize objects without these operations being interpreted by the TUI. Consider an interface in which a user places objects on a rack to perform an operation. A designer might choose to not have the system interpret the order of the blocks on the rack, so that the user could manipulate the order to help keep track of the task he or she was trying to accomplish.

As well, physical scale can be important in making a more usable TUI. Because GUI screens are so small relative to the size of our bodies, it is difficult to employ the reference frame our body provides to help us organize groups of objects in a GUI. TUIs which employ a small physical structure as a central part of the interface can fall prey to the same problem. However, TUIs which have a larger physical size can take advantage of the spatial reference frame of the user.

In the context of the Sensetable project, this experiment supports the hypothesis suggested by the work of Zhang and Kirsh, which is that Sensetable may help users organize things in space to solve problems more effectively than systems using a graphical user interface. As the Sensetable prototypes mature, one interesting area of research is continuing to explore the differences in the use of Sensetable and a graphical user interface in the context of a specific, real-world application.

2.2 Related Systems and Technologies

In addition to investigating the psychological issues relating to tabletop interaction surfaces described above, I began exploring related systems which other researchers had built before developing the first Sensetable prototype. Wellner's Digital Desk [42] system, shown in figure 2-12, introduced the concept of an interactive tabletop that was both physical and digital. Users interacted with digital content in the system by "touching" projected graphical representations on the desk. The system detected these touches using a camera and microphone. Interactions such as making calculations using a calculator projected on the desk were possible using this system. [43]

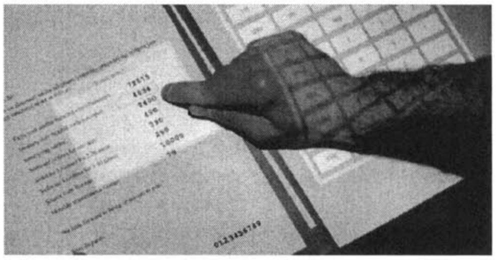

Figure 2-12: Interacting with a physical piece of paper using a virtual calculator on the Digital Desk

The Bricks project [11] pioneered the use of graspable handles for manipulating digital objects directly using two tethered Ascension Flock of Birds(tm) trackers. This system, shown in figure 2-13, illustrated some of the powerful things one could do with a platform that tracked objects in real-time, and merged input and output into one physical space. However, this system was limited in that it only provided two physical objects for the user to manipulate, and these objects were connected to the computer with wires, as shown in figure 2-14.

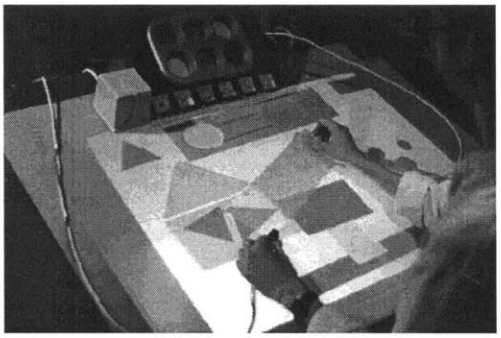

Figure 2-13: The GraspDraw application of the Bricks system.

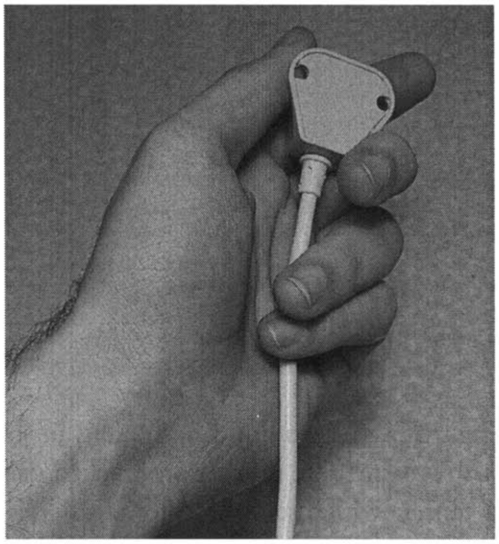

Figure 2-14: An Ascension Flock of Birds 6 degree-of-freedom magnetic tracker.

The metaDESK [36] system built on the ideas presented in the Bricks system by demonstrating the use of "phicons", or physical icons, in the context of an interactive surface. An infrared camera inside of a table

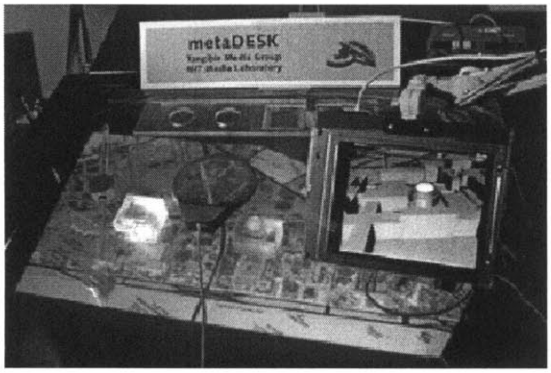

Figure 2-15: A map of the MIT campus displayed using the metaDESK system

tracked these phicons using simple computer vision techniques. Output from the system was projected into the same space using rear video projection, as shown in figure 2-15.

The Urp [39] system, shown in figure 2-16, demonstrated the use of an interactive surface for urban planning. This system used an advanced vision technique that involved tracking objects based on unique patterns of colored dots. However, the limitations of computer vision in stability, robustness, and speed were still apparent in this application.

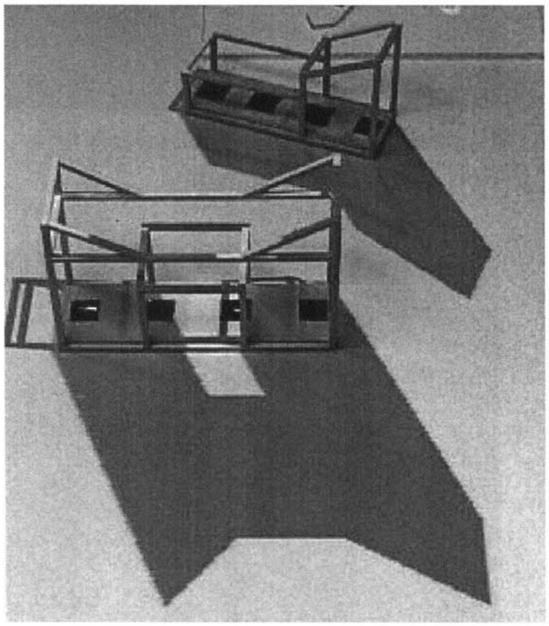

Figure 2-16: Two building models in the Urp system. The models are tracked using a camera which sees the colored squares on the bottom surface of the models. In turn, a projector above the table projects the "virtual" shadows onto the table.

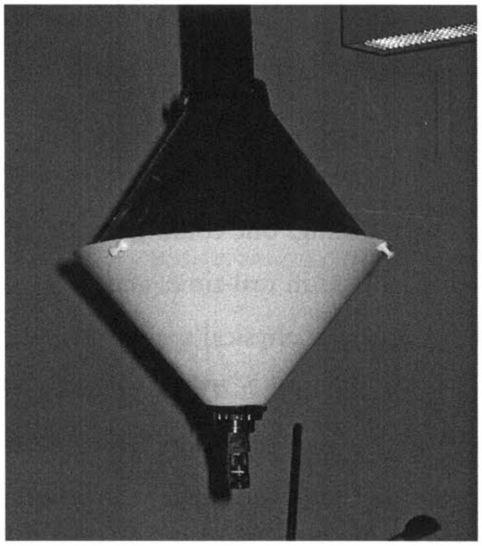

Figure 2-17: The diffuser and camera setup needed to control lighting conditions for the I/O Bulb system. This setup requires careful callibration before the computer can see the colored dots on the models in figure x.

Several commercial platforms can provide robust tracking of physical objects. However, these devices are limited by the number of objects they can track at a time. [30] Usually, a state of the art product such as the Wacom Intuos™ in figure 2-18 can track at most two input devices [41].

Zowie Intertainment, now part of the LEGO Group, released a breakthrough toy using multiple-object tracking technology at very low cost. Although their technology allows fast, high resolution tracking, the hardware only provides information about the identity and position of objects in the sensing space. However, I was interested in developing interaction techniques based on allowing the user to physically manipulate the objects using buttons, dials or by attaching modifiers. This led me to develop a new sensing platform.

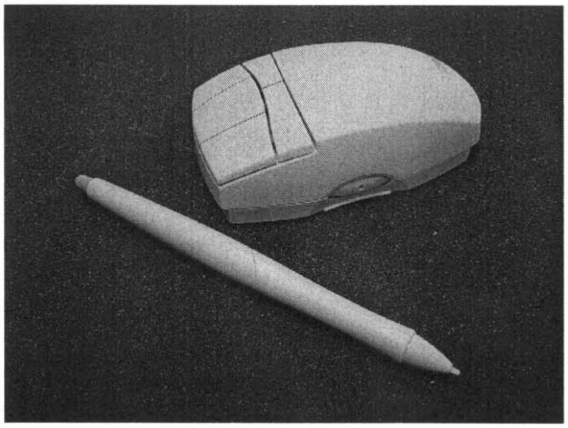

Figure 2-18: A pen and mouse that can be tracked by the Wacom Intuous system.

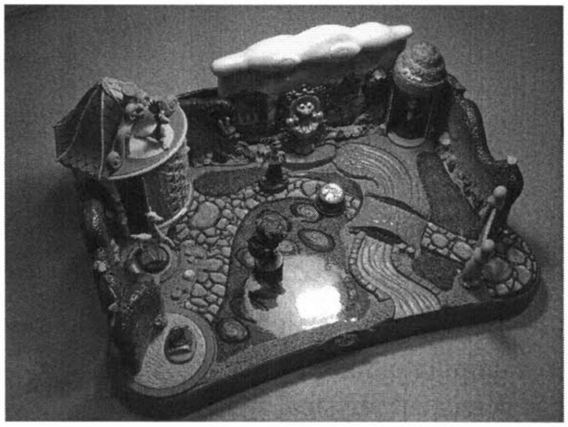

Figure 2-19: The Ellie's Enchanted Garden Playset from Zowie Intertainment (now part of LEGO).

In Urp, the use of physical models which embody the buildings they represent provides a clear advantage over an approach where all buildings are represented on the urban planner's table with roughly the same physical form. However, for more abstract applications such as system dynamics and chemistry simulation, there is no obvious intuitive physical representation of the digital objects in the system. In these applications, a more general physical form seems appropriate. Thus, in current applications, Sensetable uses the general physical form and rebinding techniques from graspable user interfaces, but it uses the persistent physical controls from tangible user interfaces. To reinforce the tangible interface principle that the puck is embodying certain digital content, I project information directly onto the pucks themselves in some applications. I discuss this in greater detail in chapter four.

2.4 Consistency of Physical and Digital State

In addition to requiring more generalized physical form, dynamic binding highlights another design challenge associated with tangible interfaces. In the system dynamics simulation application, dials on top of the pucks provide a physical representation of the changes a user has made to a parameter in a simulation. However, when a puck is rebound to another parameter, the position of its dial will not correspond to the setting of the new parameter, leading to an inconsistency. Currently, I resolve that inconsistency by setting the digital parameter to the position indicated by the physical dial. If the user wants to undo this change, he or she must physically

In Urp, the use of physical models which embody the buildings they represent provides a clear advantage over an approach where all buildings are represented on the urban planner's table with roughly the same physical form. However, for more abstract applications such as system dynamics and chemistry simulation, there is no obvious intuitive physical representation of the digital objects in the system. In these applications, a more general physical form seems appropriate. Thus, in current applications, Sensetable uses the general physical form and rebinding techniques from graspable user interfaces, but it uses the persistent physical controls from tangible user interfaces. To reinforce the tangible interface principle that the puck is embodying certain digital content, I project information directly onto the pucks themselves in some applications. I discuss this in greater detail in chapter four.

2.4 Consistency of Physical and Digital State

In addition to requiring more generalized physical form, dynamic binding highlights another design challenge associated with tangible interfaces. In the system dynamics simulation application, dials on top of the pucks provide a physical representation of the changes a user has made to a parameter in a simulation. However, when a puck is rebound to another parameter, the position of its dial will not correspond to the setting of the new parameter, leading to an inconsistency. Currently, I resolve that inconsistency by setting the digital parameter to the position indicated by the physical dial. If the user wants to undo this change, he or she must physically rotate the dial to its midway position. The end result of this approach from the perspective of the user is that a parameter in the simulation can be affected just by attaching a puck to it. This can confuse the user because the puck is supposed to be a physically manipulable representation of the data, rather than solely a tool for changing simulation parameters. One approach to dealing with this problem would be to provide extra graphical feedback to let the user know a parameter was being changed to keep it consistent with a dial. A better scenario would be that the system keeps the dials and the parameters consistent automatically without changing the parameters. This requires developing ways for the computer to control the position of the dials. This problem is an example of a larger set of problems involving giving the computer more control over the physical objects in a tangible interface. Currently, the applicability and flexibility of many tangible interfaces is limited because the physical control of system state is usually one directional. An example of one-directional physical control is the jog-shuttle dial on a VCR, shown in figure 2-22. One can physically manipulate the position of the tape by manipulating the jog-shuttle dial, but if the position of the tape changes by some other means, the jog-shuttle dial does not adjust to reflect this. The dials in Sensetable are another example of one directional physical control. The physical objects control the digital parameters, but not the other way around. Interfaces in this category have the problem that if the digital state of the system changes independently of the physical state, an inconsistency will result. Interfaces with bi-directional control can overcome this problem, as shown in figure 2-21.

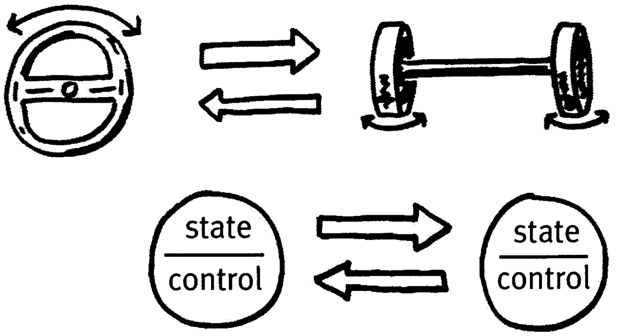

Figure 2-21: The steering wheel of a car affects the position of the car's tires, but the position of the tires affects the position of the steering wheel as well. In tangible interfaces such as inTouch, this bidirectional control ensures consistency.

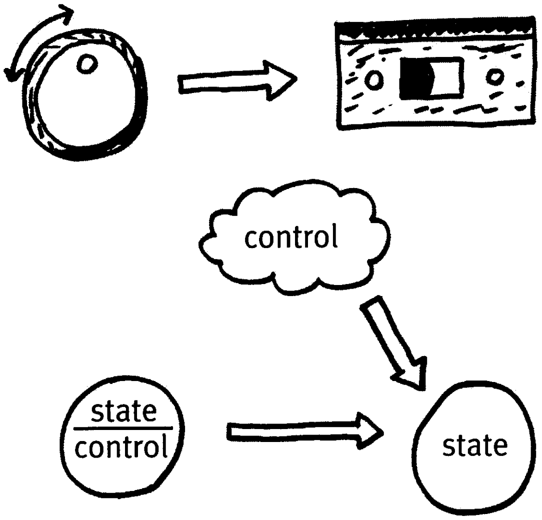

Figure 2-22: In contrast with figure x, the jog/shuttle dial on a VCR affects the position of the video tape, but the position of the video tape has no effect on the jog/shuttle dial. No attempt at consistency is made. Inconsistency between user input to the jog/shuttle dial and the motion of the tape can occur when something else causes the tape to move or stop moving. (For example, the end of the tape is reached.)

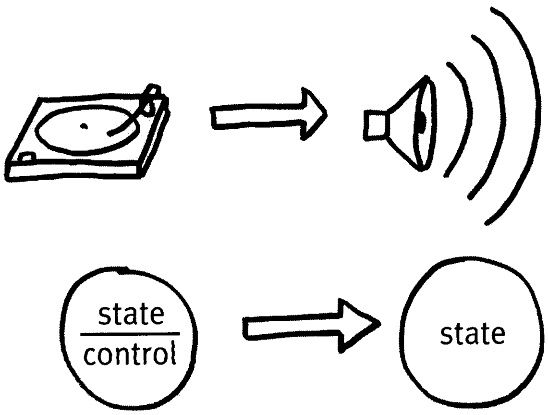

Figure 2-23: Another case is that of a record player and speaker. The user can control the location of the needle on the record to control the sound coming from the speaker. While there is no feedback from the speaker to the record turntable, consistency is still maintained between the position of the record needle and the sound coming from the speaker. This is because (in the simple case) the record player is the only thing controlling the speaker.

