# Tangible Interfaces for Manipulating Aggregates of Digital Information

**Brygg Anders Ullmer**

Bachelor of Science, University of Illinois, Urbana-Champaign, January 1995
Master of Science, Massachusetts Institute of Technology, June 1997

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in Partial Fulfillment of the Requirements for the Degree of

**Doctor of Philosophy**
at the
**Massachusetts Institute of Technology**

September 2002

**Author**
Brygg Anders Ullmer
Program in Media Arts and Sciences
August 9, 2002

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

**Accepted by**
Andrew B. Lippman
Chair, Departmental Committee for Graduate Students
Program in Media Arts and Sciences

MITLibraries
Document Services

Room 14-0551
77 Massachusetts Avenue
Cambridge, MA 02139
Ph: 617.253.5668 Fax: 617.253.1690
Email: docs@mit.edu
http://libraries.mit.edu/docs

# DISCLAIMER OF QUALITY

*** The thesis scanned contains grayscale images only. This is the most complete copy available.**

*** Pages 157-158 do not exist. A pagination error by the author.**

# Tangible Interfaces for Manipulating Aggregates of Digital Information

Brygg Anders Ullmer

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning, on August 9, 2002
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Media Arts and Sciences at the
Massachusetts Institute of Technology

## Abstract

This thesis develops new approaches for people to physically represent and interact with aggregates of digital information. These support the concept of Tangible User Interfaces (TUIs), a genre of human-computer interaction that uses spatially reconfigurable physical objects as representations and controls for digital information. The thesis supports the manipulation of information aggregates through systems of physical tokens and constraints. In these interfaces, physical tokens act as containers and parameters for referencing digital information elements and aggregates. Physical constraints are then used to map structured compositions of tokens onto a variety of computational interpretations.

This approach is supported through the design and implementation of several systems. The mediaBlocks system enables people to use physical blocks to "copy and paste" digital media between specialized devices and general-purpose computers, and to physically compose and edit this content (e.g., to build multimedia presentations). This system also contributes new tangible interface techniques for binding, aggregating, and disaggregating sequences of digital information into physical objects.

Tangible query interfaces allow people to physically express and manipulate database queries. This system demonstrates ways in which tangible interfaces can manipulate larger aggregates of information. One of these query approaches has been evaluated in a user study, which has compared favorably with a best-practice graphical interface alternative. These projects are used to support the claim that physically constrained tokens can provide an effective approach for interacting with aggregates of digital information.

[illegible]

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

# Thesis Committee

**Thesis Supervisor**
Hiroshi Ishii
Associate Professor of Media Arts and Sciences
Massachusetts Institute of Technology

**Reader**
John Maeda
Associate Professor of Media Arts and Sciences
Sony Corporation Career Development Professor of Media Arts and Sciences
Massachusetts Institute of Technology

**Reader**
Robert J. K. Jacob
Associate Professor of Computer Science
Tufts University

# Acknowledgements

I am deeply appreciative for the guidance, inspiration, and support from many people whom have made this research and seven-year journey possible.

Special thanks to:

Hiroshi Ishii, for his enormous energy and vision, for innumerable hours of discussion and guidance that gave shape to the thesis, and for the opportunity to participate in the creation of the Tangible Media Group. After long apprenticeship to a master maker of eyeglasses, I am finally learning to see many shapes and spaces that before were invisible to me.

John Maeda, for his inspiring work and vision of the designer/engineer, his belief in me, and for helping me begin to see and seek the sensuality of physical objects.

Rob Jacob, for his endless and invaluable support, guidance, patience, and belief in my work.

Durrell Bishop, for the inspiration of his work and ideas, which opened my eyes to the possibilities underlying this thesis.

Neil Gershenfeld, for offering conceptual and material tools for the making of many things, and for his visions of personal fabrication and beyond.

Terry Winograd and Mitchel Resnick, for support of my master's thesis, and much beyond.

Joe Paradiso, for helping me through my general exams, and support throughout the duration.

Bill Verplank, for support and teachings at both Interval and the Media Lab, and for first pointing me in the direction of physical constraints.

Debby Hindus, a true mentor, for helping guide me to and through the Media Lab.

Whitman Richards, for ideas of the non-accidental, and support of my interests in board games.

Joe Jacobson, for inspiration, teachings, and support in experiments with reusable paper.

John DiFrancesco, for his endless patience, support, good humor, and order out of chaos.

Linda Peterson and Lisa Lieberson, for much support and guidance on details large and small.

IBM, Mitsubishi, and AT&T, for generous fellowship support of my studies.
Special thanks to David Boor, Joe Marks and Will Hill.

To the undergraduate students who provided invaluable support with the thesis projects: Dylan Glas, Paul Grayson, Zach Malchano, Anna Lee, John Alex, Eli Kim, Jenn Yoon, Nick Fahey, Chris Fuchs, Chris Lopes, and Minpont Chien. Special thanks to Dylan for inspiring me to "trade places," and to Paul for long aquaintance, countless conversations, and help in sharpening the conclusion's key question.

To the many fellow graduate students who've taught me so much:

In TMG, to Paul Yarin, John Underkoffler, James Patten, Ben Piper, Ali Mazalek, Jay Lee, Matt Gorbet, Dan Chak, Gian Pangaro, Dan Maynes-Aminzade, Cati Vaucelle, Mike Ananny, Angela Chang, Craig Wisneski, Phil Frei, Joey Bersowska, Seungho Choo, Andy Dahley, Scott Brave, Sandia Ren, Victor Su, and Kazue Kobayashi, for their camaraderie, energy, collaboration, and patience with me during times of large writings.

In ACG, special thanks to Ben Fry, Casey Reas, Golan Levin, Nikita Pashenkov, Tom White, Bill Keays, Reed Kram, Matt Grenby, Justin Manor, and Chloe Chao.

In the VLW, special thanks to Dave Small, Suguru Ishizaki, Lisa Strausfeld, and Earl Rennison.

In the Media Lab at large, Kelly Heaton, Steve Gray, Rob Poor, Tim McNerney, Rich Fletcher, Matt Reynolds, Rehmi Post, Ravi Pappu, Wendy Plesniak, Lisa Stifelman, Thad Starner, Maggie Orth, Marina Umaschi-Bers, Tara Rosenberger-Shankar, Jen Glos, Ivan Chardin, Nelson Minar, Karrie Karahalios, Fernanda Viegas, Hyun-Yeul Lee, and many others.

Kwin Kramer, Rick Borovoy, and Matt Klicka, for support with inductive communications.

Axel Kilian, who sees so clearly and teaches so much, for invaluable support with Strata and much else.

Fellow students at a distance: Lars Erik Holmquist, Johan Redström, Staffan Björk, Jennica Falk, Takeo Igarashi, Eva Hornecker, Steven Schkolne, Michael Pruett, and David McGee.

Sony CSL, especially Jun Rekimoto, Haruo Oba, Shigeru Tajima, Yuji Ayatsuka, Tota Hasegawa, Nobuyuki Matsushita, and Kayoko Wada, for an inspiring summer of research. Also to NSF, JISTEC, and the Summer Institute in Japan for support.