In interfaces with bi-directional control, the physical controls that the user manipulates can also be controlled by the system to reflect changes in system state. One example of this type of system is the steering wheel and front wheels of a car. When the driver turns the steering wheel, the front wheels rotate accordingly. At the same time, vibrations indicative of road conditions and the position of the front wheels moves from the tires back up to the steering wheel. InTouch [2] is a good example of a tangible interface with bi-directional control. This type of interface can more easily maintain consistency between physical and digital state. Thus it can present the idea that the physical objects in an interface are embodiments of digital information in a cleaner and more consistent manner. I discuss a mechanism for integrating direct computational control of physical parts of an interface into the Sensetable project in the "future work" section of chapter six. In the absence of this direct computational control, one must design a tangible interface carefully to avoid confusing the user through inconsistency.

The research described in this chapter suggests that physical objects can aid in some problem solving tasks in several ways. A user can employ them to encode information about a task or offload computation, for example. The use of spatial memory may provide cues to allow the user to remember the location of important data and tools more quickly. Systems such as Urp and Bricks have shown that computationally augmenting these physical objects can further help the user during many problem solving tasks. However, these systems have also demonstrated the need for a robust, wireless object tracking platform like Sensetable. Having explained the motivation behind the Sensetable platform, in the next chapter I will discuss the implementation of the three Sensetable prototypes.

3. Implementation

So far, I have developed three implementations of the Sensetable platform, each with somewhat different performance characteristics. The first implementation used modified Wacom digitizing tablets to sense objects. Once this system was complete, I began working on a system which used modified sensing hardware produced by Zowie Intertainment. At the same time, I began collaborating with other researchers in the MIT Media Lab to develop a sensing platform from scratch. In this chapter, I discuss the technical details of each platform's implementation, as well as the software architecture shared among the hardware platforms.

3.1 Wacom-based Implementation

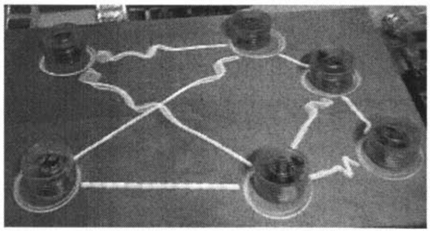

Figure 3-1: The Wacom-based implementation of Sensetable

The initial implementation, known as Sensetable 1.0, uses a pair of modified commercially available Wacom Intuous(tm) sensing tablets that are placed next to each other to form a 52cm x 77cm sensing surface. These tablets are an appealing technology to use for the Sensetable project because they can sense the positions of objects with roughly 1000 dpi resolution, and have very low latency compared to computer vision based approaches. As well, the mice used with these tablets each have a 32 bit serial number, which is useful for identifying mice when they move from one sensing surface to another. On the other hand, these tablets can only track two objects at a time. To circumvent this problem, I built the pucks to be tracked by augmenting the mice with a circuit to switch the sensing coils inside of the mouse on and off randomly. The random number generator I use ensures that each puck is turned on about one third of the time.

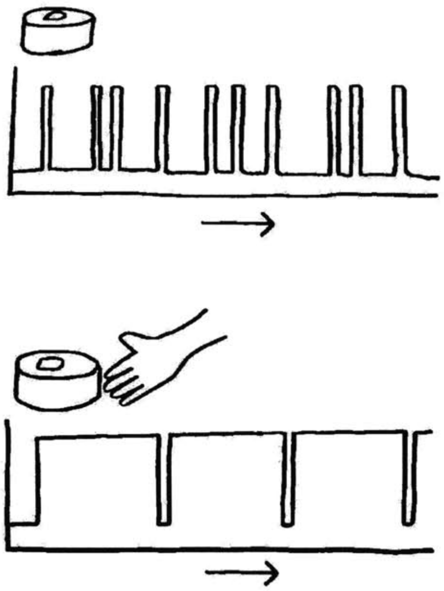

Figure 3-2: The capacitance sensor detects when the puck is touched, and increases the duty cycle of the coil inside to decrease tracking latency.

This duty cycling approach yields a tracking latency of less than a second. To reduce this latency, I added a capacitance sensor to sense when the puck is being touched. This sensor monitors an antenna wire wrapped once around the circumference of the puck. When the puck is touched, the microprocessor inside it detects a capacitance above a certain threshold, and it turns that puck on 100% of the time, as shown in figure 3-2. In this way, the system can track objects that are being touched at a latency equal to that of an unmodified Wacom(tm) tablet. Objects that are not being touched are updated with a higher latency.

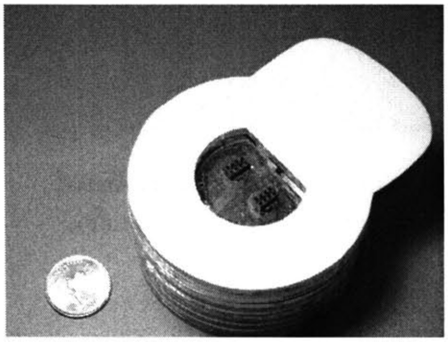

Figure 3-3: A Sensetable puck, with a socket for attaching a dial or modifier. A US quarter is shown for scale.

The pucks have two sockets inside of a crescent shaped recess on their top surfaces, shown in figure 3-3. These sockets connect to a 16 wire bus which is used to communicate with dials and modifiers which can be placed on top of the pucks. Currently, four of these pins are used to communicate with the dials, four are used to communicate with the modifiers, and eight pins are reserved for later use. The modifiers have a unique digital ID, and bus connectors on the top and bottom so they can be stacked. Currently the stacking order cannot be detected, but it is possible to add more intelligence to the modifiers to allow this. Some modifiers are shown in figure 3-5. Because the dials use the same bus connector as the modifiers, they can be used while attached directly to a puck or while on top of a modifier or series of modifiers. A dial is shown in figure 3-4.

Limitations

If more than two pucks on top of one of the sensing tablets are touched at the same time, tracking latency increases. In testing with one and two users, this limitation was not a problem, because users did not typically move more than two objects at a time. However, I have not tested the interface in collaboration scenarios with larger groups of people. The other prototypes, which are described later in this chapter, do not to have this limitation.

Another limitation is a 3.5 cm gap in the sensing field due to interference between the two boards, where the two sensing elements touch each other. The other prototypes do not have this problem.

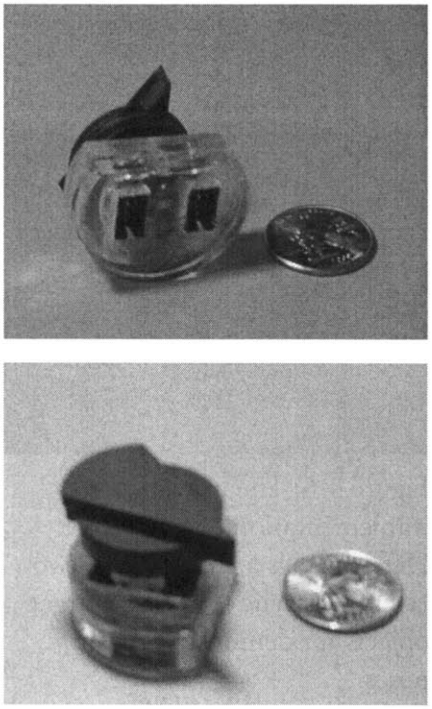

Figure 3-4: The top and bottom of a dial that plugs into a Sensetable puck.

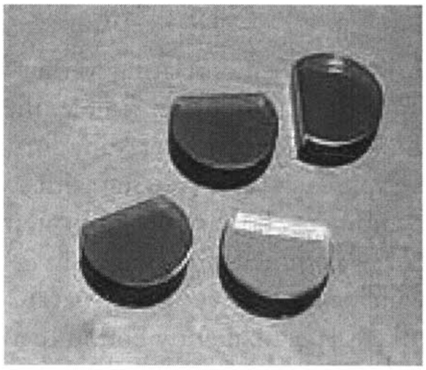

Figure 3-5: Some modifiers, with unique digital IDs, which plug into the puck above.

3.2 Zowie-based Implementation

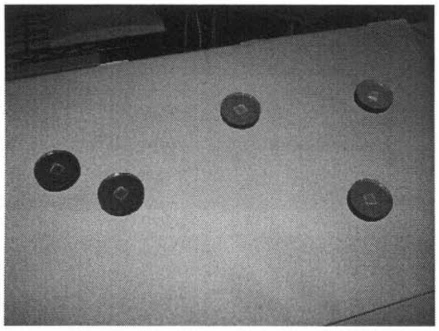

Figure 3-6: The Zowie-based implementation of Sensetable. Here the tags are encased in two layers of acrylic to provide larger objects for demonstration purposes.

After the Sensetable 1.0 implementation had been completed, I began work on two other implementations. Each of these implementations aimed to overcome different limitations of the initial Wacom-based prototype. One of these implementations, known as Sensetable 1.5, was based on commercial tag tracking technology developed by the Zowie Intertainment corporation, which was subsequently bought by the LEGO corporation. Zowie had based their development effort on some patented technology licensed from Scientific Generics corporation.

Zowie developed this technology for use in computer games for children. They developed two games in which children used a series of figurines like those in figure 3-8 on top of a larger play surface to control the action happening on the computer screen. For example, in one of Zowie's games called Redbeard's Pirate Quest™, the child could move the physical models representing characters such as a pirate around the model of a pirate ship. The pirate ship model included several areas where one could place characters to trigger specific actions on screen. For example, one could place the model of the pirate behind the cannon on the ship in order to see the pirate fire the cannon on the screen.

Aside from issues relating to the implementation of Sensetable, the Zowie platform is interesting for its approach to interfacing with computers. Specficially, it is one of few commerical systems in which a series of several physical tokens is permanently bound to a series of digital associations, and the position of those objects maps directly onto a series of computational results. It is exciting that Zowie chose this mechanism for interacting with a computer in the context of childrens' play. Hopefully more commercial products will explore this interface style in the future.

While I had initially hoped to gain access to the software development kit that Zowie developed, it turned out that this was not possible due to various intellectual property related concerns. So working with Jason Alonso and Ali Mazalek of the Tangible Media Group, I reverse-engineered the system in order to use it in the Sensetable development effort.

On a technical level, the Zowie sensing technology is capable of tracking up to nine tags, over a surface measuring 26 cm by 36cm. This dimension is the size of the sensing surface used in the "Ellie's Enchanted Garden" system, but it is not clear what the fundamental limitations are on how large a sensing surface this system could support. The only information readily available about each tag is its x and y position on the sensing surface. However, it is also possible to estimate z position within a small distance from the board. It may be possible to perform some computation to infer some rotation information as well, but I have not explored this.

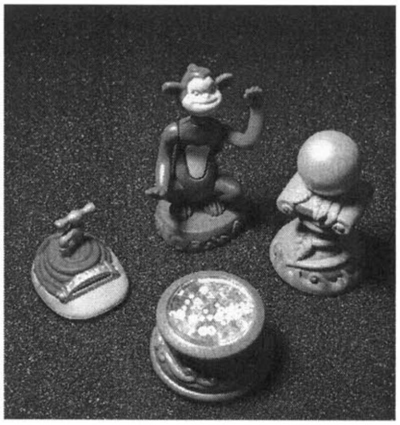

Figure 3-7: Several figurines from the Zowie Ellie's Enchanted Garden playset.

Two other notable qualities of the Zowie sensing hardware are that the sensing surface is both transparent and flexible. The transparency makes it possible to think about using the system with back projection, as well as having sensing surfaces which are not planar. One example where non-planar sensing surfaces would be useful is in the current Sensetable configuration of a flat sensing surface and two rear flat-panel displays. Being able to sense objects not just on the tabletop, but also on the surfaces of the rear displays as well would open up new possibilities for ways to share information between the rear displays and the tabletop surface. A potential application which would take advantage of the system's transparency is placing the sensing surface on top of the display screen of a laptop as in figure 3-9. In this way, it would be easy to explore interactions which involve displaying graphical information around the physical objects themselves, while at the same time minimizing the need for unusual interface hardware.

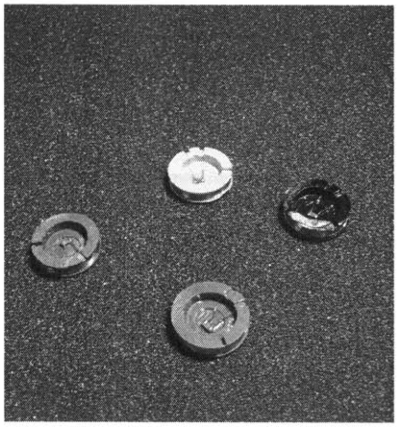

Figure 3-8: Each of the figurines in a Zowie-based playset contains a tag with a unique resonant frequency.

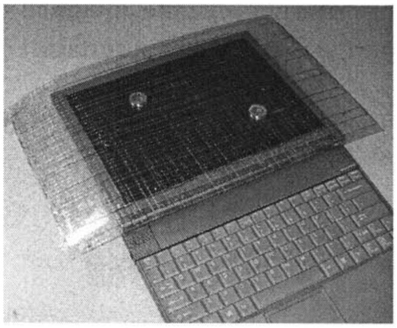

Figure 3-9: One potential way to use the Zowie circuit with a laptop display screen.

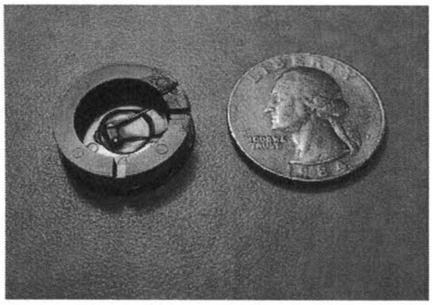

Figure 3-10: A Zowie tag, with a US quarter shown for scale.

Each tag consists of an inductor and a capacitor in parallel. Together, these components form what is known as a "tank circuit" , which resonates when excited by electromagnetic energy at a certain frequency. This frequency varies as a function of the inductance and the capacitance of the components in the tag [16]. Each of the tags on the sensing surface must have a unique resonant frequency for the sensing technique to work. Because these tags consist of only two components, they can be quite small, as shown in figure 3-10.

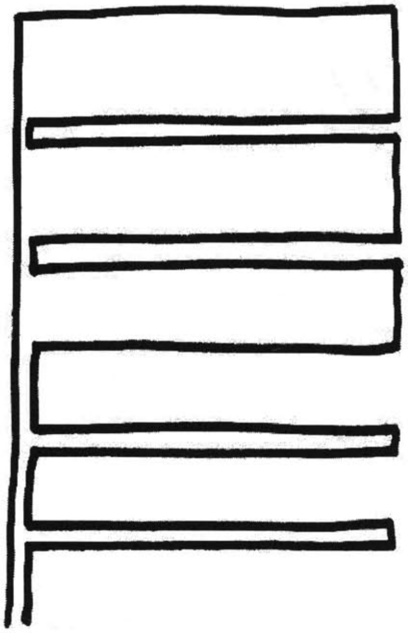

Figure 3-11: Typical pattern of one of eight separate antennae on the Zowie sensing surface.

These tags are tracked using a series of overlapping loops of wire in the sensing surface. When a particular loop of wire emits electromagnetic energy at a certain frequency, tags resonant at that frequency which are within that loop of wire will resonate. Tags outside of the loop of wire will not resonate, regardless of their resonant frequency. The antenna does not have to be a perfect circle for this to hold true. In fact, a variety of antenna shapes can be used, as shown in figure 3-11. The Zowie system uses eight loops of wire which cover the whole sensing surface. Each loop covers a different region of the sensing surface. Thus, depending on the location of a tag, it will resonate to a different degree with each of the eight sensing coils. By measuring the level of resonance with each of the sensing coils, one can compute the location of a tag on the surface.

Four of the coils are used to sense X position, while the other four sense Y position. As a tag is lifted off of the sensing surface, the level of resonance with all eight antenna loops drops off proportionally to the distance from the surface, so this drop in resonance can be used to sense Z position. Coils that are used to sense X and Y position are symmetrical with respect to the Y and X axes respectively, so that the position of a tag along only one axis affects the level of resonance with the antenna.

Linear position along the X or Y axis is computed as follows. The levels of resonance with each of the antennae, a,b,c and d, vary sinusoidally as a function of position. If one end of the board is considered 0, and the opposite end is considered 2*pi, then a and b vary as a cosine and sine of the position along that axis, as shown in figure 3-12. Likewise c and d vary as a cosine and sine of the position along that axis. Neither a and b nor c and d can uniquely identify a position along the axis, but a/b varies as a tangent of the 0 to 2*pi value along the axis, and this value can provide a unique position calculation, as shown in figure 3-13. However, this position measurement is a relatively low-resolution one, because the tangent function only has one period within the length of the board. However, the functions of c and d have four periods within the length of the sensing surface. To get a higher resolution position measurement, I use the arctan(a/b) function to determine which of four board quadrants a tag is in. These quadrants each correspond to one of the four periods of the function arctan(c/d). Once the particular quadrant of the board is known, the function arctan(c/d) can be used to uniquely identify position with higher precision and accuracy.

With the mechanism described above, the board can track up to nine objects in real-time. However, the Zowie board requires careful manipulation of the hardware flow control lines on the serial line before it will provide any tracking data at all.

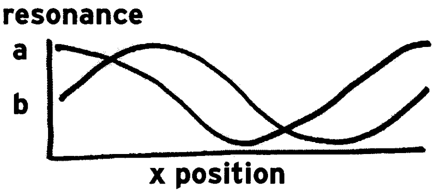

Figure 3-12: Antenna elements A and B resonate according to a sine and cosine function of position, repectively. The functions have one period over then length of the board.

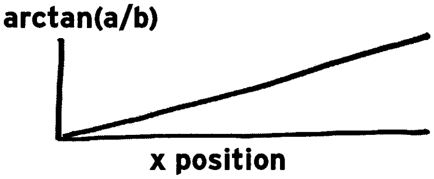

Figure 3-13: Arctan(A/B) varies linearly with position.

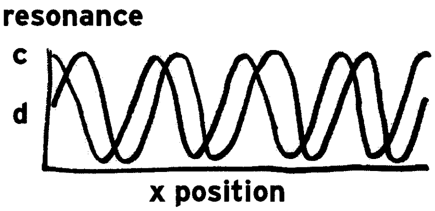

Figure 3-14: Antenna elements C and D resonate in a manner similar to that of A and B, except that the sinusoidal functions have eight periods over the length of the sensing surface.

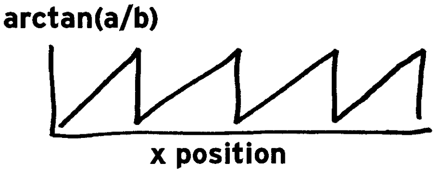

Figure 3-15: Arctan(C/D) can be used to determine a more specific position value. Arctan(A/B) can be used to determine the correct period of the function for the position measurement.

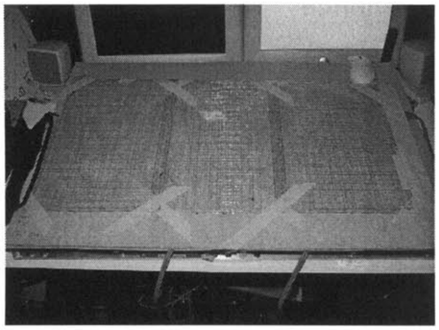

Figure 3-16: The sensing elements from three Zowie playsets tiled together. There is a slight overlap of the sensing area to eliminate the gaps in sensing area present in the Wacom-based prototype.

Once the board is powered on, it must be sent an initialization sequence. After this, the software queries the board to determine whether the board firmware has been loaded or not. If not, the firmware is loaded over the serial line. After this, the board can be polled for the presence and position of each tag in turn. The need to load the firmware suggests that one might be able to significantly enhance the functionality of the zowie board by modifying the firmware. Based upon some simple decompilation, the firmware seems to be based on the instruction set of the Intel 8051 microcontroller. One modification that might be useful to explore would be adding the ability to track more than nine tags. This would simply involve telling the board to resonate each of the antenna coils at a different frequency than the nine tags currently used. One could then construct tags with the appropriate inductance and capacitance to resonate at the new frequency.

Because the sensing surface of a single Zowie board is rather small, I tiled several of the boards together to obtain a larger sensing surface as shown in figure 3-16. These boards overlap by about 3 cm on each side to eliminate gaps in the sensing area. The current prototype uses three Zowie boards tiled together. A Comtrol Rocketport serial card communicates with the boards. This card provides eight high speed serial ports, and up to four of these boards can be installed in a single computer. In addition, the Rocketport board works very well with Linux, which runs on the computers running Sensetable. The software polls each board for tags in turn. Because it takes about one second to poll all three boards for all tags, it only polls for all tags once every 100 times through the polling cycle. During the rest of the polling cycle, the software only polls for tags which were present during the last complete poll for all tags. This drastically reduces the latency between when a tag is moved on the sensing surface and when the software application is aware of the tag's new position. One could also reduce latency further by polling for tags on multiple boards at the same time, and sending polling requests for multiple tags to a single board at the same time. However, I have not yet explored these these possibilities because the approach described above provides sufficiently low latency by itself.

Another challenge associated with the Zowie platform is developing mechanisms to track information about object orientation and other physical controls on a tag such as buttons, switches and modifiers. I have constructed an orientation sensing tag by placing two tags beside each other to form a larger "meta-tag." The software uses the relative positions of these two tags to infer the orientation of the meta-tag. The downside of this approach is that it reduces the overall number of independent tags that can be simultaneously tracked. This meta-tag includes a momentary pushbutton switch on top. This switch sits in parallel with the inductor in the tank circuit on each tag, as shown in figure 3-17. The switch breaks the circuit when pressed, stopping the tag from resonating. If one tag were used with a pushbutton rather than two, the system might become confused if the button was held down for a long time, because from the perspective of the supporting software the tag would have disappeared.

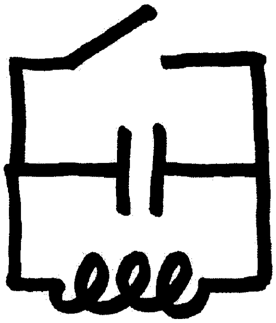

Figure 3-17: Schematic for a Zowie tag that can be disabled with a switch.

However, a momentary button press could be detected using one tag because it involves the sudden absence of a tag's resonance during the polling process for a brief period of time, without the characteristicly slower decreases in resonance level associated with removing a tag from the board by picking it up or sliding it sideways off of the sensing surface. With two tags together in the meta-tag, both momentary pushbuttons and toggle switches are possible. Figure 3-18 shows one such meta-tag. Even when the switch disables the coil of the tag it is attached to, the unmodified tag can still report the position of the meta-tag. Position information is available using both tags if a pushbutton switch is used. The tradeoff here is that no rotation information can be obtained when the button is pushed down. However because this is a momentary switch, in practice the temporary loss of rotation information should not be a problem.

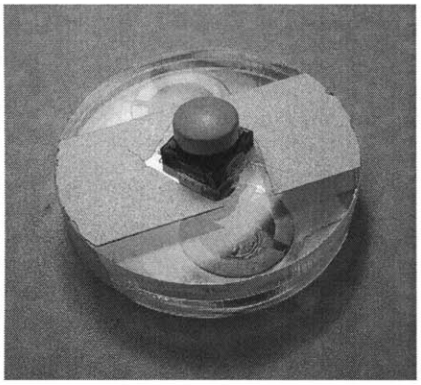

Figure 3-18: A "meta-tag" composed of two Zowie tags and a momentary pushbutton switch. The system can sense the position and orientation of this tag, as well as whether or not the button is pressed. It cannot detect changes in rotation while the button is pressed.

Another way to improve the functionality of the Zowie-based Sensetable implementation would be to add a small PIC microcontroller onto each tag, as shown in figure 3-19. This PIC could detect the state of various attached controls, and periodically disable the resonant tank circuit of the tag using an optoisolator or a MOSFET to signal this information to the software reading the sense data. A PIC might also be used to overcome the current limitation of nine objects per Zowie board. If the PIC were attached to several Zowie tags in a larger meta-tag, the PIC could potentially even switch between enabling various Zowie tags based on which tags were enabled in nearby meta-tags. In this way, the PIC controlling a meta-tag could dynamically change the resonant frequency of that meta-tag to avoid conflict with neighboring ones. As well, the random scheduling techniques employing capacitive sensing which we used on the Wacom-based Sensetable could be applied to this platform. While all of these PIC based approaches are exciting, they share the disadvantage that they

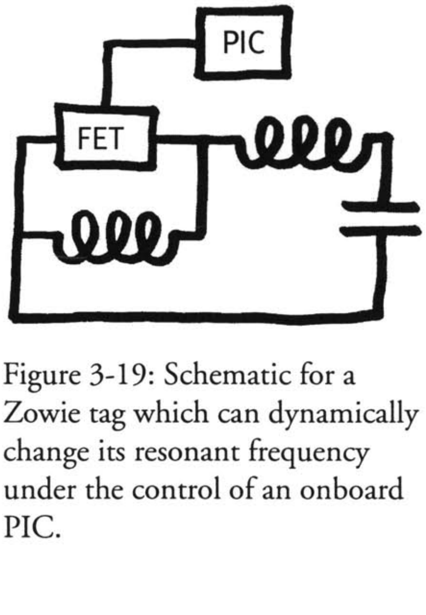

Figure 3-19: Schematic for a Zowie tag which can dynamically change its resonant frequency under the control of an onboard PIC.

require on-board power. This issue might be addressed to an extent by having an area in the interface where pucks not currently being used would be stored. These pucks could have metal contacts on the bottom which would recharge the pucks in this recharging area.

3.3 Capacitive Implementation

In addition to the Wacom and Zowie-based Sensetable implementations, I have also developed a Sensetable implementation based on hardware developed at the MIT Media Lab by Matt Reynolds of the Physics and Media Group. This implementation is known as Sensetable 2.0. Together with Gian Pangaro, I have implemented the necessary software and firmware, as well as a few hardware modifications necessary to make the system work. The advantage of developing this system from scratch inside the lab is that we have have the freedom to make changes to the design at a very level in order to maximize performance along a variety of axes such as latency, power consumption, physical size, etc.

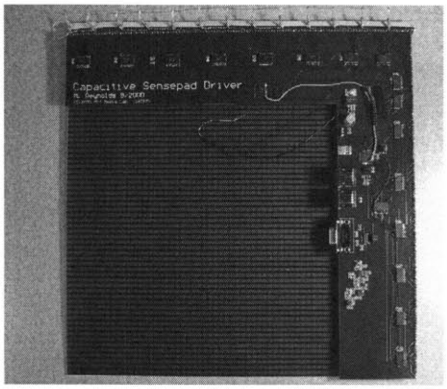

Figure 3-20: The first prototype of the sensing element of the capacitive Sensetable implementation.

One of the main features of this system is that it is designed to be tileable. Each sensing element consists of a 14" square surface, shown in Figure 3-20. These can be put together in a variety of configurations to yield interaction surfaces of various shapes and sizes. These sensing tiles communicate data about the tags on top of them through an RS485 network back to the host computer. This computer then uses information about how the boards are physically organized to assemble the tag data into a larger coordinate space.

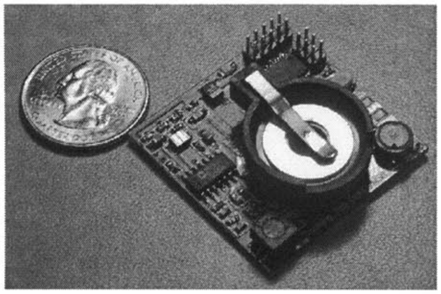

Figure 3-21: A position sensing tag for the capacitive Sensetable implementation.

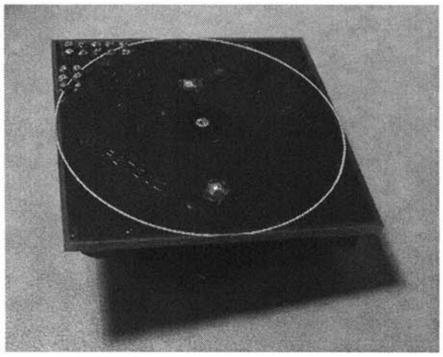

Figure 3-22: The bottom of the same tag. The center of the bottom layer includes a small circular antenna element which picks up pulses from the board below.

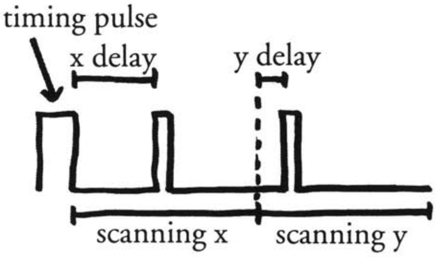

Figure 3-23: Timing pulses detected by a tag on the sensing surface. The tag uses the time interval between pulses to determine its location on the surface.

Each tile uses an array of capacitive antennae to determine the positions of objects. There are 64 antennae in the X direction on the top layer of the sensing circuit board, and 64 antennae in the Y direction on the bottom of the circuit board. Each of the tags tracked by this surface has a small circular antenna on its bottom surface. This antenna capacitively picks up signals coming from the antenna element directly beneath it on the sensing surface. The tags contain an amplification circuit which favors incoming signals oscillating at 200 kHz.

To detect the location of a tag on the sensing surface, the PIC microcontroller first sends a timing synchronization pulse, which involves oscillating all of the antenna lines at the same time. When each of the tags on top of the board detects this pulse, they reset an internal timer. The tags are able to differentiate the synchronization pulse from the other pulses coming from the system because it has a longer duration than other pulses. The sensing surface then oscillates each antenna element in turn; first those in the X dimension, then those in the Y dimension. When an antenna underneath a tag is oscillated, the tag detects that oscillation and uses its timer to measure the duration between this pulse and the initial timing pulse. Figure 3-23 shows the pulses normally detected by the antenna on each tag. The tag uses this timing information to compute its X and Y position on the surface. The tag then radios this information back to the sensing board using a very low power RF transmitter. The sensing board in turn relays this information back to the host computer.

There is no collision detection in the radio transmission of information from the tags to the sensing surface. This means that the tags must use a collision avoidance scheme. We have explored two such schemes. The first scheme avoids collisions deterministically, but takes a long time to transmit data. The second scheme may occasionally lose tag information, but is faster. In the first scheme, the sensing surface is divided into a 10 x 10 grid of locations. Each of these locations has a scheduled time to transmit its position data back to the underlying surface. Once the underlying board has finished oscillating each of its antennae in sequence, each tag waits for its transmit time slot based on its location, and then transmits. This scheme guarantees no collisions, as two tags cannot physically occupy the same location on the board. However, since most of the transmit slots will be unused, this scheme wastes a fair amount of time. It is best for applications in which one expects many tags to be present on a single sensing tile at the same time.

The second collision avoidance approach involves simply waiting a random amount of time before transmitting data. This is a simple and common scheme. It is best for applications where one does not expect a single tile of the sensing surface to hold many tags at a time.

One hybrid scheme which would be interesting to explore would involve switching between the two schemes described above on a per-tile basis depending on the number of tags on each tile. As each tile would know the number of tags on top of it, the tile could vary the length of the timing pulse sent to the tags to let the tags know which scheme to use.

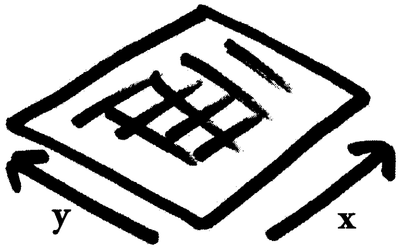

Figure 3-24: In turn the board underneath the tags strobes every antenna line along the X axis, and then every line along the Y axis.

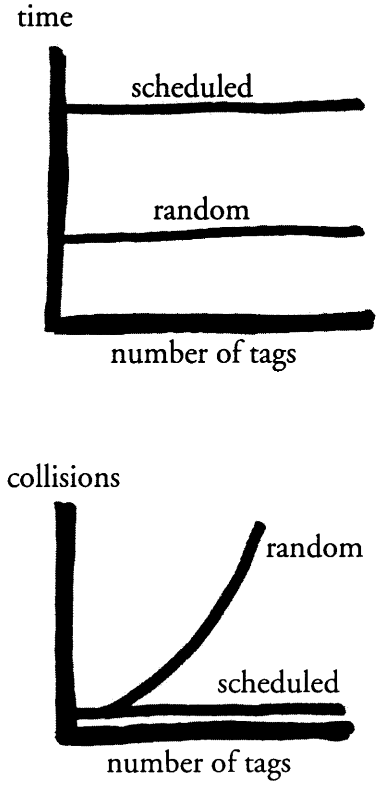

Figure 3-25: Scalability of the two tag communication strategies.

In the simplest implementation, this system is only able to sense the position of a tag. As in the Zowie-based Sensetable, two tags can be physically attached to each other to form a larger meta-tag which is orientation-aware. However, we are currently developing a new tag circuit which is able to detect position and orientation using just a single tag. This circuit will work by sensing pulses from the underlying antenna arrays at two separate corners. It will use the timing of these pulses to determine the position of two of the corners, and thus its orientation.