Interval Research, my "first graduate school," and to the amazing people who opened my mind to many new ways of thinking and doing. Special thanks to Paul Freiberger, Terry Winograd, David Liddle, Debby Hindus, Bill Verplank, Michael Naimark, Lee Felsenstein, Jim Boyden, Dick Shoup, Meg Withgott, Jonathan Cohen, Philippe Piernot, Margaret Minsky, Ellen Levy, Sean White, Karon MacLean, Marc Davis, Neal Bhadkamkar, and many others.

The many good people of ICC who provided wonderful support with Strata/ICC.
Special thanks to Shoji Itoh, Tetsushi Nonaka, Haruo Noma, Chie Arai, Seiji Sawaguchi, Koichi Kido, Hiroshi Kanechiku, Shiro Yamamoto, and many others.

Joe Branc, Joel Stanfield, and Steve Whittaker, for highly valued support and encouragement.

Ron MacNeil, for opening the doors to the Media Lab and introducing me to design.

John W. Durham, for detailed comments on the abacus and backgammon.

Martin Nilsson, for early feedback on the query interfaces from a commercial perspective.

Hannes Vilhjálmsson, for six years of friendship, support, and long nights of rich discussion.

My parents, brother, sister-in-law, and yet-to-be-met nephew, for their unwavering love, support and guidance in pulling everything into perspective.

...and to Miriam Konkel, my inspiration of recent years and my dream for the future...

# Contents

# chapter 1

# introduction

# 1 Introduction

For most of the history of computing, people have relied on screen-based text and graphics as the primary means for representing digital information. Whether the screen is desk-mounted, head-mounted, hand-held, or embedded in the physical environment, the prevailing combination of screens and general-purpose input devices has cultivated a predominantly visual paradigm of human-computer interaction.

While this paradigm has proved very successful, the overwhelming dominance of graphical user interfaces (GUI) in general and the "WIMP" (windows-icon-menu-pointer) interaction style in particular have sparked interest in alternative approaches from the research community for more than a decade (e.g., [Green and Jacob 1991]). Perhaps the central critique of WIMP-style interfaces concerns their major asymmetry between "input" and "output" interaction modalities. While often employing millions of pixels of graphical output, WIMP interfaces generally rely upon a single locus of pointer-driven input, in a style largely devoid of physical and kinesthetic affordances, or other handles for engaging with a world of multiple users, each with two hands and a lifetime of physical skills.

Partly in response to such issues, the last decade has seen a wave of new research into ways to link the physical and digital worlds. These efforts have led to the growth of several major research themes, including augmented reality, mixed reality, ubiquitous computing, and wearable computing. While these approaches increase the integration of computation with the physical environment, most continue to heavily rely on traditional GUI and WIMP techniques, consequently subjecting themselves to many of the corresponding limitations. Moreover, these efforts have made increased use of visual display channels while generally maintaining pointer- and keyboard-based interactions, thus further increasing the asymmetry of input/output modalities.

Simultaneously, a new stream of interface research has begun to explore the relationship between physical representation and digital information, highlighting kinds of interaction that are not readily described by existing frameworks. Fitzmaurice, Ishii, and Buxton took an important step towards describing a new conceptual framework with their discussion of "graspable user interfaces" [1995]. Building upon this foundation, Ishii and I extended these ideas and proposed the term "tangible user interfaces" (TUIs) in [Ishii and Ullmer 1997].

Among other historical inspirations, we suggested the abacus as a compelling prototypical example. In particular, it is key to note that when viewed from the perspective of human-computer interaction (HCI), the abacus is not an "input device." The abacus makes no distinction between "input" and "output." Instead, the abacus beads, rods, and frame serve as manipulable physical representations of numerical values and operations. Simultaneously, these component artifacts also serve as physical controls for directly manipulating their underlying associations.

This seamless integration of *representation* and *control* differs markedly from the mainstream graphical user interface (GUI) approaches of modern HCI. Graphical interfaces make a fundamental distinction between "input devices," such as the keyboard and mouse, as *controls;* and graphical "output devices" like monitors and head-mounted displays, for the synthesis of visual *representations*. Tangible interfaces, in the tradition of the abacus, explore the conceptual space opened by the elimination of this distinction.

## 1.1 Basic concept and approach

Tangible user interfaces are broadly concerned with *giving physical form to digital information.* At the highest level, there are two basic facets of this approach. First, physical objects are used as representations of digital information and computational operations. Secondly, physical manipulations of these objects are used to interactively engage with computational systems.

This description can be inverted into several key questions. First, what kinds of information and operations might one wish to represent and manipulate with physical objects? And secondly, what kinds of physical systems might be used to mediate these interactions?

### *1.1.1 Earlier approaches*

Likely the most popular application of tangible interfaces has been to use physical objects to model various kinds of physical systems. For example, tangible interfaces have been used to describe the layout of assembly lines [Schäfer et al. 1997, Fjeld et al. 1998], optical systems [Underkoffler and Ishii 1998], buildings [Underkoffler and Ishii 1999], furniture [Fjeld et al. 1998], and so on. One paradigm for these systems is based upon "interactive surfaces," where users manipulate physical objects upon an augmented planar surface. The presence, identity, and configuration of these objects is then electronically tracked, computationally interpreted, and graphically mediated.

Another tangible approach for modeling physical systems draws inspiration from building blocks and LEGO™. Such "constructive assemblies" of modular, interconnecting elements have been used for modeling buildings [Aish 1979, 1984; Frazer 1982, 1995; Anderson et al. 2000], fluid flow [Anagnostou et al. 1989], and other geometrical forms [Anderson et al. 2000]. While instances of "interactive surfaces" and "constructive assemblies" may take on a wide variety of embodying forms, illustrative examples are loosely depicted in Figure 1.1.

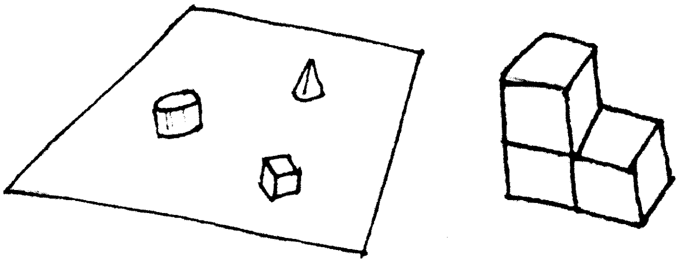


**Figure 1.1a,b: Loose illustrations of "interactive surface" and "constructive assembly" approaches**

The above examples provide several possible answers to §1.1's leading questions. While interactive surfaces and constructive assemblies have broader applications, prior to this thesis they were most often used to represent and manipulate inherently geometrical systems, associating physical objects with corresponding digital geometries and properties. An important benefit is that these systems can often take advantage of existing physical representations and work practices, while extending these with the benefits of computational augmentation. However, a corresponding limitation is that many kinds of digital information have no inherent physical or geometrical representations.