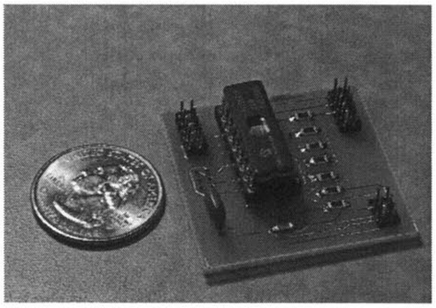

Figure 3-26: This board can be added to a tag to sense dials and modifiers. A US quarter is included for scale

To monitor information about additional physical controls that might be attached to a puck, this system uses an additional circuit board which is connected to the main tag board. This board is the same size and shape as the main tag board, and contains 5 digital and 3 analog I/O pins and a PIC microcontroller. This board receives power from the lower board, and transmits data back to it about the state of the digital and analog I/O pins on the top-layer board. The lower board periodically sends a full update of the state of all pins on the top board back to the sensing surface at the same time it is sending its position data. In most cases, it only sends data about changes in state that have happened since the last complete update.

One of the things that differentiates this implementation from the other two is that in this implementation, the tags know where they are. In the other two implementations, the tags cause a resonance with the underlying surface, but the tags themselves do not have any information about their position. The fact that the tags compute the location information in this implementation plays a large role in making it able to track many more tags than the other two implementations can. The process of determining tag positions is a parallel computation which takes place on each of the PICs on the tags. In fact, the limiting factor in the number of tags that can be tracked using this implementation is the number of tags that can physically fit on the sensing surface.

3.4 System Architecture

Two dual processor 866MHz Intel(r) Pentium(r) III Xeon(tm) computers are used to drive the system. One receives the data from the sensing surface and displays graphics onto the sensing surface in response. A second computer drives two vertical displays at the rear of the sensing surface, which can provide extra information relevant to the interaction happening on the table. In the system dynamics simulation application, this second machine also performs the actual simulation. In the future I plan to use both machines together to simulate larger system dynamics models in real-time. The system architecture is shown in figure 3-27.

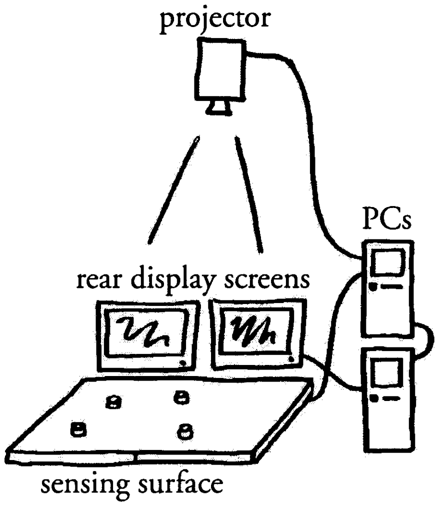

Figure 3-27: Architecture of the system. The top PC in the diagram reads sensing data from the sensing surface itself, and renders images onto the projector above the table. The bottom PC renders onto the two rear display screens, and provides extra compute power for some applications, such as system dynamics simulation.

To make it easier to develop applications which run on the Sensetable platform, we have implemented an application program interface which provides a consistent interface mechanism for each of the three Sensetable implementations described in this thesis. This API, designed primarily by Professor Robert Jacob, provides a uniform interface to each of the three Sensetable implementations. In addition, a Java version of this software layer provides an interface to the Senseboard system [19]. The interface uses a callback model, where the application programmer can register a variety of

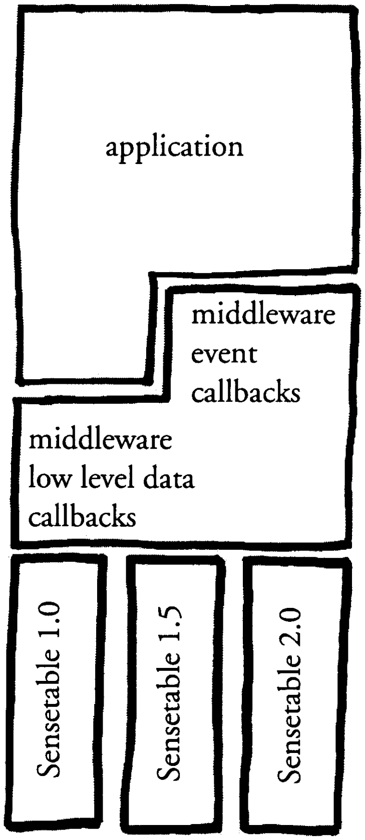

Figure 3-28: Software architecture of the Sensetable system.

functions which are called when various events occur. At the lowest level, the programmer can register an event handler that is called whenever new data arrives from the sensing surface. At a higher level, the programmer can register callback functions for when a puck moves, when its state changes (i.e. a button is pressed, or dial turned), or when a puck enters or leaves the sensing surface.

All of the Sensetable hardware implementations have a small amount of tracking jitter which can at times cause the position and orientation values reported by the sensing surface to vary slightly. To accommodate these differences, the application programmer can set the levels of sensitivity for movement and rotation events. For example, if the movement threshold level is set to five pixels, a movement event will occur once for every five pixels of motion. Each time an event based on the position or the orientation of the puck fires, the middleware layer stores the position and orientation of the puck. Another event fires when this stored information about the puck differs from the newly reported information about the puck by larger than the programmer specified threshold. This approach allows the system to filter out jitter while still detecting very slow intentional movements of the puck by the user.

To further deal with jitter, Sensetable 1.0 and 1.5 have extra filtering to reject some position readings from the sensing surface. These filtering routines simply compare each position value read from a puck with the last position of that puck. If a value differs from the one before it by an amount larger than a certain threshold, that value is rejected. This approach greatly reduces visible jitter.

While each of the three Sensetable platforms can be used with the same API, each platform has certain qualities which make different interaction techniques suitable for it. The 1.0 implementation has relatively large pucks. While these are suitable for applications where a user would typically only have one or two hands on pucks at a time, the interaction space would become cluttered when more than a dozen pucks were used at the same time. The gap in the center of the sensing space is also an issue that the application programmer must consider with this prototype. Ideally, an application should help the user avoid this space by moving items the user may want to interact with out of the space.

In contrast to the Sensetable 1.0 platform, the Sensetable 1.5 platform eliminates gaps in the sensing surface, and has much smaller tags. These features make this platform appropriate for developing more complex "meta objects" with various form-factors and multiple tags per object. The transparency and flexibility of the sensing surface could also open up interesting possibilities from the application point of view, but these possibilities remain largely unexplored.

System	Technology	Puck diameter	Tags need batteries?	Number of tags	Orientation	Dials and modifiers	Buttons
Sensetable 2.0	Capacitive	3.2 cm	yes	many	with meta-tag	yes	yes
Sensetable 1.5	Zowie	2 cm	no	9	with meta-tag	no	yes
Sensetable 1.0	Wacom	8.2 cm	yes	6	yes	yes	yes

Figure 3-29: A comparison of the three Sensetable implementations.

The Sensetable 2.0 platform is most appropriate for applications which require more than ten tags. While the tags in this implementation are currently a bit larger than those in Sensetable 1.5, they are much smaller than those in Sensetable 1.0. It should be possible to shrink the size and power consumption of these tags considerably as the design matures.

Together, these implementations show that one can implement a platform for tracking objects on a flat surface using a variety of techniques. These techniques will inevitably involve tradeoffs of tag size, scalability, power requirements, latency etc. Having shown that is possible to construct the Sensetable platform, in the next chapter I will present some good reasons for constructing the platform, in the form of applications and interaction techniques that are interesting and possible on this platform, but are less feasible on existing sensing platforms.

4. Applications and Interaction Techniques

The Sensetable applications I have implemented include a system for analyzing system dynamics models, a tool for children to learn about chemical reactions, and three applications for real-time musical performance and composition. For these applications a development process like the following occured: The process started with an idea for something for which Sensetable might make a good interface. In the process of implementing each application, I would experiment with new interaction techniques to respond to design challenges in that application. For example, techniques for binding and unbinding pucks to data developed in response to the challenge that there were not always enough physical pucks to map to all pieces of digital data at the same time. In this chapter, I discuss each of the Sensetable applications and the interaction techniques I explored in the context of each application.

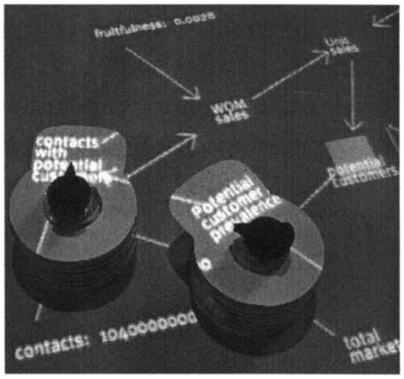

Figure 4-1: The system dynamics application running on top of Sensetable

4.1 System Dynamics Simulation

The most mature Sensetable application is a tool for analyzing models of complex processes using system dynamics simulation, shown in figure 4-1. System dynamics is a method for studying complex feedback systems in fields such as business and the social sciences. It involves the analysis of computer models to conduct "what if" analysis on a system. Using this analysis, one can develop an understanding of how the different parameters in a model affect each other. For example, in a model of the fox and rabbit populations in a forest, the size of each population would have an effect on the size of the other because of the predator/prey relationship between foxes and rabbits. One might hypothesize that an increase in the fox population would lead to a decrease in the rabbit population. One could then adjust the fox population in a simulation of the model to test this hypothesis.

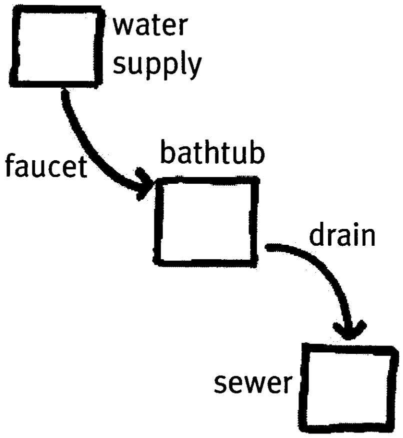

Figure 4-2: A portion of a system dynamics model. The amount of water in the bathtub affected by the amount flowing into the faucet, and the amount flowing out of the drain. The boxes in the diagram are known as "stocks" or "levels" and the arrows are known as "flows".

A system dynamics model consists of a series of nodes (such as the rabbit and fox populations above) connected via a series of edges. The edges represent flows from of information or material from one node to another. Figure 4-2 shows a simple example. The amount of water in a bathtub (called a "level" in system dynamics) is a function of the amount of water in the bathtub earlier, plus water that had been added through the faucet, minus water that has gone down the drain. In this model, as water flows into the bathtub, the level of water in the reservoir supplying the bathtub would decrease. As the level in the bathtub decreased, the level in the sewer would increase.

One important difference between system dynamics simulation and other simulation approaches such as discrete event simulation is its emphasis on causal loops. System dynamics is good at understanding how patterns of activity affect themselves over time. For example, a common problem used to discuss simulation methods is modeling the length of the line at a bank. A discrete event model of the process might model new people joining the end of the line according to a random arrival rate. People would leave the line at a rate dictated by the number of tellers. The insight that system dynamics simulation brings to this problem is the relationship between the length of the line and the rate at which people join the line. If the line is quite short, people walking by the bank are more likely to get in line to take care of banking business. On the other hand, if the line starts to extend outside of the bank itself, potential customers will be less likely to get in line because they do not want to wait. In other words, using system dynamics simulation to analyze the length of the line at a bank would help one understand the causal loop through which the rate of people leaving the line affects the rate at which people enter the line. [Jim Hines, personal communication]

These causal loops often behave counterintuitvely. For example, in a simulation of a business supply chain, one might find that the inventory in a particular warehouse tended to oscillate between a surplus and a shortage of parts. A typical response might be to take action elsewhere in the company to correct the inventory as soon as a problem was noticed. However, the system dynamics model of the causal loops involved might reveal that by waiting longer before reacting to inventory problems, one might cause the oscillations to subside more quickly. Because of the counter-intuitive nature of these causal loops, an important part of analyzing a system dynamics model is adjusting the parameters in a causal loop to determine how the changes affect certain key parameters in the model. The system dynamics application which runs on top of sensetable is designed to facilitate quick and easy adjustments of different parameters in the model.

4.1.1 User Interaction

When a user first begins interacting with the system dynamics application, he or she sees a complete version of the system dynamics model to be analyzed on the vertical display at the left rear of the interface. Directly below this graph is a display of several portions of the model that contribute significantly to the model's overall behavior. The author of the model has selected these portions in advance. The user can move one of these subgraphs from the vertical display to the tabletop sensing surface using the data sharing technique described below. As the puck is moved away from the screen, the subgraph expands to fill the TUI space, while one node in the subgraph stays attached to the puck.

The user can then bind other pucks to nodes in the graph. Once a puck is bound to a node, one can use the dial on top of the puck to change the value of the parameter corresponding to the node. When one changes a parameter, the system completely recomputes the simulation of the model using the new value of the parameter. A simulation engine called HinesSight, developed by Dr. Jim Hines, performs the actual simulation. The system then updates graphs of the levels over time on the rear-right display to reflect the results of the new simulation. In addition, small thumbnail graphs of the parameter values over time appear next to the corresponding parameters on the table. If one would like to manipulate a parameter, and there are no free pucks available, one can unbind a puck from a node by shaking it from side to side. Once unbound, a puck can be attached to any unbound node. Once one is through adjusting parameters within a particular subgraph, he or she can return it to the on-screen space and choose another.

4.1.2 Interaction Techniques

4.1.2.1 Binding and Unbinding

One goal for the Sensetable project is to provide seamless coupling between the physical pucks and the digital data they represent. Users should be able to think about manipulating the pucks as manipulating the digital data itself, rather than just using a tool to manipulate the digital data as one might think of using a mouse to press a button in a GUI. Thinking about the interaction as manipulating the digital data itself presents a simpler model of the interaction to the user. In addition, if the pucks are used to represent the digital data rather than just as tools to grasp and interact with it, the interaction requires fewer steps on the part of the user. However, in the context of Sensetable, these advantages must be reconciled with the need to interact with more digital objects than one has pucks available. One may need to dynamically rebind the pucks to different digital objects, but one should do this in a way which is as seamless as possible, and requires little effort on the part of the user.

When dealing with small models on Sensetable in the system dynamics application, one approach is to attach a physical puck to a digital item just by moving the puck within a certain proximity of the object to be bound. This method is simple and works well, but as the complexity of the graph increases it can become difficult to select something to be bound without accidentally selecting something else first. The system dynamics application incorporates two measures to address this issue. First, the spacing of digital items displayed near an unbound puck dynamically adjusts to make it easier for the user to select a particular one. As well the application requires an increased the amount of time for the binding process to occur. If the user moves the puck toward an item on the table, the system displays graphical feedback that indicates the given item will be bound to the puck shortly if the puck is not moved. Before the binding process is complete the user can move the puck to cancel the operation.

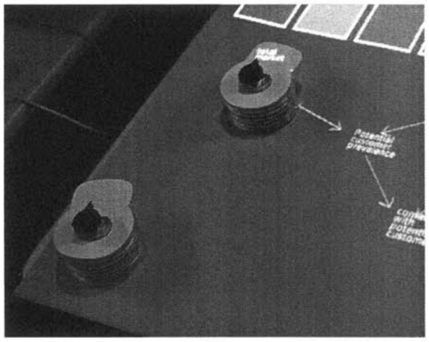

Figure 4-3: The puck at the top of the image is bound to a parameter in the simulation. The puck at the bottom is unbound.

To unbind a digital item from a puck, one uses a shaking gesture. This approach is appealing because the visual effect seems to suggest that the physical forces being applied to the puck are breaking the bond between it and the digital item. However, when first interacting with the system, many users expected that they could unbind a puck from its associated digital information by picking the puck up off of the sensing surface and placing it down on top of some other digital item on the surface. While this is quite a reasonable expectation, our Wacom-based prototype has difficulty differentiating the removal of a puck from the sensing surface from a puck switching itself on and off as part of the time-sharing scheme the prototype system uses. The second generation prototypes of the system include the ability to detect when objects have been lifted off of the sensing surface, so we intend to explore the "paperweight" metaphor offered by this technique in the future.

I wanted to make it easy for users to attach and detach the pucks to and from digital items in the system. But in doing so, I did not want to complicate the metaphor that the puck was a physical embodiment of the data itself. Initially, the software projected information about the corresponding digital content in front of the pucks on the table. This led one user to comment that pen or wand shaped objects might make more sense for manipulating the data, because they would not obscure so much of the information in front of them on the table. This comment suggested that the user was not treating the puck as a physical embodiment of the digital data. At the suggestion of a test user, I experimented with projecting information about the puck onto the puck itself, (as seen in figures 4-4 and 4-5) rather than in front of the puck. This change cleared up some confusion about what the pucks represented. I am interested in exploring other methods of displaying information about a digital items' state on the puck itself. One such approach involves a fold-down display, which is described in the continuing and future work section.

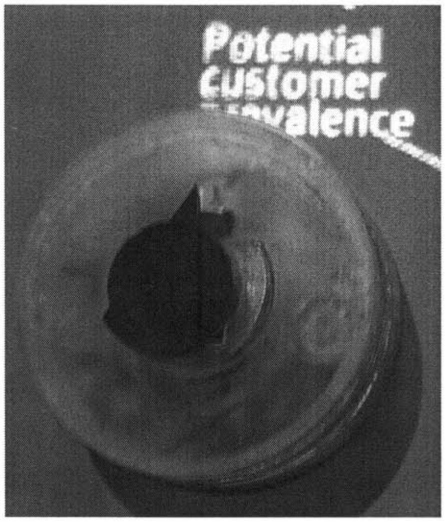

Figure 4-4: The initial strategy of projecting information about a node in the graph in front of the corresponding puck.

4.1.2.2 Use of Dials

In the system dynamics application, users can employ the dials on top of the pucks to adjust parameters in the simulation, as seen in figure 4-6. Users liked the idea of being able to physically manipulate simulation parameters in this manner. However, when using an early prototype of the dial functionality, users had two criticisms. First, they wanted graphical feedback on the sensing surface about the value of various parameters over time. The feedback displayed on a screen behind the surface was not sufficient. Second, they wanted graphical feedback near the dials themselves to provide a better sense of what the dial setting was at a particular point in time. After I made these changes, one could use the dials by focusing just on the table surface itself, rather than having to divide one's attention between the input on the sensing surface and the output of a rear display screen.

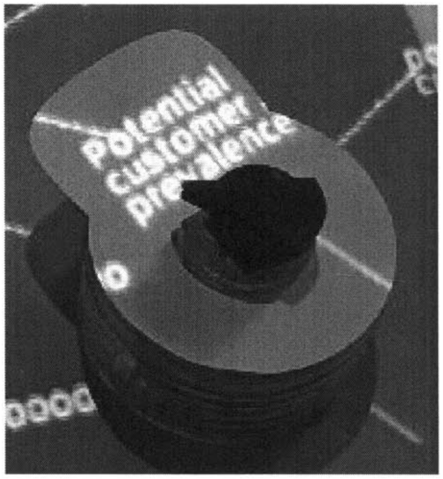

Figure 4-5: Projecting directly onto the puck itself.

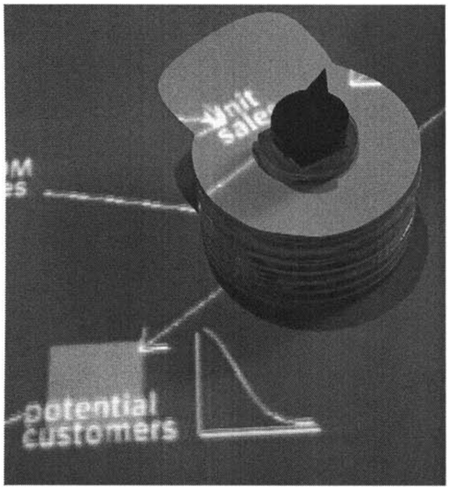

Figure 4-6: A graph of "potential customers" as a function of time. This graph is updated as the "unit sales" dial is adjusted.

In previous research there has been little exploration of this approach to physically modifying computational parameters. The AlgoBlock [35] system allowed children to adjust simple computer programs by rotating knobs on top of physical bricks. However, each of these dials was permanently attached to its corresponding brick, and could only modify one program parameter. The Sensetable project involves the use of dials and modifiers on top of the pucks in a more dynamic role.

4.1.2.3 Tangible Visualization Techniques

At times, users may wish to interact with more data at one time than can be legibly displayed on the sensing surface. In the context of the system dynamics application, I have explored several techniques to deal with this issue. The first is a layout algorithm which adjusts the prominence with which objects are displayed on the table. Each digital item is assigned an importance according to a "scoring process" based on application specific criteria, and the model is searched for any items that overlap with each other. When a pair of overlapping items is found, the one with less importance is darkened to the point where it is still barely visible, and the graphical information associated with the other item is much easier to read. The faint presence of an object provides the user with a cue that more information is available there, so he or she can focus on it using the techniques described below.

Indicating center of attention

While pucks are primarily used to move and manipulate digital items on the table, one can also use them to indicate interest in a particular region of the table. Using the scoring process described above, digital items near a puck recieve higher display priority, and thus become more visible. In a display space crowded with information, this yields a Fisheye [14] like effect where more detail is provided in the areas of user interest. The use of multiple pucks in the interface provides an easy way for the user to simultaneously indicate several areas of interest in the sensing space.

Semantic Zooming

Another technique Sensetable employs to give users intuitive controls over information display is a semantic zooming [31] technique in which the distance between pucks on the table affects the level of detail used to show the information between the two pucks. One example is the abstract graph structure used to represent simulations in system dynamics. Rather than changing the size of individual items displayed on the table, the scoring process described above is used to fade less important items into the background as two pucks come closer together. Nodes are faded into the background when they begin to interfere with the display of a more important node. Figures 4-7 and 4-8 show this interaction. With this approach, one can show different parts of the model in different levels of detail at the same time on the sensing surface. In contrast, related approaches such as the metaDESK [36] display information at only one level of detail at a time. While the metaDESK example involves displaying information with a very literal interpretation of space such as a map, the system dynamics application involves the use of this technique for physical navigation of digital data with no inherent spatial component.

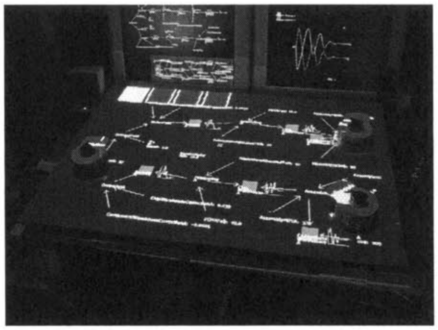

Figure 4-7: With the pucks spread apart from each other, the user gives equal display prioprity to all parts of the graph on the table.

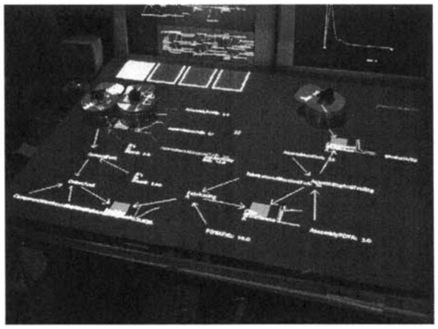

Figure 4-8: Here, the user moves two pucks closer together to collapse the region of the graph between them into a smaller display space.

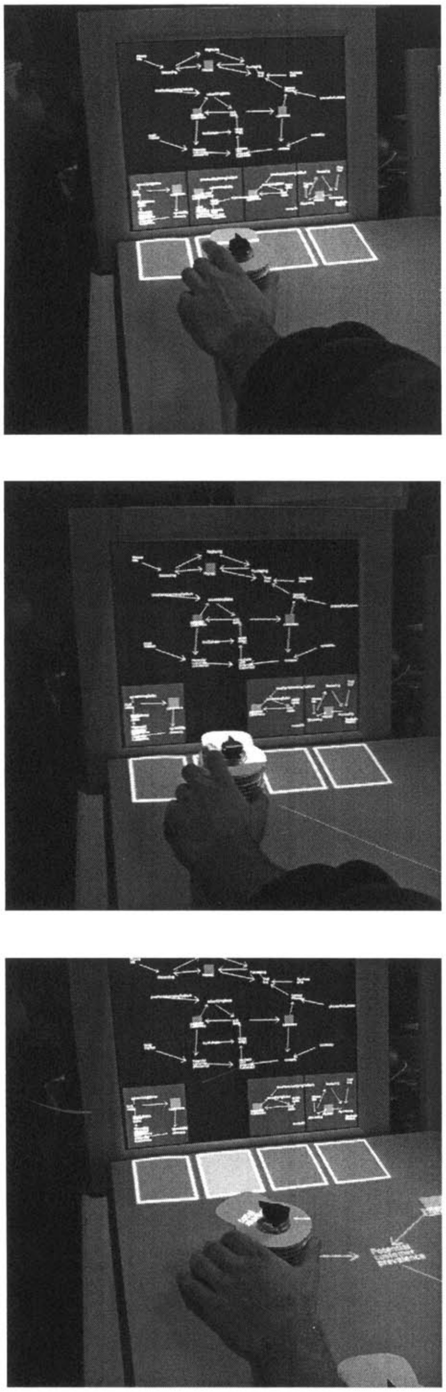

Figure 4-9: The process of moving information from the screen to the tabletop.

Sharing information with an on-screen display

During the process of developing the system dynamics application, it became clear that for larger system dynamics models, users would need the ability to work with a portion of a model on the table. It also seemed clear that for some tasks, a user might wish to share data between the tabletop interaction surface and an on-screen display in order to use tangible and WIMP interaction techniques together. Using the notion of a spatially continuous workspace, I have explored a method for this type of data sharing using Sensetable's physical, tracked objects as the means of transport and control. A flat panel display is aligned with the left side of the rear of the sensing surface so that the display area of the flat panel begins where the display and sensing surface of the tabletop ends. Digital information that can be moved between the screen and tabletop space is displayed in boxes along the lower edge of the flat panel display, as seen in the top image of figure 4-9. The top portion of the rear display shows a higher-level view of the information for context. Directly below each of these boxes is a corresponding box projected on the sensing surface itself. When a puck is placed in one of these boxes, the contents of the corresponding on-screen window "slide" down onto the tabletop, highlighting the box with the puck inside it, as seen in the middle image of figure 4-9. Once the contents of the box have moved into this small portion of the tabletop space, the puck that is now bound to these contents can be used to move and manipulate them on the table, as seen in the bottom image of figure 4-9. As the puck is moved, the contents expand to fill a larger part of the tabletop interaction space in a spring-like motion.

There is some previous work involving spatially continuous workspaces which include on-screen WIMP interfaces. The mediaBlocks system [37] provides a method for moving data between a physical container and an on-screen WIMP interface which involves placing a tagged wooden block in a socket on the side of the screen. More recent augmented surfaces work [33] adds the notion of a spatially continuous connection between the screens of portable computers and nearby tabletops and wall surfaces. In this work, users can employ their mouse cursor to move objects to and from the physical world. Data can be associated with physical objects, but only with the mouse cursor.

4.2 Chemistry

Figure 4-10 shows a tool built on top of Sensetable for teaching students about chemical reactions. This was the first proof-of-concept application developed on top of the Sensetable platform. In this application the user can map the pucks to atoms or molecules, and then move these around in the workspace. When the atoms and/or molecules which are needed for a particular chemical reaction are brought into close physical proximity, the reaction occurs. The user can then manipulate the reaction products to use them in other reactions. The user can place modifiers on top of the pucks to change the electrical charge of the atom or molecule.

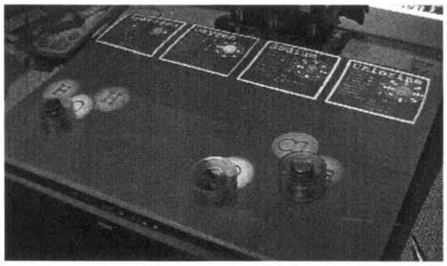

Figure 4-10: A chemistry application running on top of Sensetable.

There are four slots for mediaBlocks [37] along the rear edge of the table. When a mediaBlock containing a certain atom from the periodic table is placed in one of these readers, a visual representation of the atom and various information about it slides out of the block and onto the display surface as described in the original mediaBlocks work. Once this information appears on the rear surface of the table, the user can place a puck on top of the periodic table entry to pull an atom of that type onto the workspace.

Atoms and molecules that are no longer of interest to the user can be removed from the workspace by dragging them to a portion of the table which contains atoms not currently in use. If this space becomes full, an atom or molecule is removed to make room for others. The user can also store molecules in mediaBlock-like objects using a physical receptacle on the side of the surface. These can then be brought back into the system for later use, or potentially transported to other environments, such as an on-screen GUI for further study.

This application currently has a simple model of the requirements necessary for a chemical reaction to take place. The system has a set of rules which specify reactions that typically take place at room temperature and pressure. If all of the reactants required by a certain rule are present, the reaction occurs. I am interested in implementing a more sophisticated model which considers environmental factors such as temperature and pressure when deciding which reactions will occur. The user might control these factors by setting them to values which apply to the entire workspace. Alternatively, they could be controlled on a more local scale. For example, a puck could be bound to a "Bunsen burner" which could then be moved throughout the workspace to add kinetic energy to different parts of the system.

4.2.1 Binding Content to Pucks

The chemistry application employs several approaches for binding content to pucks for different circumstances. When a user wishes to bind a puck to an atom or molecule in the workspace, the user simply places the puck on top of it. Users expected this operation to work when using the interface for the first time. Most of them tried it without it having been explained or demonstrated to them. To bind a puck to a new atom of a particular element, the user can place the puck in the panel at the top of the interface, and then move the puck away from the panel, as shown in figure 4-11. This is similar to the tool trays used by Fitzmaurice in the GraspDraw application [9] except that the content of these panels can be dynamically changed.

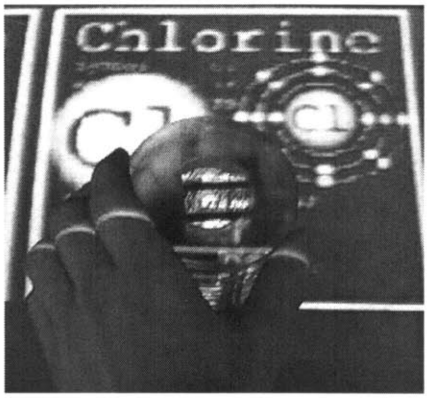

Figure 4-11: The user places a puck onto the chlorine panel to "pull off" a chlorine atom. The new atom is then bound to the puck.

Another manual binding approach I have explored is using the modifier on top of the puck to specify content. With this approach, pucks do not have any content associated with them unless a modifier is attached. This approach did not seem well suited for the chemistry application, in part because it was difficult to use this method of binding in concert with any software controlled automatic binding. Because the software cannot add and remove modifiers from the pucks, any software controlled rebinding would be inconsistent with the modifiers. As well, without an extra layer of abstraction the number of modifiers available provides a limit on the number of digital objects that can be represented.

In addition to these manual approaches to binding, the system also employs automatic binding when a reaction occurs. The user brings the necessary atoms close to each other to cause a reaction. The products of a reaction are automatically bound to the pucks which were originally associated with the reactants. This interaction is illustrated in figure 4-12. Some molecules or pucks will be left unbound after the reaction if the number of products is different than the number of reactants.

4.2.2 Unbinding Pucks

The chemistry application provides two ways for pucks to be manually unbound from digital content. The user may drag the content to the "recycle bin" region of the workspace, in which case it will be unbound from the puck and saved in case the user wishes to retrieve it later. Or the user may drag the atom or molecule to a graphical panel on the side of the workspace. This panel represents the contents of a physical container of molecules which can be carried to computing environments outside of Sensetable (such as a GUI). Here the content will be unbound from the puck and placed in the panel associated with the external container. In addition, I have employed the shaking gesture described in the system dynamics application to unbind pucks.

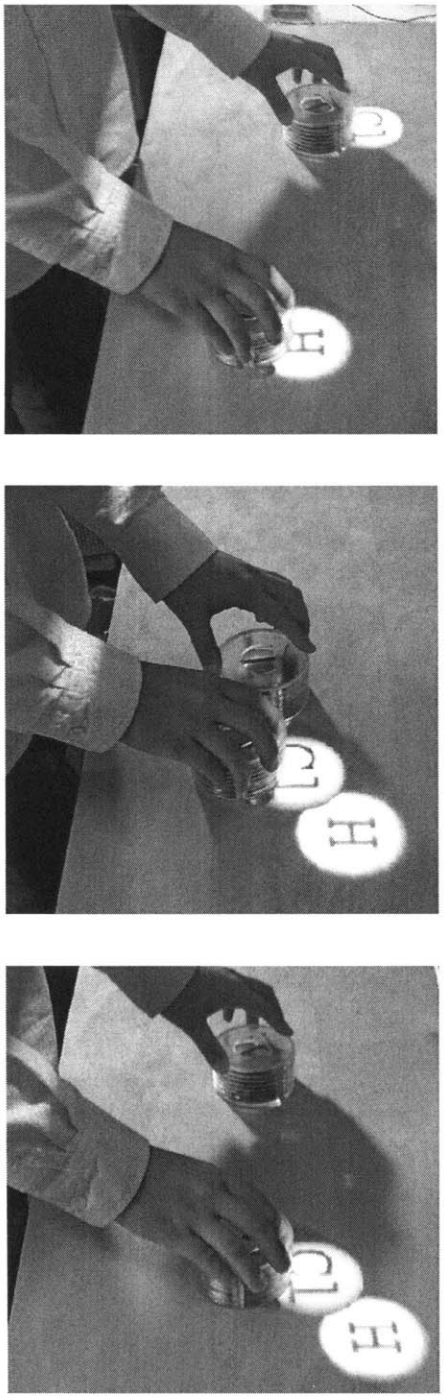

Figure 4-12: After a reaction takes place, the system automatically rebinds the products of the reaction to pucks involved in the reaction.