### *1.1.2 Thesis approach*

This thesis develops a new paradigm for physically interacting with digital information, and applies this to interaction with a kind of information that has not previously been accessible from tangible interfaces. Where previous tangible interfaces have made one-to-one mappings between physical objects and digital elements, this thesis uses physical tokens to describe and represent *aggregates* of digital information. This significantly increases the scalability of tangible interfaces, allowing a small number of physical elements to manipulate moderate to large collections of digital information. This design choice also potentially allows simple physical actions to apply powerful computational operations over large collections of information.

Digital information aggregates have no inherent physical representation, nor any intrinsic physical language for their manipulation. To support these interactions, this thesis develops an approach that combines two kinds of physical/digital artifacts: *tokens* and *constraints.*
In the context of this thesis, *tokens* are discrete, spatially reconfigurable physical artifacts that each describe or represent an element or aggregate of digital information. *Constraints* are structures that physically channel how tokens can be manipulated, often limiting their movement to a single physical dimension. The physical manipulation of tokens within these constraints (e.g., token entrance, exit, translation, and rotation) is then mapped to a variety of computational interpretations. This approach is loosely illustrated in Figure 1.2.

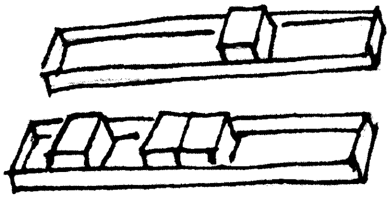


**Figure 1.2: Loose illustration of token+constraint approach**

These systems of tokens and constraints serve as a kind of physical/digital "language" that can be mutually interpreted by both people and computers. Taken separately, tokens and constraints are not individually "actionable." Combined together, tokens and constraints represent fully formed, manipulable computational expressions.

In this thesis, tokens represent information aggregates in two ways. In the first approach, tokens serve as "containers" for lists of online information elements. In the second approach, tokens represent parameters that describe relationships across large collections of information.

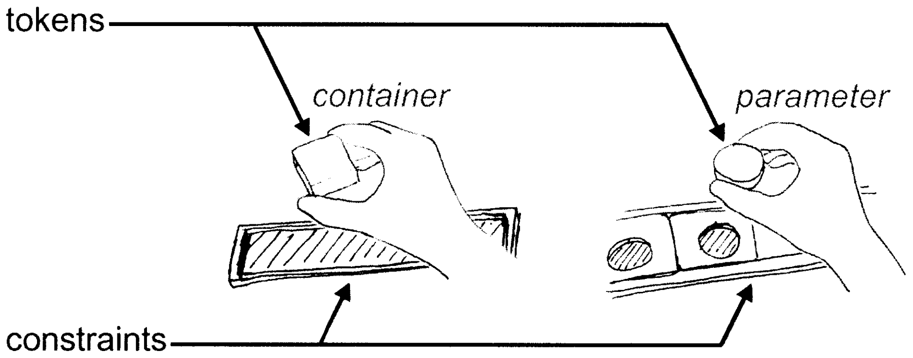


**Figure 1.3: Representational approaches of tokens and constraints**

The combination of tokens and constraints has two phases of interaction: *associate* and *manipulate*. These are illustrated in Figure 1.4. In the first phase, one or more tokens are *associated* with a specific constraint structure. This is accomplished by placing the token within the physical confines of the constraint, and can be reversed by removing the token. In addition to establishing a physical relationship between the token and constraint, this action also establishes a computational relationship between the token and constraint's corresponding digital bindings and interpretations.

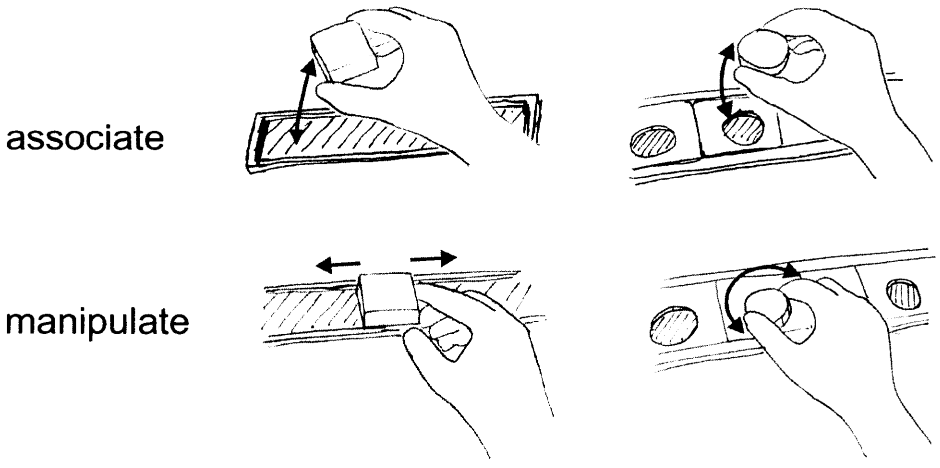


**Figure 1.4: Two phases of interaction with token+constraint interfaces**

In the second phase, tokens are *manipulated* within the confines of this constraint. When placed within a constraint, each token is generally constrained to move with a single degree of freedom (either translational or rotational). If a single token is present within a constraint, then the physical configuration of the token is interpreted with respect to the constraint.

If multiple tokens are present within the constraint, their configurations may be interpreted either with respect to each other, or with respect to the parent constraint. In addition, token + constraint interfaces typically contain multiple constraints, and the token/constraint roles can be "nested." This is loosely illustrated in Figure 1.5. These possibilities allow for a potentially wide range of combinations.

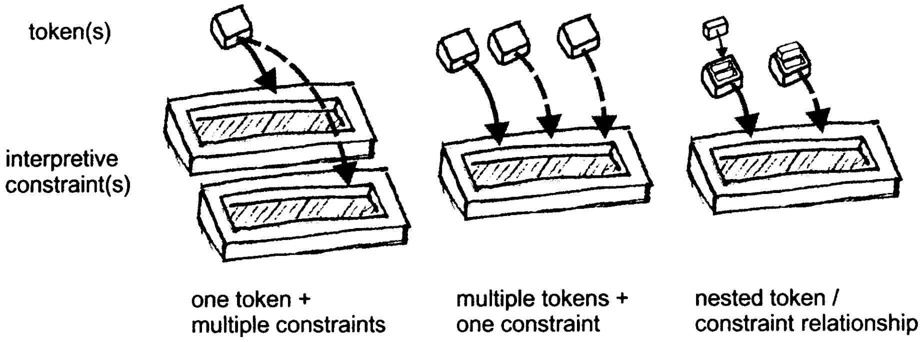


**Figure 1.5: More complex combinations of tokens and constraints:**
*one token + multiple separate constraints; multiple tokens + a single constraint; nested token/constraint relationships*

Another important aspect of the associate and manipulate phases of interaction is that they often correspond with the expression of *discrete* and *continuous* interactions and relationships. The associate phase is generally discrete and binary in state; tokens are generally interpreted as either present or absent from a given constraint. In contrast, the manipulate phase often involves the spatially continuous reconfiguration of tokens within the confines of a parent constraint. This combination supports the benefits of both discrete expressions (e.g., commands and discrete relationships), as well as continuous ones (e.g., continuous scalar values and relative+absolute indices within aggregates of digital information).

In some respects, token+constraint interfaces realize a simple kind of physical/digital "language." However, this thesis is not oriented toward the design of "tangible programming languages" (though several outside examples of this approach will be discussed). Instead of the deliberate, cumulative expressions of programming languages, the token+constraint interfaces of this thesis are used to embody interactive workspaces where physical actions bring an immediate interpretation and response by the system. In this respect, the thesis approach closely follows the principles of "direct manipulation" articulated in [Shneiderman 1984].

A key property of token+constraint interfaces is that they use physical properties to help express digital syntax, both of individual interface primitives and of systems as a whole. By mechanically limiting which tokens can be accommodated and what configurations these tokens can assume, the constraints can physically express and partially enforce the syntax of their associated digital operations. Also, the structure and configuration of multiple constraints can help encode and partition the cumulative syntax of multifunction systems.

While not eliminating the possibility of meaningless expressions, token+constraint systems physically express to users something about the kinds of interactions the interface can (and cannot) support. This use of physical properties to encode digital syntax provides both conceptual and technological aids for the realization of tangible interfaces.

## 1.2 Distinguishing properties of tangible interfaces

While related systems extend back decades in time, the concept of tangible user interfaces was not articulated prior to the work of this thesis. For this reason, while the thesis focuses upon a particular subset of the TUI design space, it is important to describe the properties that broadly distinguish tangible interfaces from other approaches.

From the perspective of this thesis, tangible user interfaces will be considered to be systems that use *spatially reconfigurable* physical artifacts as *representations* and *controls* for digital information. The physical/digital artifacts at the center of tangible interfaces – the *tangibles* – can be seen as having four distinguishing physical properties:

1) physically embodied
2) physically representational
3) physically manipulable
4) spatially reconfigurable

### *1.2.1 Physically embodied*

The broadest property and criteria of tangible interfaces is that digital information or functionality is somehow embodied in physical form. At some level, every computational device involves some form of physical artifact. The distinction of concern in this thesis surrounds the modalities by which people perceive and interact with digital information.

In most conventional interfaces, human-computer interaction is mediated in visual form through dynamic text and graphics, or (less frequently) through audio. However, the audio and visual modalities are "intangible;" they are not physically accessible to direct haptic manipulation. (It is worth noting that the word "tangible" derives from the Latin "tangibilis" and "tangere," meaning "to touch.") While force-feedback devices (e.g., the Phantom™) do provide for direct haptic engagement, they too most often mediate interaction with simulated "virtual" objects that have no real physical embodiment.

In contrast, the game of chess provides an interesting counterexample. At the beginning of each game, a certain kind of object is positioned at each of the chessboard's four corners. These objects are not "interfaces to the rook behavior." Rather, they are *physical embodiments* of the rook – or more succinctly, they *are* rooks.

A broad variety of computational interfaces have been explored as examples of the "tangible interfaces" concept. Some of these are marked by all five properties of this section; others have few of them. While it is unproductive to insist on an overly selective interpretation, physical embodiment is the broadest property and criteria that can be expected of any kind of tangible interface.

### 1.2.2 *Physically representational*

Physical representation is one of the most important properties of tangible interfaces, and a focus point in this thesis. The term "representation" can have many meanings; these will be considered more carefully in §1.4. The principle interpretation within this thesis is that artifacts serve as physical manifestations of particular kinds of digital associations. Examples include the use of objects as embodiments of specific data, as containers for data aggregates, or as specifiers of digital parameters.

The specific physical form of these physical artifacts can vary widely. On the one hand, they can be literally or iconically representational, as in the physical miniatures of buildings from the project leading into this thesis (the metaDESK). Alternatively, TUI artifacts can be symbolically representational, bearing no material resemblance to the digital associations for which they stand. Many of the systems in this thesis represent digital elements for which there are no physical world counterparts, making symbolic representation the only alternative.

The role of physical representation has several additional implications for tangible interfaces. First, when physical artifacts are used to serve different kinds of roles (e.g., as data container vs. function specifier), it is important to develop different physical forms to reflect these functional differences. Secondly, physical representation is a factor not only for individual physical artifacts, but also for the physical space in which they are used. This relationship between physical *tokens* and *reference frames* relates closely to the token+constraint approach, and will be an important topic in the thesis (e.g., see §2.6.2).

### 1.2.3 *Physically manipulable*

In §1.0, tangible interfaces were described as employing physical artifacts both as *representations* and *controls* for computational media. Within tangible interfaces, this capacity for control derives from the direct physical manipulation of interface artifacts. This physical manipulation is in turn sensed and monitored by the underlying system, and mapped to corresponding digital interpretations.

An important aspect of physical manipulability is that TUI artifacts are generally *graspable*. This property has been prominently highlighted by the work of Fitzmaurice [1996]. This means that objects can be taken within the hand, and physically manipulated with the hand and fingers. This has implications for physical scale and accessibility, and relates to ideas about physical affordances that are discussed within §2.2.3.

### 1.2.4 *Spatially reconfigurable*

For hundreds or even thousands of years, buttons, knobs, levers, and other physical mechanisms have been used to control a wide variety of mechanical, electrical, and computational devices. While these physical controls generally employ some form of translational or rotational movement, they are usually fixtured to a particular spatial location of the host device. In the few cases where such controls are not mechanically anchored to their parent device – e.g., the remote control of a television or stereo – the spatial manipulation of the controls (e.g., the orientation of a remote control) generally is not sensed or interpreted in any way. From the perspective of §1.1.2 and Figure 1.4, these interfaces contain a "manipulate" phase, but no "associate" phase.

In contrast, the spatial reconfiguration of physical elements – their physical placement and removal, translation and rotation – is the central mode of interaction with tangible interfaces. While these compositional elements often will be mechanically constrained, their spatially reconfigurable state will take on special significance in the thesis.

As a corollary, tangible interfaces are generally composed of *discrete* physical artifacts. Much of recent interaction design has worked toward the physical aggregation of diverse functional elements into complex integrated devices. In contrast, tangible interfaces work to discretize interfaces into multiple distinct physical elements, each with their own digital bindings. The presence of multiple discrete objects also suggests that tangible interfaces are built with coordinated systems of interacting physical elements. The design and interpretation of these systems will form one of the major topics of this thesis.

At the same time, this aspect is not meant to contradict the "tangibility" of important recent systems such as the Illuminating Clay work of Piper, Ratti, and Ishii [2002]. As indicated by the sentences above, the "discrete" term is intended to distinguish from the integrated input devices, mechanisms, and approaches of traditional human-computer interfaces, and not the continuous nature of the Clay work's composing elements. This distinction is considered in more detail within §8.2.4.