4.3 Abstract Visual Form

Another exciting application domain for Sensetable is interaction with abstract visual representations of computational form. Figure 4-13 shows one such interaction. The Aesthetics and Computation Group and the MIT Media Lab, led by John Maeda, has done a lot of exciting work exploring various dynamic visual forms generated with the aid of computation. Much of the work done by this group has been interactive in nature, but in most cases, this interaction has used a standard keyboard, mouse, and desktop display. While this group's work has shown that many compelling interactions are possible using a standard keyboard and mouse, I believe that Sensetable can offer some unique possibilities for interacting with dynamic visual forms.

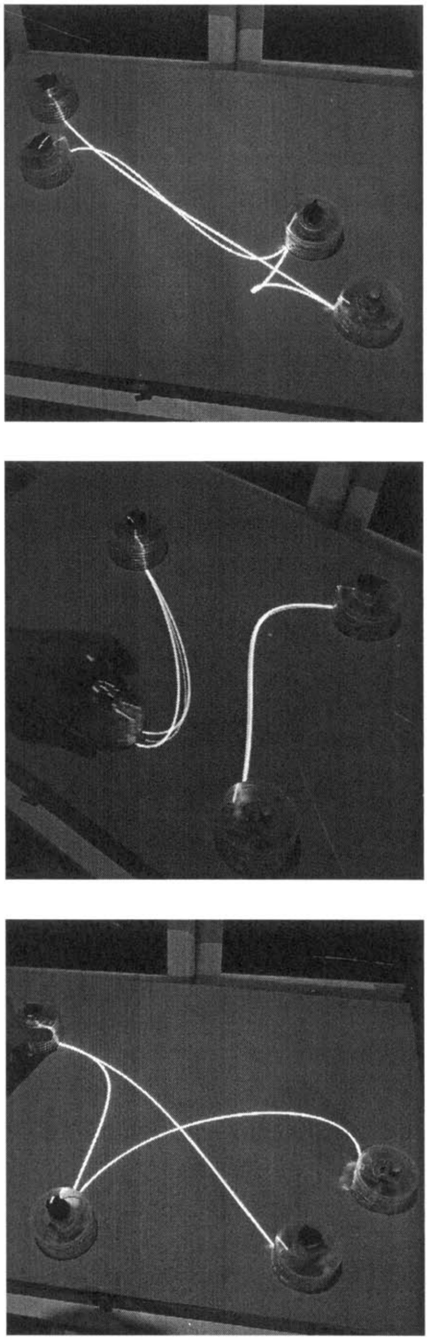

Figure 4-13: Some interactive visual forms on the Sensetable platform.

One preliminary exploration in this area is a series of flowing lines which connect the pucks on the Sensetable surface. The shape of these lines reacts when the position or orientation of the pucks is changed. When two pucks are brought within close proximity of each other, the curve connecting them is removed if it is present, or created if it is not present.

From a computational perspective this is the simplest application that has been implemented on top of Sensetable. However, many people who have seen the system have commented that this application creates the most compelling interaction they have seen on Sensetable. People interacting with it enjoy spending time exploring different ways of arranging the pucks to create visually pleasing shapes. Several features seem to add to the appeal of this application. First, the input and understanding required from the user is very simple. The only "mode" present in the interface is whether a line is being drawn between two objects or not. The user does not need to learn sophisticated interaction techniques to use the system. Second, the system produces compelling but sometimes surprising output as a result of the user's input. It is often difficult to predict what the lines will look like if the pucks are in a certain position, but once the pucks are placed in that position, the result does not confuse the user due to the simplicity of the interaction.

4.4 Music Applications

I have also explored the use of Sensetable in the domains of musical performance and composition. Tangible interfaces have a lot of qualities which make them appealing for these domains. Musical applications often require precise control of parameters, and careful timing in order to produce the desired sounds. When compared with Graphical User Interface techniques, tangible interfaces can provide better control of many parameters at the same time, because they allow for bimanual manipulation, and do not require the user to constantly remap a physical control (such as a mouse) to between many graphical controls. The use of multiple physical objects to represent multiple aspects of digital state also allows for persistent physical representation of digital state. This may help users understand and adjust that digital state quickly and more intuitively. Something else that makes the exploration of musical applications using Sensetable exciting is that the music industry has created a variety of musical controls and input devices which could be considered tangible interfaces. Figure 4-14 contains one example. The sliders and knobs on

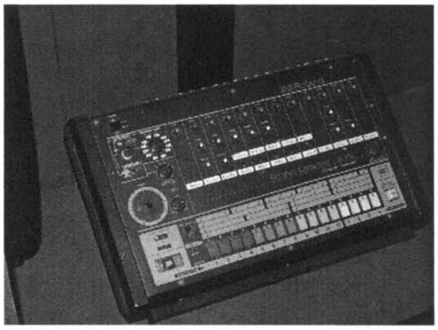

Figure 4-14: A Roland 808 drum machine. The controls at the bottom each consist of a button and an LED, providing an integration of input and output spaces. As well, the many knobs at the top of the machine provide a physically manipulable and persistent representation of the machine's internal state.

this mixer serve as input mechanisms for adjusting the state of various parameters, but at the same time they represent the current setting of the particular parameter. Some mixers even have motor-actuated faders so that the computer can modify the slider position to keep it consistent with the digital setting of a parameter, should the digital setting change through some other means. As well, these mixers often use separate physical representations for each parameter. The demand for and development of interfaces that incorporate principles of tangible interface design in the music industry suggests that musical applications can be a good way to test and develop tangible interfaces. As well, ideas developed to meet the demands of musical applications might be interesting when used in tangible interfaces for other applications.

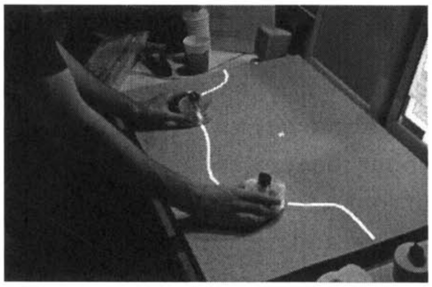

Figure 4-15: A user physically manipulating a waveform to produce a sound.

4.4.1 Wavetable

The first application I developed is called Wavetable. (Thanks to Ben Recht of the MIT Media Lab Physics and Media Group for this name.) In this application, a sound wave is drawn horizontally across the table, as shown in figure 4-15. The wave is initially flat, but the user can place pucks on the table to manipulate the shape of the wave. In the first version, the system distorted the sound wave according to an approximation of a Bezier curve, but it turned out that using linear interpolation of the sound wave produced a more pleasing sound. As the user adjusts the shape, the sound wave is being continuously played through the soundcard of the computer. Currently the wave is played through the speaker at a rate of 261.63 Hz, or middle C, but the intent of the project is that a MIDI keyboard can be connected to the

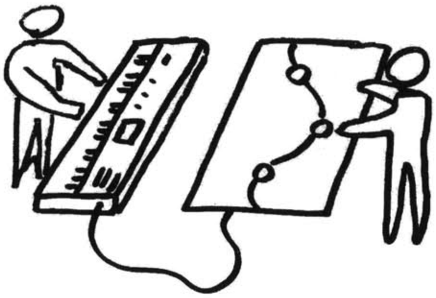

Figure 4-16: Concept sketch of users producing music collaboratively using a keyboard and Wavetable.

computer, so that the keyboard can control the pitch of the notes while the table controls the tamber of the notes. In this way, the keyboard and table could be used by two people collaboratively. I am also excited about applying this method of interaction to other synthesis techniques, such as scanned synthesis. [40]

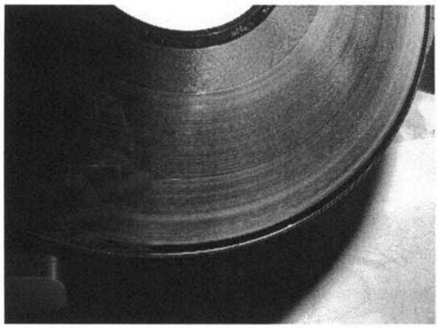

Figure 4-17: The grooves in a record provide visual feedback about the start and end points of songs, the type of rhythm in a song, and the location of loud and quiet parts within the song, among other things.

4.4.2 Disc Jockey Application

The second musical application developed using Sensetable is a system for dynamically mixing different audio tracks in real-time to produce a collage of sound. Today, disc jockeys often use a laptop or a pair of record turntables and a mixer to perform this task. Each of these interfaces has some advantages and disadvantages. Turntables provide precise physical control. As well, one can look at the surface of a record and visually develop an understanding of what the record sounds like in different parts, as shown in figure 4-17. This makes searching for a particular portion of a song on a record much easier than without visual feedback. Another interesting feature of record turntables is the mechanism for navigating through the linear stream of audio on the record. One can move through the sounds on the record in a linear manner with fine grain control by rotating the record in the desired direction underneath the tone arm. One can also move the tone arm itself for coarser grained control. However, turntables are limited in the flexibility of the types of sound they can produce. Laptops are more flexible in this regard because sound generated via software synthesis can be mixed with pre-

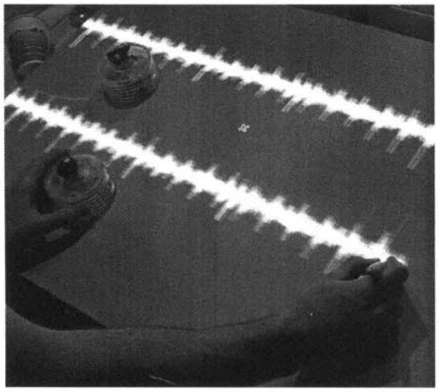

Figure 4-18: In this application, the user receives real-time visual feedback about the upcoming portions of a song.

recorded sound. However, laptops are quite limited in terms of the precision of control they provide to the user, because they usually rely on a GUI in a musical performance setting.

In the Sensetable DJ application, the user employs a Sensetable puck to navigate through a tree structure representing a group of songs the user might want to play. This hierarchy corresponds to a file and directory hierarchy in the computer. Once the user has found a song he or she would like to use, the user selects it by holding the puck over it. The system loads the audio track, and presents the user with a graphical representation of what the song sounds like, based on the digital sound data. This representation is shown in figures 4-18 and 4-19. The user can press a button on top of one of the pucks to play or pause the corresponding audio track. The user can rotate the puck to adjust the position within the audio file. While the track is playing, the adjustment is a very fine-grained one. The adjustment is much coarser when the track is not playing. Users can also visually "stretch" or "compress" an audio track using the Wacom pen along with the puck corresponding to the audio track. While the user is performing these operations, he or she receives visual feedback on the table about the content of the audio tracks he or she is mixing together. This visual feedback helps the user perform what DJs refer to as "beat-matching", the process of synchronizing the rhythms of two or more songs by carefully adjusting their tempo. This strong visual feedback makes it possible for a novice to beat-match two songs in under twenty seconds. In contrast, a novice using two record turntables for this task might need several attempts over several minutes of time before being able to match the beats of two songs.

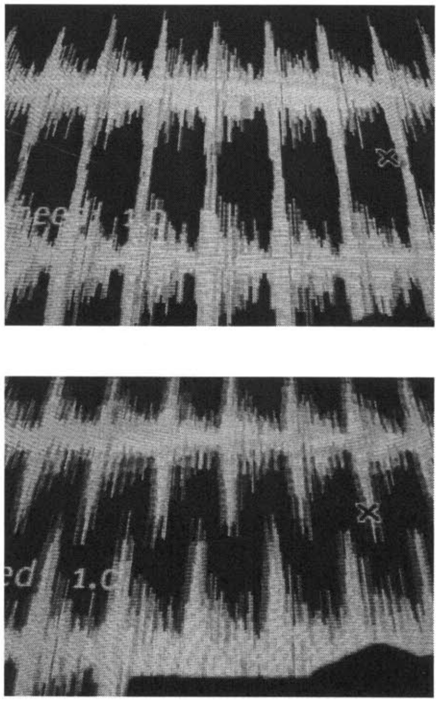

Figure 4-19: In the top image, the beats of the two songs are not synchronized. Thus, the peaks in the sound wave are not vertically aligned. In the bottom image, the user has visually aligned the peaks, and thus synchronized the rhythms of the two audio tracks.

4.4.2.1 Navigating an Audio Stream

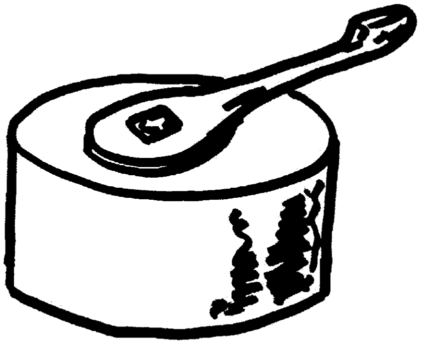

Figure 4-20: A specially shaped modifier for the Sensetable DJ application. The modifier includes a button on top, and an attachment which makes it easy to spin the puck with one finger.

This application is the only Sensetable application in which pucks are bound to linear streams of media. For this application, turning the puck to move forward and backward within a stream of audio seems to work well, based on very preliminary evaluation with a DJ in the Media Lab. The approach of using fine-grained position adjustment when the song is playing and coarse-grained adjustment when the song is not playing yields a similar interaction to that provided by a record turntable: When the needle is picked up to make a coarse grained adjustment of what part of the song is playing, the music stops. However, fine grained adjustments can be made while the record is playing by rotating the platter with one's hand. One issue with this technique of using puck rotation to index into the audio stream is that it could be tedious at times to rotate the puck many times if one wanted to drastically change the position of the index into the song. To address this issue, I plan to explore a physical attachment for the pucks to make rotation easier. As shown in figure 4-20, the user could place his or her finger near this attachment to rotate the puck much more fluidly.

4.4.2.2 Use of Stylus as Tool

Another unique feature of this application among Sensetable applications is the use of a Stylus as a physical tool for manipulating data. In the wavetable application, the pucks were used solely as tools for distorting an audio waveform, but this application did not include physical instantiations of both tools and data at the same time. The distinct physical forms of the puck and stylus may make the manipulation of physical tools and data at the same time more intuitive.

In this application, users can speed up or slow down an audio track by using the puck associated with that track and the stylus as two anchor points with which to stretch or compress the track. When the stylus tip is pressed against a point along the graphical representation of the audio stream and then dragged, the system adjusts the track tempo proportionally to the amount of distance dragged with the stylus. Users found this technique a bit counter-intuitive. This issue might be addressed by using a more specific physical form for the tempo adjustment operation.

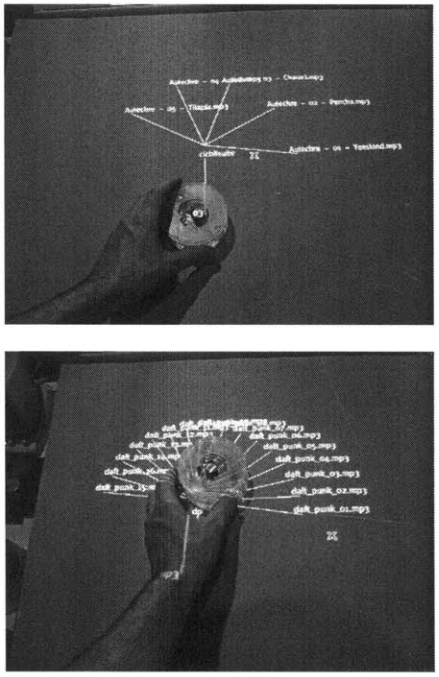

Figure 4-21: Using a puck to navigate a tree of songs. The current node of the tree is centered on the puck. As the user moves the puck near a child or parent node, that node is selected and all of its children appear.

4.4.2.3 Tree Navigation

Another interaction technique I explored in the course of developing this application is a method for quickly navigating tree data structures using a single puck on the table. This interaction is shown in figure 4-21. Initially the user places a puck on the graphical representation of the tree's root node, and moves it to the child node he or she would like to explore. As the user moves the puck toward a particular child node, children of that node also begin to be visible. A particular child node is selected by briefly holding the puck over that node. One can select the parent node of a point in the tree as well. This node is displayed opposite the child nodes around the puck. During the process of navigating through a tree, the user may find that the puck is nearing the edge of the sensing surface. The user can adjust the physical position of the puck without changing its position within the graph by pressing a clutch button on top of the puck.