## 1.3 Tokens and constraints

Taken together, the above four properties describe systems that use *spatially reconfigurable* physical artifacts as *representations* and *controls* for digital information. These properties are taken as describing the core space of "tangible interfaces." It is worth noting that a number of interesting design spaces are exposed by relaxing or reversing certain properties. For example, relaxing the "physically manipulable" criteria exposes the space of "ambient displays" [Wisneski et al. 1998]. Similarly, inverting the "physically discrete" property highlights efforts such as the recent "Illuminating Clay" system, which develops interesting interpretations of a continuous sheet of malleable clay [Piper et al. 2002]. These broader alternative spaces will be considered in the discussion chapter.

This thesis examines the space of tangible interfaces as defined by the four properties of §1.2, developing new representational approaches for giving physical form not only to digital information itself, but also to aspects of the syntax for manipulating this information. Syntax is defined by the OED as "the order and arrangement of the words or symbols forming a logical sentence" [OED 1989]. It is the grammar of ways in which objects can be combined together to form expressions that can be meaningfully interpreted by the interface.

In graphical interfaces, software may visually express the ways with which graphical objects can be combined, and can directly enforce consistency between user actions and allowable configurations. However, the physics of the real world differs from that of graphical spaces. Software and graphical output alone cannot enforce consistency in configurations of discrete physical objects.

This thesis focuses on interactions between tokens and constraints that physically channel how tokens can be manipulated, often limiting movement to a single physical dimension. These structures will sometimes be referred to as "interpretive constraints" to reflect their role in mapping compositions of physical tokens to various digital interpretations. This term has also been used in similar ways within linguistic, legal, and theological contexts, where it describes a phrase or text that is considered within the "interpretive constraint" of some earlier precedent.

As discussed in §1.1.2 and illustrated in Figure 1.3 and Figure 1.4, the token+constraint approach centers on a hierarchical relationship between two kinds of physical elements: tokens and constraints. Tokens can be placed within or removed from compatible constraints. Compatibility is generally expressed through the physical shape of the tokens and constraints, with incompatible elements rendered incapable of mechanically engaging with each other.

The manipulation of tokens within constraints can take several forms. In the simplest cases, tokens may be inserted or removed, but no further mechanical engagement with the constraint is possible. In the language of §1.1.2, this corresponds with supporting an "associate" interaction, but no "manipulate" phase. In other cases, constraints allow child tokens to be reconfigured within a limited range of movements, typically limited to a single translational or rotational degree of freedom. These three basic combinations are loosely illustrated in Figure 1.6.

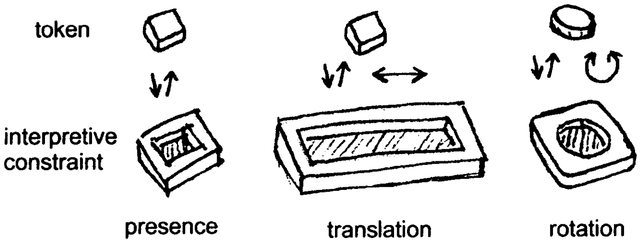


**Figure 1.6a,b,c: Basic combinations of tokens and constraints:** *presence; presence+translation; and presence+rotation*

In addition to these basic combinations of tokens and constraints, several other combinations are possible. First, as mentioned before, tokens can be moved between different constraints to apply different digital operations. Secondly, some constraints can contain multiple physical tokens, whether of a single kind or multiple different kinds. In these cases, the relative and absolute positions of the tokens both with respect to each other and to the constraint can all potentially signal different interpretations.

Moreover, the tokens/constraint relationship can be nested. Here, a given artifact can serve both as a "parent" constraint for one or more "child" tokens, and simultaneously as a "child" within another large frame of reference. A familiar analogy in American culture is found in the game of "Trivial Pursuit," with its "pie" tokens that each have receptacles for six child "wedges." These examples of more complex token/constraint combinations are loosely illustrated in Figure 1.5.

When viewed from the perspective of computer science and "object-oriented" programming, physically constrained tokens illustrate a kind of "multiple inheritance." When placed within a constraint, tokens simultaneously represent both the container for an aggregate of data, and also the control for acting upon this embodied digital content. While this kind of behavior is uncommon in the world of graphical interfaces, it seems to follow straightforwardly from the physical properties of tangible interfaces.

### *1.3.1 An example: rack constraints*

The thesis develops examples of both translational and rotational constraints. One of the thesis' primary examples is "racks" that structure the manipulation of physical tokens within a linear constraint. Several of the particular configurations of racks and tokens that will be developed in the thesis are illustrated in Figure 1.7.

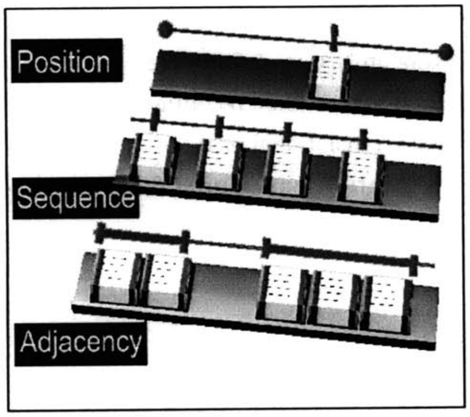


**Figure 1.7: Physically structured syntax of tokens upon racks**

These configurations are not "arbitrary," and can be expressed as the combination of several basic physical properties. Specifically, the configurations of Figure 1.7 (and others) can be described in terms of the relative and absolute positions of tokens with respect to the constraint and to each other. This will be discussed further in §2.6.2.

Interpretive constraints physically express to users something about the kinds of interactions the interface can (and cannot) support. Interpretive constraints also help to support consistency by mechanically restricting the physical relationships that objects can express.

Interpretive constraints do not fully specify the syntax of physical/digital expressions. For example, Figure 1.7 illustrates three different syntactic expressions that can be expressed upon the same rack structure. This suggests the need to distinguish and potentially select between multiple alternate mappings, which will be discussed further in §4.10.3.

Similarly, interpretive constraints do not eliminate the possibility of invalid expressions. Speaking of such issues, Ten Hagen said:

> Syntax describes choice – what you can say. It will allow many [digital expressions] that don't make sense. You need to decide the borderlines where you stop [invalid expressions] by syntax, semantics, or not at all. [1981]

### *1.3.2 Strengths of token+constraint approach*

Interpretive constraints have a number of strengths, which can be approached from several standpoints. From a representational standpoint, interpretive constraints help to express:

- *the set of physical tokens that can take part within a given interpretive constraint:*
  The mechanical structure of interpretive constraints can help express physical/digital compatibilities with certain subsets of tokens, as encoded in physical properties like token size and shape.
- *the set of physical configurations these physical tokens can take on:*
  Objects are mechanically limited to configurations that have well-defined computational interpretations, and encourage alternate reconfigurations within these constraints.
- *the demarcation between interaction regions with different computational interpretations:*
  the well defined boundaries of interpretive constraints aid the combination+integration of multiple constraints. These boundaries also aid the integration of interpretive constraints into self-contained physical devices, as demonstrated by the thesis' mediaBlocks system.

Viewed from a somewhat different perspective, interpretive constraints help to simplify and structure:

- *human perception* – interpretive constraints use physical properties to perceptually encode digital syntax. Among other things, they shift cognitive load to external representations (see §2.2.1), and support perceptual chunking of object aggregates.
- *human manipulation* – interpretive constraints provide an increased sense of kinesthetic feedback from the manipulation of tokens, stemming from passive haptic feedback yielded by the mechanical constraints. Interpretive constraints also support the manipulation of aggregates of multiple physical objects. This is realized both through manipulation of entire constraint structures (e.g., moving a rack full of tokens), or through actions like "sweeping"/"bulldozing" a sequence of tokens constrained by a rack.
- *machine sensing* – interpretive constraints can significantly simplify the sensing of a tangible interface's physical state. These structures can often allow continuous sensing

with two, three, or even six degrees of freedom per object to be replaced by a serious of discrete sensing points. This can simplify implementation, increase scalability, and increase flexibility in the physical forms that tangible interfaces can assume.

- *machine interpretation* – interpretive constraints can also simplify the interpretation of physical configurations, by virtue of limiting them to a smaller space of relatively well-defined states. This is both an implementational aid, and can also help to minimize or eliminate error conditions.

It is also important to add that interpretive constraints can be used either as the primary interaction devices of a TUI, or in combination with other TUI elements and approaches. The two major systems of this thesis are based almost entirely upon interpretive constraints. At the same time, earlier systems have combined simple interpretive constraints with freely manipulable interactive surface approaches (e.g., [Fitzmaurice 1995]), sometimes using interpretive constraints as menu-like structures. This prospect will be revisited in Chapter 7.

### *1.3.3 Digital mappings of token+constraint configurations*

A number of alternatives for mapping token+constraint relationships onto digital interpretations are possible. Several of these are summarized in Figure 1.8.

| Physical relationships | Interaction Event | Digital interpretations |
|---|---|---|
| Presence | Add/Remove | Logical assertion; activation; binding |
| Position | Move | Geometric; Indexing; Scalar |
| Sequence | Order change | Sequencing; Query ordering |
| Proximity | Prox. change | Relationship strength (e.g., fuzzy set) |
| ├Connection | Connect/Discon. | Logical flow; scope of influence |
| └Adjacency | Adjacent/NAdj. | Booleans; Axes; other paired relations |

**Figure 1.8: Grammars for mapping physical relationships to digital interpretations**

These physical relationships are not limited to physically structured approaches. For example, the same relationships can also be expressed within freely manipulable interfaces, which usually possess a superset of the physical degrees of freedom of physically structured approaches. Nonetheless, these mappings are all well supported by the token+constraint approach, and in this case are accompanied by the kinds of benefits discussed in the last section.

The thesis develops a number of these physical relationships and mappings. The "mediaBlocks" system develops presence-, position-, and sequence-based mappings through an array of racks, "slots," "chutes," and "pads." The tangible query interfaces employ presence, position, and adjacency-based mappings, again using rack and pad constraints. Figure 1.8 will be revisited in the discussion of these systems.

## 1.4 Representational approach

As earlier discussed, tangible interfaces are closely concerned with notions about representation. "Representation" is a rather broad term, taking on different meanings within different communities. In artificial intelligence and other areas of computer science, the term often relates to the software programs and data structures serving as the computer's *internal* representation (or model) of information. In this thesis, the meaning of "representation" will center upon *external* representations – the external manifestations of information in fashions directly perceivable by the human senses.

The space of external representations can be divided into two broad classes. First, *tangible representations* are considered in this thesis to be information that is physically embodied in concrete, "tangible" form. Alternately, *intangible representations* are computationally mediated displays that are perceptually observed in the world, but are not physically embodied, and thus "intangible" in form. For instance, the pixels on a screen or audio from a speaker are examples of intangible representations, while physical chess pieces and chessboards are examples of tangible representations.

In some respects, intangible representations approximate audio/visual representations – the transient displays that are products of ongoing computations. As a clarifying heuristic, when the power to a tangible interface is removed, it is the "intangible representations" that disappear, and the embodied, persistent "tangible representations" that remain. Tangible interfaces are products of a careful balance between these two forms of representation.

### 1.4.1 Existing representations

There are many different ways in which physical artifacts can potentially serve as "tangible representations of digital information." In some cases, application domains are marked by inherent representations that can be directly employed within tangible interfaces. For instance, many physical world phenomena, including the domains of classical physics, geography, and architecture, lend themselves to direct mappings between their physical-world subjects and geometric representations. These domains simplify the mapping challenge considerably, and have been among the most thoroughly explored genre of tangible interfaces. Such inherently geometrical domains were explored by the "metaDESK" research leading up this thesis [Ullmer and Ishii 1997], and have since been compellingly illustrated by the holographic and urban planning simulators of John Underkoffler [Underkoffler and Ishii 1999].

Simultaneously, many domains – e.g., economics, information management, computing systems, etc. – do not possess such inherent mappings. In some cases, representational conventions may already exist. As an example, the metrics of stock market performance are not inherently geometrical; representations such as linear plots of selected dependent variables vs. time are human constructions. However, stocks lend themselves well to geometric (spatial) representation, and have developed strong representational conventions that form a shared language for those fluent in the domain.

### 1.4.2 New representations

In some domains, neither inherent representations nor representational conventions exist. For example, many of the core primitives for graphical user interfaces – e.g., menus and hyperlinks – use invented visual representations that are peculiar to the unusual "physics" of dynamic graphical spaces. These representations are generally not "arbitrary;" successful examples employ good visual design principles, such as using spatial juxtaposition and visual structure to emphasize relationships and foster legibility. However, these examples do illustrate the design of new representations particular to the properties of the new medium.

To consider an example, it is interesting to compare "buttons" and "hyperlinks" within graphical interfaces. GUI "buttons" build on the metaphor of physical buttons. One "presses" the button (general through a mouse or some other mediating device), and some digital action is activated. A GUI button often changes its visual appearance when pressed, but the button's visual representation generally "continue to exist" throughout the course of the interaction.

In contrast, hyperlinks are invented representations, exhibiting behaviors that are specific to the peculiar physics of dynamic computational displays. When pressed, hyperlinks instantly transport the user to another graphical "place." Hyperlinks generally vanish during this "teleportation" process. The physical action of invoking buttons and hyperlinks is similar. However, the two devices have quite different conceptual models, and typically are visually represented in different ways.

This thesis develops new kinds of physical representations for applying tangible interfaces to the manipulation of digital information. Both of the major thesis systems give physical form to digital operations that previously have been accessible primarily or strictly through traditional computer interfaces. At the same time, while these newly embodied physical/digital actions do not directly have prior physical work practice, they may draw on the full history of human experience with the physical world.