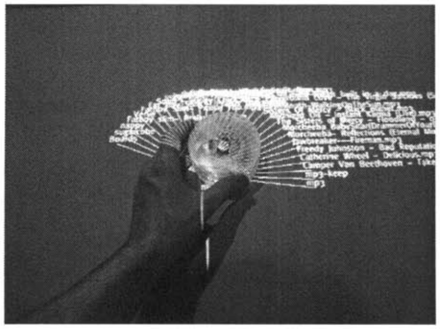
Freedy Johnston - Bad Reputatio
Catherine Wheel - Delicious.mp
Camper Van Beethoven - Tak
mp3-keep
mp3

Figure 4-22: In cases like this, a large number of child nodes can make the selection process difficult. In these cases a more sophisticated selection technique is needed.

One issue with this technique is that when a node has many children, the textual representations of the child nodes may be so crowded together that they are illegible, as in figure 4-22. One approach to address this problem would be to give more display space to a subnode which is near the puck. This way, the user could quickly browse through a crowded list of a node's children by moving the puck near them. As the display space given to various child nodes changes, these nodes necessarily must move around on the table surface. One important detail of this technique is ensuring that nodes the user is trying to select are not turned into "moving targets," making the selection process more difficult. As well, nodes the user is not interested in selecting must not be moved near the puck in a way which causes inadvertent selection. One way to achieve this is to adjust the display space given to a child node only when the puck is a certain distance away. This implies that the space given to the node must be expanded to its maximum value when the puck is approaching, but still not very near that node.

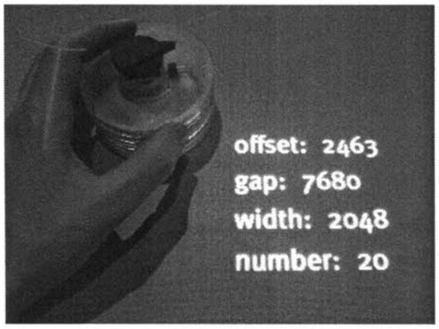

Figure 4-23: A user interacting with the first implementation of the parameter-based sampler.

4.4.3 Parameter-based Sampler

The most recent musical exploration on the Sensetable platform is an application which produces abstract musical patterns by pulling short samples from portions of existing audio tracks. This work is a collaboration between Ben Recht of the Physics and Media Group at the MIT Media Lab and myself.

Short samples from an existing audio track are played in rapid succession as the user adjusts parameters in the sampling process such as the number of samples, the length of each sample, the time interval between samples, the location within the initial audio track where the sampling begins, and what type of filtering to apply to the output, if any. One complaint some musicians have had about this genre of electronic music, known as "glitch" music, is that the use of computer algorithms for much of the sound production takes away the opportunity for real-time improvisation and performance. The parameter-based sampling application on Sensetable attempts to allow one more control over the sound production process, perhaps enough to be suitable for improvisation in a musical performance.

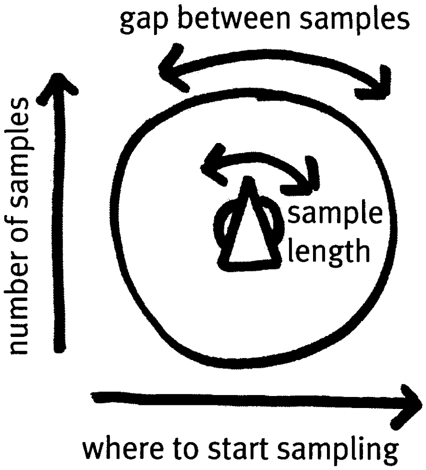

Figure 4-24: The initial control mapping for the parameter-based sampling application.

To experiment with the concept, we initially used a very crude mapping between the various attributes of a puck and the parameters of the sound generation algorithm. This mapping is shown in figure 4-24.

While this mapping helped us realize that we could produce interesting sounds using this method, it was very difficult to remember and use. I experimented with different mappings, but it seemed inherently difficult to map these parameters onto the cylindrical form of the pucks used in Sensetable at the time. A new puck design, which provides a more intuitive physical representation for what parameters various aspects of the object actually control would be useful for this application. Figure 4-25 shows a drawing of a potential new puck and mapping. Here, the use of projection on various portions of the puck could reinforce the association between manipulations of different parts of the puck and effects of different

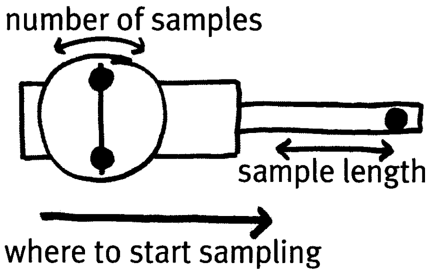

Figure 4-25: A puck design with an application specific physical form. This design is based on Zowie tags built into a larger physical structure. The "sample length" tag on the right is attached to a bar which slides in and out of the puck. Thus the physical length of the tag grows with the sample length.

sound generation parameters. One interesting aspect of Sensetable which this application illustrates is that new physical forms can be designed and built quickly to tailor the interaction with Sensetable to the needs of a specific application or set of users.

These applications demonstrate that Sensetable can be applied to a wide variety of application domains. We have talked to sponsors of the MIT Media Lab about other application domains including military command and control, multiplayer games, graphic design and others. In addition, most of the interaction techniques presented here are difficult to use with other sensing technlogies because of issues with tracking speed, accuracy and occlusion. While I have presented the interaction techniques here in the context of specific applications, the techniques themselves are general enough to be applicable to a larger variety of applications. In the next chapter, I discuss the evaluation of these techniques in the context of the system dynamics simulation application.

5. Evaluation

During the design and development process of the system dynamics application I asked people with varying levels of system dynamics experience to use the system. Their experience ranged from being a professor conducting research in system dynamics to having only a cursory knowledge of the field. Some of these tests were conducted with pairs of users working together, while others involved a single person using the interface while giving us verbal feedback about it. I conducted ten of these sessions that lasted from 30 to 60 minutes. Eight users participated in these tests, with several trying the interface at two or three stages of the development process.

Initially, users reported having difficulty analyzing models with more than 25 nodes in the system. They commented that the automatic graph layout algorithms in the system removed some of the information that was encoded in the original layout of the system dynamics model. The person developing a system dynamics model usually carefully designs the layout of the nodes in the graph so that important causal loops in the model can be readily identified and studied. By adjusting the layout of the graphs on the sensing surface, the software often removed some of this information.

After discovering this problem, I began to investigate other methods of dealing with limited screen real estate. Current on-screen system dynamics simulation packages address the problems stemming from limited screen real-estate by breaking up the model into a larger number of "views," each of which displays a certain feature of the model. One can switch between these views using a menu. This approach to interacting with smaller portions of a system dynamics model at a time suggested the method of sharing data between the screen and tabletop portions of the interface described in the interaction techniques section. The use of this technique in the system dynamics application is shown in figure 8.

Once the system provided a static graph layout on the rear context display while allowing the user to manipulate the layout of a portion of the graph on the tabletop surface, users had an easier time of using models of around 25 nodes in size. However, with graphs that were closer to 50 nodes, the dynamic representation of the graph on the table could get so different from the representation displayed on the rear-screen that users reported having a hard time finding specific parts of the model. Clearly some more sophisticated visualization techniques are necessary to make this approach scale well. Some possible approaches are showing higher-level structures in the model such as cycles rather than individual nodes and edges. The user could then select an individual cycle to get a higher detail representation of it. As well, there is still room to improve the way screen real-estate is divided among a group of nodes depending on where the pucks are. For example, one might display the text labels of the nodes at a relatively low resolution until a puck was near, and then the font size could increase. Another strategy would be to use the author's layout information on the table as well as the rear context display. When the user attached a puck to a node of the graph and then moved the puck, the graph could deform slightly to keep the graphical representation of the mapping intact, while retaining the overall context of the graph at the same time. Finally, one could display a completely static representation of the graph on the table, and rely on the author to develop a coherent layout. With this approach, binding and unbinding could be performed based on the proximity of a puck to a bindable item on the table.

5.1 Setting Parameters with Dials

One of the comments users had during testing was that the dials on top of the pucks often did not have enough precision to obtain desired parameter setting in the System Dynamics simulation application. I was aware of the low precision of the knobs before the testing started, but did not believe the precision would be an issue, because this method of simulation focuses more on isolating overall trends of behavior rather than making specific numerical predictions. However, it turned out that in the Sensetable 1.0 implementation the precision of the dials was in fact a limitation. One approach to addressing this is using the orientation of the puck itself as a mechanism for setting the value of a parameter. This approach has two clear advantages. First, because the puck can be rotated infinitely in either direction, there is a great deal of flexibility in how the application decides to map the orientation to the value of a particular parameter. Second, the removal of the physical dial on top of the puck would solve the problem described earlier in which a puck is bound to a parameter with its knob off-center, causing an inconsistency between the physical and digital representations of the parameter value. However, we have not yet implemented this technique in this application because there seem to be some problems with it. First, it is difficult to move the puck in the X-Y plane without rotating it slightly. Some research suggests that in cases like this, where two related input dimensions are tied to two unrelated input parameters, performance in setting the parameters to desired values decreases. [20] In other words, binding a parameter setting to rotation might be a difficult mapping for one to use. A second issue with this technique might be a poorer physical affordance for how to adjust a parameter. Nevertheless, I intend to implement this approach to parameter setting to see which method users like better.

5.2 The Need for More Simulation Feedback

Another comment users of the system gave is that they were interested in seeing graphical plots of any parameter in the simulation, rather than just the stocks. As a first approach to addressing this, I provided a Wacom stylus which could be used as a "magic wand" to tap the projections of various parameters on the tabletop surface. This tapping would cause the graph of a parameter that was on the table to be removed, or if one was not there, it would be displayed. The users who requested this feature reacted well to this implementation of it in a later testing session, but the feature has undergone relatively little testing thus far.

5.3 Binding and Unbinding

Users also found that the process of binding and unbinding pucks to data was difficult. Users would occasionally say things such as "oh, I did something wrong." or "Did that work?" when they were trying to bind or unbind pucks. Part of this problem is certainly due an inflexible implementation of this part of the system. Even with more flexible shake-detection routines though, I expect users would still have some problems. Pucks were occasionally bound accidentally even when the user was not touching them. A portion of the graph might move around and come to rest on the unbound puck. If the node remained on top of the puck for a certain amount of time, the puck would be bound to the node. The user might not see this happen if his or her attention was focused elsewhere. However, if the dial on top of the puck were adjusted off center, this binding operation might have an unpredictable effect on the simulation results.

In addition to accidental binding, pucks were occasionally "shaken loose" from their corresponding digital parameters due to sensing artifacts at the edges of the sensing area and near the gap in the middle between the two boards. Again, users did not always notice when this happened. As well, even when they did notice it, they didn't understand why it had happened unless it was explained to them.

5.4 Problems Caused by Sensing Errors

While many of the issues described above might be addressed with more informative graphical feedback to the user, I believe some of them point to a key design challenge in sensing techniques for tangible user interfaces. When a user interacts with a graphical user interface, the objects he or she is manipulating exist in the screen. As such, the user can not expect these objects to be bound to the laws of the physical world. The computer may choose to make these objects move, change or disappear without advance notice to the user.

Thus, the user knows that when he or she manipulates the objects on the screen, the result happens according to rules determined by software. In contrast, the physical manipulation which happens in a tangible interface is closer to the laws of the physical world. If the user moves or manipulates a physical object to accomplish a computational task, the user may assume that the position and state of the object is obvious (and valid) to the computer, just as it is obvious to him or her. However, with current sensing technology, this is not always the case. If the computer is not able to accurately sense the actions of the user, several bad things may happen. A technically savvy user may realize that the system did not sense his or her actions. From this point on, the user must think about the physical objects in the system not as physical embodiments of data, but as "an unreliable mechanism for controlling the data." In other words, the underlying structure of the interface becomes exposed in a way which may negate some of the benefit obtained via the simplicity of interaction in a tangible interface.

In the case that the user does not understand that the sensing mechanism has failed, the user will be quite confused. He or she may then reject or question his or her mental model of how the interface is supposed to work, because the interface is not behaving as expected. Graphical interfaces have mechanisms for communicating the user that things are not operating the way they should be. An hourglass cursor may appear, or a dialog box which indicates that there is trouble. However, tangible interfaces do not have these signals yet. I believe novice users may come to tangible interfaces with a higher series of expectations for the underlying software and hardware due to the lack of these cues. When the interface does not perform as expected due to problems with the sensing hardware, the user may simply assume that he or she has made a mistake.

This tendency has two implications. The first is that for tangible interfaces to be truly successful, we must think of ways to communicate to the user that things are not functioning correctly, without forcing the user to return to the mental model of interacting with a desktop computer. Second, tangible interface designers should favor reliable, mature technologies when designing systems, because failures in the hardware and software of an interface based on physical objects may be very costly in terms of the amount that they confuse novice users.

5.5 Haptic Feedback

Another comment one user had about the system was that she wanted haptic feedback from the dials on top of the pucks. She noted that the system made it easy to change parameters in a simulation model which might in fact be very difficult to change in reality. Her vision was that haptic feedback could be used to immediately let the user know when this sort of conflict occured. While this is an exciting idea, the constraint of current battery capacity makes it difficult to implement in a wirelessly tracked puck. However, I am working on an alternative approach to this problem, which is described in the future work chapter of this thesis.

5.6 Discussion

One surprising part of developing the system dynamics application was the different role of the model layout in on-screen space and in tabletop space. Traditionally in system dynamics models that are displayed in a WIMP interface, the author uses the spatial organization of the model to communicate information about important structures in the graph. For example, loops in the model sometimes cause patterns of oscillating behavior. One usually arranges the nodes in important loops so that it is very clear that the nodes form a loop. Thus in a WIMP context it can often hinder the process of analyzing the graph to adjust the layout of nodes from their original positions. However, there are also benefits one may achieve from adjusting the layout of the graph. Reorganizing the nodes may make a problem solving process easier by allowing the user to offload computation from his or her mind to the environment, as discussed in the related work section. For example, if one wanted to determine which among a group of nodes had an oscillatory effect on a parameter in the simulation, one might arrange the nodes to be tested in a line, and then adjust the dial on top of each corresponding puck in sequence and see what happened in response. As one tested each node, one might sort the nodes into two groups on the table depending on whether they contributed to the oscillation or not. At the end of this process, the arrangement of the nodes on the table would hold the answer to the original question, without any need on the part of the user to memorize or write anything down during the process.

By providing a static layout of the graph on the left rear display, and a dynamic, manipulable version on the tabletop, I believe Sensetable provides some of the better aspects of both interface styles for the problem domain of system dynamics. The screen provides a frame of reference for the analysis going on, and the tabletop allows the user to look at and manipulate a more manageable portion of the model during the process of analysis. In general, this seamless connection between the screen and tabletop allows one to move pieces of digital content to whichever space is best suited for the task at hand. While the current connection makes little use of the keyboard and mouse, we expect that as the Sensetable applications continue to mature, the keyboard and mouse will be quite useful in the graphical portion of the interface for tasks that are not done well in the tangible part of the interface.

5.6.1 Why Tangible?

The user testing suggests that Sensetable may provide several benefits over traditional GUI-based techniques for analyzing system dynamics models. First, the ability to manipulate the physical dials and see real-time feedback about the change in simulation results was very exciting to users. They enjoyed being able to use both hands at the same time to adjust two different parameters simultaneously. One commented that this approach helped him "develop an intuition more quickly" about what the model would do. This interface often involves one less level of indirection between the human hand and the actual computational change taking place than does a mouse adjusting a slider [9]. I hypothesize that as the application matures, this ease of manipulating parameters may lead to more thorough analysis of models, which may in turn lead to a better understanding of the model's behavior. My experience with users thus far is suggestive, but not sufficient to evaluate this claim.

The fact that Sensetable affords collaboration between users is also important. Instead of collaborating verbally while one person adjusts parameters with a keyboard and mouse, Sensetable allows different people to change parameters simultaneously. For example, this feature would be useful if managers of separate manufacturing plants owned by a company wanted to look at how various changes in their respective plants' production would affect the company as a whole. Each could control the parameters associated with his or her factory while observing the aggregate effect on the company.

In summary, preliminary user testing suggests that the design prinicples behind Sensetable platform are sound, but that I must do more work both to perfect the interaction techniques and to develop more informative graphical feedback to let the user know when unexpected things happen due to problems sensing the users actions. As well, further improving the performance of the sensing hardware will help reduce the likelihood of these problems.