For example, the "rack" constraint investigated in depth within the thesis leverages a physical "affordance" provided by supporting surfaces that has been used by people for thousands of years. Racks similar to the thesis system have been used as sorting shelves for type slugs, photographic slides, game pieces (e.g., the tile rack of the Scrabble™ game), and so forth. While the digital interpretations applied to these physical structures are new, the vocabulary of physical structures developed in the thesis will be largely familiar.

## 1.5 Overview of supporting systems

### *1.5.1 First supporting system: mediaBlocks*

This thesis presents two major systems that apply tokens + constraint approaches to the manipulation of digital information aggregates. The first of these is *mediaBlocks,* a system for physically capturing, retrieving, and manipulating digital media such as images and video. MediaBlocks are small wooden blocks that are used for the capture, transport, and control of online media. These blocks do not actually store media internally. Instead, mediaBlocks are embedded with digital ID tags that allow them to function as "containers" for online content, or alternately expressed, as a kind of physically embodied URL.

MediaBlocks interface with media input and output devices such as video cameras and projectors, allowing digital media to be rapidly "copied" from media sources and "pasted" into media displays (Figure 1.9b). MediaBlocks are also compatible with traditional GUIs, providing seamless gateways between tangible and graphical interfaces. Finally, mediaBlocks are used as physical "controls" in tangible interfaces for tasks such as sequencing collections of media elements, as demonstrated by the media sequencer device (Figure 1.9a).

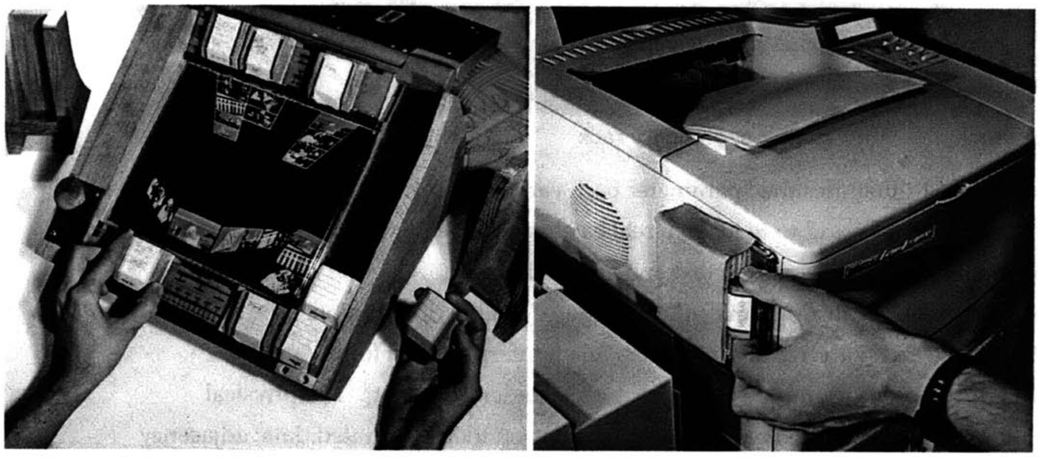


**Figure 1.9a,b: mediaBlocks sequencer, printer slot**

### *1.5.2 Second supporting system: Tangible query interfaces*

The second major thesis system is a series of tangible interfaces for physically expressing parameterized queries to database systems. These systems use several kinds of physical tokens that represent query parameters and data sets. These tokens are used in combination with constraints that map compositions of tokens onto the expression and visualization of database queries. Example of these interfaces are illustrated in Figure 1.10 and Figure 1.11.

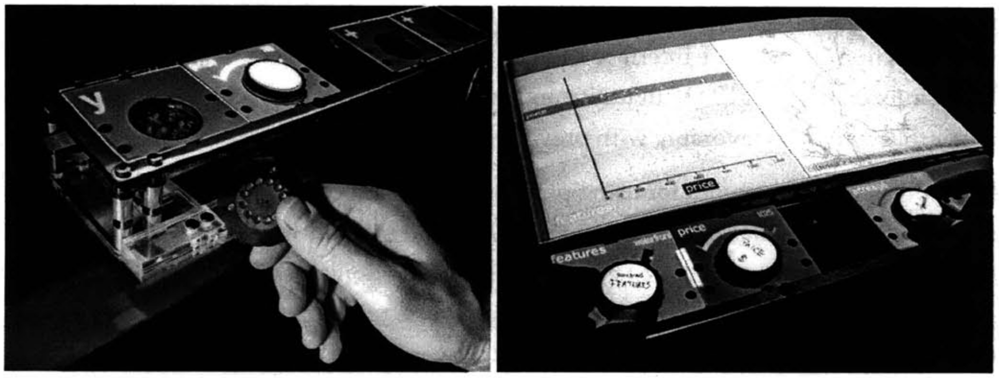


**Figure 1.10a,b: Parameter wheels on query rack, in system overview**

Figure 1.10 illustrates the "parameter wheel" approach for tangible interfaces. Here, round disks called "parameter wheels" are bound to database parameters, which can be placed within round "pads" that in turn are embedded in a "query rack." The placement of these wheels onto the rack selects active query parameters and describes data visualizations, while the rotation of parameter wheels allows physical manipulation of the associated value range.

**Figure 1.11a,b: Parameter bars: standing, on query rack**

Alternately, Figure 1.11 illustrates the "parameter bar" variation of the query interfaces. Here, three "parameter bars" are bound to fields of a real estate database ("acreage," "price," and "square footage"). Each object displays a text and color label identifying the parameter to which it is bound, as well as a histogram of the parameter's value distribution. The range of this distribution is selected with two sliding levers, in an interpretation following the graphical "dynamic queries" technique [Ahlberg and Shneiderman 1994]. The physical adjacencies of these bars map to Boolean query operations on their associated data; adjacency maps to "AND," while non-adjacency maps to "OR." These interpretations are visualized directly upon the query rack, with query results presented on an adjacent display surface.

## 1.6 Thesis statement

This dissertation works to support the following thesis:

> *The use of physically constrained tokens can provide an effective means for interacting with aggregates of digital information.*

Webster's defines "effective" as "producing or capable of producing a result." In the context of human-computer interaction, an effective interface should provide a mechanism for interacting with digital information that has both utility and applicability; and that achieves these ends in a manner that compares favorably with prevailing best-practice.

Toward this, the thesis presents a framework for using physical tokens and constraints as a technique for interacting with aggregates of digital information; and describes two systems that make effective applications of this technique to significant problem domains. Specifically, the thesis will present mediaBlocks, a TUI for manipulating aggregates of online digital information within physically situated contexts; and tangible query interfaces, TUIs for expressing, visualizing, and manipulating database queries. The thesis will argue the significance and utility of both systems, and seek to empirically demonstrate both quantitative and qualitative performance benefits of the latter system over best-practice graphical user interfaces through user studies.