6. Conclusions and Future Work

I have presented Sensetable, a platform for tracking multiple objects wirelessly on a flat surface with high accuracy and low latency. The use of an electromagnetic sensing approach frees one from the problems typically associated with computer vision based approaches to object tracking. These include occlusion, susceptibility to variations in lighting, and higher latency. In addition to overcoming these issues, our sensing approaches allow one to give the tracked objects state which can be physically manipulated with controls such as dials and modifiers.

Using this platform, I have explored some new interaction techniques including changing the distance between pucks to control the amount of information displayed between them, using pucks to indicate points of interest for a "fish-eye" like approach to displaying crowded graphs, and using gestures to bind and unbind physical pucks with digital content.

Among other applications, I have implemented an application on top of Sensetable to analyze system dynamics models. Users familiar with system dynamics tested the interface during the development process. For them the most valuable part of the interface was the ability to quickly adjust multiple parameters using the dials and see real-time feedback. While users also valued the ability to move the nodes around using the pucks, they found the association between the pucks and nodes unclear until I began projecting the names of the nodes onto the corresponding pucks themselves.

During the process of developing the system dynamics application, I developed a workspace that included a seamless interface between display screen and tabletop components. The rear display screen preserves the original structure of the system dynamics model and provides a reference frame for the investigations performed using the tangible component of the interface. On the other hand, the tangible component allows the user quickly to investigate the effect of parameter changes on the model, and to reorganize portions of the model in support of this investigation.

The most important result of the Sensetable project thus far is that from the standpoint of technology and of interaction design, electromagnetic sensing platforms that track multiple objects wirelessly are a viable and interesting tool for building tangible interfaces.

6.1 Continuing and Future Work

I am currently working on developing several aspects of this work more thoroughly. In the area of interaction techniques, I plan to continue the investigation of how Sensetable can be combined with other approaches to the user interface, such as WIMP, speech based interfaces, etc. Hopefully research in this direction will lead to interfaces which can solve problems that cannot be readily solved using just a single one of today's predominant approaches to the human-computer interface.

One interesting aspect of this exploration is the use of Sensetable and the two rear display screens as a general workspace in which a user could interact with several applications simultaneously. As a simple example, suppose a graphic designer were using this system to design the cover of a record album. The left rear screen might contain a font design application with a traditional GUI interface. However, if the user wanted to physically manipulate the shape of one of the letters on the font, he or she could execute a GUI command which would make the letter available at the bottom of the screen, were he or she could grab it with a puck and pull it down onto the TUI workspace. The right rear screen might contain a graphic layout program, in which the user was arranging the graphic elements on the album cover, including some text in the font being designed on the left screen. The user might also drag the album cover down onto the table, and use pucks to physically adjust the layout of the cover. If the user were to manipulate letters from the font and the graphic layout using TUI pucks, he or she might be able to quickly iterate on the design of the album cover. He or she could explore how changes in the graphical elements of various letters in the font could affect the aesthetics of the design as a whole, and quickly change the overall graphical layout accordingly.

I am also excited about exploring interaction techniques that relate solely to tangible interfaces. One example is the stacking of modifiers on top of a puck. I anticipate using the stacking of modifiers to allow the user to perform "what if" analysis in a system dynamics simulation. For example, if a certain node represents the population of an animal in a forest, one modifier could mean that natural predators of the animal were removed, another could mean that the population was struck by some sort of disease, and so on. By composing these modifiers on top of the puck representing the animal population, users could experiment with a variety of scenarios

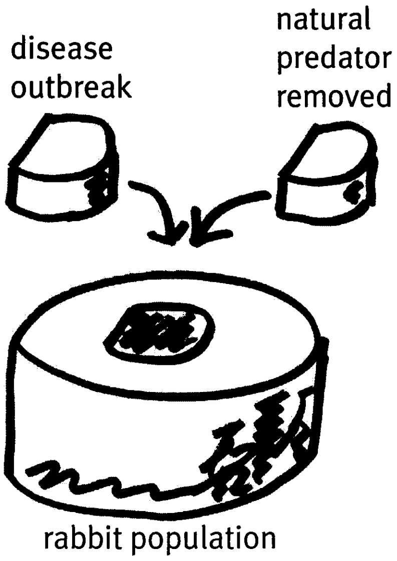

Figure 6-1: A potential scenario in which modifier stacking would be useful.

within a simulation. I have completed the development of the hardware necessary to support this interaction, and am currently working on completing the software necessary to experiment with the technique.

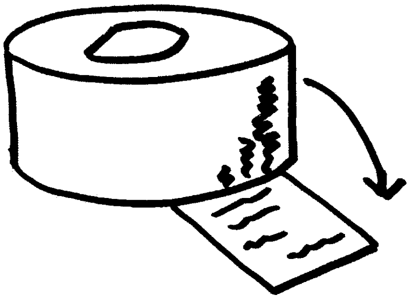

Figure 6-2: A fold-down display on the side of a Sensetable puck. When the user opens the display, the projector above could use it to display more information.

Another area to explore is placing various types of controls on the pucks themselves. One example is the use of a fold down display surface attached to the side of a puck, shown in figure 6-2. If the puck can sense when the display surface is folded open, the position and orientation of the puck on the sensing surface can be used to project extra information about the puck onto the surface. This technique might be used in the system dynamics application to display graphs of various simulation parameters as a function of time. A user will be able to open the display of a puck bound to a node in the simulation to see a plot of that node's behavior over time.

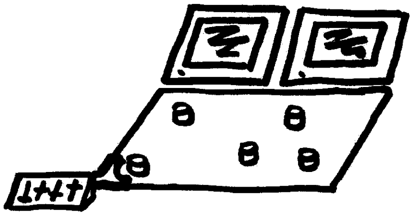

Figure 6-3: Sensetable, with the addition of a bank of motorized potentiometers. These could be used to display and manipulate more detailed information about a puck on the table.

In addition to controls on the pucks themselves, I am interested in using arrays of motorized potentiometers as a way of augmenting the current Sensetable interface. First, these sliders could be used as a way to represent and interact with multiple parameters that correspond to a single puck on the table. For example, if a puck in the system dynamics simulation application represented a higher level abstraction, such as a warehouse in a manufacturing supply chain, there might be several parameters which corresponded to that warehouse. By bringing the puck over to slot near a bank of sliders on the sensing surface, one could use them to set the various parameters, as shown in figure 6-3. If these sliders were motorized, they could be used to address some of the problems described in this thesis relating to consistency between physical and digital representations. The computer could use the motors to physically change parameters or keep them consistent with each other. One could implement haptic feedback using them. Carrying the idea of haptic feedback further, the sliders could facilitate remote collaboration. Users in separate places could manipulate sliders which affected different parameters. If the users tried to move a parameter in a way which conflicted with the intentions of other users, all users involved could use the haptic feedback to help understand what the other users were trying to do.

The Sensetable platform may also prove to be a great platform for conducting more experiments on how people interact with tangible user interfaces. The system's tracking functionality might help one perform more sophisticated analysis of the way users manipulate and arrange objects. As well, the use of projection directly on top of the pucks might cause users to interact with these objects differently than they did in the experiment described in chapter two, for example.

Finally, I am excited about continuing the musical explorations with Sensetable to the level where Sensetable can be used in a performance setting. In addition to making new types of musical expression possible, the application of Sensetable to musical applications can provide a way of evaluating the performance and utility of the project using a demanding, real-time task. One possible performance scenario involves three performers standing around a triangular Sensetable each using the table to modify and create pieces of sound, as in figure 6-4. The users could then pass these physically embodied pieces of sound off to each other across the table to collaboratively create improvisational music in real-time.

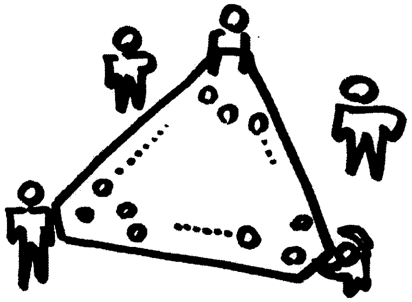

Figure 6-4: Performers collaborating to produce music using Sensetable.

Bibliography

1. Barnett, V. and Lewis, T. *Outliers in Statistical Data.* John Wiley and Sons, Boston MA, 1978

2. Brave, S., Ishii, H. and Dahley, A. *Tangible Interfaces for Remote Collaboration and Communication.* Proceedings of CSCW '98 Conference on Computer Supported Cooperative Work (Seattle, Washington USA, 1998), ACM Press, 169-178.

3. Brewer, B. *The integration of spatial vision and action.* In N. Eilan, R. McCarthy and B. Brewer (Eds.). Spatial Representation. Blackwell, Oxford UK, 1993 (296-316).

4. Buxton, W. and Myers, B. A. (1986). *A Study in Two-Handed Input.* Proceedings of ACM CHI'86 Conference on Human Factors in Computing Systems, 321-326, New York: ACM.

5. Buxton, W. A. *Three-State Model of Graphical Input.* Proceedings of Human-Computer Interaction -INTERACT '90 (1990), 449-456.

6. Campbell, J. *The role of physical objects in spatial thinking.* In N. Eilan, R. McCarthy and B. Brewer (Eds.). Spatial Representation. Blackwell, Oxford UK, 1993 (65-95).

7. Dahley, A., Wisneski, C., and Ishii, H., *Water Lamp and Pinwheels: Ambient Projection of Digital Information into Architectural Space* (short paper), in Summary of Conference on Human Factors in Computing Systems CHI '98, (Los Angeles, April 1998), ACM Press, pp. 299-300.

8. Dumas, S. T., Jones, W. P. *A Comparison of Symbolic and Spatial Filing.* Proceedings of CHI '98 Conference on Human Factors in Computing Systems (San Francisco, CA USA, 1985), ACM Press, 127-130.

9. Fitzmaurice, G., *Graspable User Interfaces*, Ph. D. Thesis., University of Toronto, 1996.

10. Fitzmaurice, G. & Buxton, W. *An empirical evaluation of graspable user interfaces: Towards specialized space-multiplexed input.* Proceedings of CHI '97 Conference on Human Factors in Computing Systems (Atlanta GA, March 1997), ACM Press, 43-50.

11. Fitzmaurice, G., Ishii, H. and Buxton, W. *Bricks: Laying the Foundations for Graspable User Interfaces.* Proceedings of CHI '95 Conference on Human Factors in Computing Systems (Denver, Colorado USA, 1995), ACM Press, 442-449.

12. Frei, P., Su, V., Mikhak B. and Ishii, H. *Curlybot: Designing a New Class of Computational Toys.* in Proceedings of Conference on Human Factors in Computing Systems CHI '00, ACM Press, pp. 129-136, 2000.

13. Forrester, Jay Wright, *Industrial Dynamics.* MIT Press. Cambridge, Mass, 1961.

14. Furnas, G., *Generalized Fisheye Views.* in Proceedings of Conference on Human Factors in Computing Systems CHI '86, ACM Press, pp. 16-23, 1986.

15. Hinckley, K., Pausch, R., Goble, J. C. and Kassell, N. F. (1994). *Passive Real-World Interface Props for Neurosurgical Visualization.* Proceedings of CHI'94 Conference on Human Factors in Computing Systems, 452-458, New York: ACM.

16. Horowitz, P., Hill, W., *The Art of Electronics.* Cambridge University Press. Cambridge, UK. 1989.

17. Ishii, H., Fletcher, R., Lee, J., Choo, S., Berzowska, J., Wisneski, C., Cano, C., Hernandez, A., and Bulthaup, C., *musicBottles,* in Conference Abstracts and Applications of SIGGRAPH '99 , Emerging Technologies, (Los Angeles, California USA, August 8-13, 1999), ACM Press, pp. 174.

18. Ishii, H. and Ullmer, B. *Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms.* Proceedings of CHI '97 Conference on Human Factors in Computing Systems (Atlanta, Georgia USA, March 1997), ACM Press, 234-241.

19. Jacob, R.J.K., Ishii, H., Pangaro, G.,and Patten, J., *A Tangible Interface for Manipulating and Organizing Abstract Information.* Submitted to Symposium of User Interface Softawre and Technology, UIST 2001.

20. Jacob, R.J.K., and Sibert, L.E., *The Perceptual Structure of Multidimensional Input Device Selection.* Proc. ACM CHI'92 Human Factors in Computing Systems Conference, pp. 211-218, Addison-Wesley/ACM Press, 1992.

21. Kirsh, D. *Complementary strategies: why we use our hands when we think.* Proceedings of 17th Annual Conference of the Cognitive Science Society (Pittsburgh PA, August 1995) .

22. Kirsh, D. *The intelligent use of space.* Journal of Artificial Intelligence 73, 1-2 (1995), 31-68.

23. Kirsh, D. and Maglio, P. *On Distinguishing Epistemic from Pragmatic Action.* Cognitive Science 20, (1994), 513-549.

24. Lansdale, M. W. *Remembering about documents: memory for appearance, format, and location.* Ergonomics, 34, 8 (1991), 1161-1178.

25. MacKenzie, C. L. and Iberall, T. *The Grasping Hand.* North-Holland, Elsevier Science, Amsterdam, 1994.

26. Malone, T. W. *How do People Organize their Desks? Implications for the Design of Office Information Systems.* ACM Transactions on Office Information Systems 1, 1 (January 1983), 99-112.

27. Mandler, J. M., Seefmiller, D., Day, J. *On the coding of spatial information.* Memory & Cognition 5, 1 (1997), 10-16.

28. Naveh-Benjamin, M. *Coding of Spatial Location Information: An Automatic Process?* Journal of Experimental Psychology: Learning, Memory, and Cognition 13, 4 (1987), 595-605.

29. Norman, D. A. *Things That Make Us Smart.* Addison-Wesley Publishing Company, Reading MA, 1993.

30. Paradiso, J., et al., *Sensor Systems for Interactive Surfaces.* IBM Systems Journal, 39(3-4), 892-914,2000.

31. Perlin, K., and Fox, D., *Pad: An Alternative Approach to the Computer Interface.* in Proceedings of ACM SIGGRAPH '93, ACM Press, pp. 57-64, 1993.

32. Polhemus. A Rockwell Collins Company.
http://www.polhemus.com

33. Rekimoto, J., and Masanori, S., *Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments.* in Proceedings of Conference on Human Factors in Computing Systems CHI '99, ACM Press, pp. 378-385, 1999.

34. Robertson, G., Czerwinski, M., Larson, K., Robbins, D. C., Thiel, D., van Dantzich, M. *Data Mountain: Using Spatial Memory for Document Management.* Proceedings of UIST '98 Symposium on User Interface Software and Technology (San Francisco, CA USA, November 1998), ACM Press, 153-162.

35. Suzuki, H. and Kato, H., *AlgoBlock: A Tangible Programming Language, a Tool for Collaborative Learning.* in Proceedings of the Fourth European Logo Conference, Athens, Greece, pp. 297-303, 1993.

36. Ullmer, B. and Ishii, H. *The metaDESK: Models and Prototypes for Tangible User Interfaces.* Proceedings of UIST '97 ACM Symposium on User Interface Software and Technology (Banff, Alberta, Canada, October 1997), ACM Press, 223-232.

37. Ullmer, B., Ishii, H. and Glas, D. *mediaBlocks: Physical Containers, Transports, and Controls for Online Media.* Proceedings of SIGGRAPH '98 Special Interest Group on Computer Graphics (Orlando, Florida USA, July 1998), ACM Press, 379-386.

38. Underkoffler, J., Ullmer, B. and Ishii, H. *Emancipated Pixels: Real-World Graphics in the Luminous Room* . Proceedings of SIGGRAPH '99 Special Interest Group on Computer Graphics (Los Angeles, CA, USA, 1999), ACM Press, 385-392.

39. Underkoffler, J., and Ishii, H., *Urp: A Luminous-Tangible Workbench for Urban Planning and Design.* in Proceedings of Conference on Human Factors in Computing Systems CHI '99, ACM Press, pp. 386-393, 1999.

40. Mathews, M., Verplank, B., *Scanned Synthesis.*, http://www.csounds.com/scanned/

41. Wacom Technology, http://www.wacom.com

42. Wellner, P., *Interacting with paper on the Digital Desk.* Communications of the ACM, 36(7), 86-96, 1993.

43. Wellner, P., *The DigitalDesk calculator: Tangible manipulation on a desk top display.* in Proceedings of ACM Symposium on User Interface Software and Technology, UIST '91, ACM Press, pp. 27-34, 1991.

44. Yates, F. A. *The Art of Memory.* University of Chicago Press, Chicago IL, 1996.

45. Zhang, J. *The Nature of External Representations in Problem Solving.* Cognitive Science 21, 2 (1997), 179-217.

46. Zhang, J. and Norman, D. A. *Representations in Distributed Cognitive Tasks.* Cognitive Science 18, 1 (1994), 87-122. 10

25[illegible]