Moreover, I claim that these interaction techniques are applicable for interacting with content spanning a variety of important application domains. For example, queries can be used to frame interaction with many kinds of digital information, including applications in science, industry, and commerce; interfaces to many Web-based services; and many other uses. I have not chosen query tasks because of any unusual suitability to tangible interfaces, but rather as a strong applications example for a major class of information interfaces. I believe that successful applications of TUIs to query interfaces, complemented by positive results from user studies, will offer strong support for the thesis claim that physically constrained tokens can provide an effective approach for interacting with aggregates of digital information.

## 1.7 Thesis contributions

In supporting the thesis statement, the dissertation makes a number of specific contributions.

1) Identification and characterization of tangible user interfaces as a distinct and cohesive stream of research

The thesis is broadly concerned with the design and characterization of "tangible user interfaces." This approach is supported both by systems designed toward the thesis, as well as other recent and previous work, and has grown into a widely recognized subfield of human-computer interaction. However, the identification, expression, and characterization of tangible interfaces as a distinct and cohesive research stream are original to the work of this thesis.

2) Identification and demonstration of the token+constraint approach, providing physically embodied syntax for the structured composition of physical/digital elements.

The thesis develops the use of physical tokens and constraints as an approach for physically expressing and structuring the syntax of tangible interfaces. This approach is positioned alongside "interactive surfaces" and "constructive assemblies" as a third major approach for the design of tangible interfaces. I argue and empirically demonstrate that token+constraint systems have a number of conceptual and technical benefits, both with respect to traditional graphical interfaces as well as other tangible interface approaches. Simultaneously, physically constrained tokens may also be used in combination with other tangible and graphical interfaces, such as integrating query functionality within tangible interfaces focused upon predominantly geometrical interactions (e.g., urban planning).

3) Proposal and realization of techniques for physically representing and manipulating aggregates of digital information.

Prior to the work of this thesis, tangible interfaces made one-to-one mappings between physical artifacts and individual elements of digital information. This thesis has proposed new techniques and demonstrated systems for interacting with both small and large aggregates of digital information. Specific demonstrations include the mediaBlocks sequencer, with support for physically expressing the aggregation and disaggregation of media elements; and tangible query interfaces, for physically expressing query relations over large volumes of digital information.

4) The mediaBlocks system

The mediaBlocks system develops new techniques for the physical embodiment of online information; physical containers for data aggregates; physical copy and paste between physically situated devices; objects embodying a fusion of physical containment + control ("multiple inheritance"); and applications such as the composing and editing of media presentations.

5) Tangible query interfaces

The two tangible query interface prototypes develop new adjacency-based grammars; physical representation of query parameters; mechanical grouping of multiple tokens (allowing physical manipulation of multiple tokens as an aggregate); parameter wheels; display-integrated double sliders; dynamic binding for objects with embedded displays; and applications for query interactions with large collections of digital information.

6) Experimental comparison of tangible query interfaces with best-practice graphical interface techniques.

Tangible query interfaces will be experimentally compared with best-practice graphical interfaces. These user studies will seek to empirically establish quantitative and qualitative benefits of tangible interfaces utilizing physically constrained tokens.

## 1.8 Thesis overview

The following chapter considers the broad conceptual background underlying tangible interfaces. The chapter begins by considering two historical examples that have been inspirations for this thesis – the abacus and board games. The chapter then provides an overview of relevant perspectives from several scientific and design disciplines. Conceptual perspectives from human-computer interaction are also considered, including several principles and models that broadly relate to tangible interface design. Finally, the chapter discusses several interaction models that are specific to tangible interfaces, both original to this thesis and as developed by others.

Chapter 3 presents specific research systems of relevance to tangible interfaces and the thesis approach. The chapter begins with broad areas of relevance, such as ubiquitous computing and augmented reality. Systems illustrating freely manipulable, mechanically coupled, and physically structured TUI approaches are then considered individually. GUI approaches relating to the task domains of the thesis systems are also considered.

Chapters 4 and 5 present detailed considerations of mediaBlocks and tangible query interfaces, the two major TUI systems supporting the thesis. These chapters discuss both the specific functionality of these systems; the fashions in which these leverage token+constraint techniques; as well as the design and implementational approaches used for these systems.

Chapter 6 presents an experimental evaluation of the one of the tangible query interface prototypes. This chapter develops both quantitative and qualitative support for the thesis approach as an effective means for human-computer interaction.

Finally, Chapter 7 considers some of the implications for relaxing or reversing the underlying TUI properties presented within §1.2. A more subjective discussion on the role and importance of physicality within an increasingly digital world is also considered. The chapter then discusses the balance between physical and digital representations, and considers prospects for the applied use of tangible interfaces following the thesis approach.

## chapter 2

# conceptual foundations

# 2 Conceptual foundations

Humans are clearly no newcomers to interaction with the physical world, or to the process of associating symbolic functions and relationships with physical artifacts. This chapter considers the broad conceptual background underlying tangible interfaces. The chapter begins by considering several historical examples that have been inspirations for this thesis – the abacus, board games, and early token-based accounting systems. Next, an overview will be provided for related areas of study from the social sciences, including psychology, cognitive science, semiotics, and anthropology. Several perspectives from the design community will also be considered. The chapter then turns to the discipline of human-computer interaction, reviewing several principles and models that broadly relate to tangible interface design. Finally, the chapter discusses several models that are specific to graspable and tangible interfaces, both as developed by others and original to this thesis.

## 2.1 Background of motivating examples

Two particular kinds of physical artifacts served as inspirations for this thesis: the abacus and board games. Both are believed to date back on the order of 5000 years, both to Mesopotamian origins among the earliest civilizations of recorded history [Ifrah 2001; Bell 1979; Masters 2002; Britannica 2002a]. In their Roman incarnations, instances of both artifacts shared the same name for many centuries – *tabula,* meaning "table" or "board" – and there is some evidence that the same physical equipment was at times used for both gaming and calculation. [Pullan 1969; Durham 2002a,b; Bell 1979; Kowalski 2002] Another older and less familiar example – the user of clay tokens as the basis of primitive accounting systems – is believed to have flourished between five and ten thousand years ago, overlapping with and perhaps giving rise to the earliest abacus and board games. The history and evolution of these three kinds of artifacts will be considered briefly, with an orientation toward issues of physical form and representation that hold relevance to tangible interfaces.

### *2.1.1 The abacus*

The earliest versions of the abacus are believed to date to Sumerian origins on the order of 2700 BC [Ifrah 2001]. The abacus concept is believed to have grown around the use of tokens upon ruled boards or tables. Beads, pebbles, or metal disks (called *calculi* in Latin) were used as counters. These were placed between or upon the ruled lines of "counting boards" (Figure 2.2).

From early Roman times until ca. 1000 AD, the Latin term for such counting boards was "tabula." A variety of devices and uses appear to have been known by this name. One use was the calculating device that later became known as the abacus. Additionally, an ancestral form of Backgammon (known in an earlier form as "duodecim scriptorum") was also known as Tabula. This was played on boards with a remarkably similar layout to the counting boards, and the two may have been used interchangeably [Durham 2002a,b]. Tabula was also sometimes used as a name for the gridded boards and games that are ancestral forms of chess and checkers.

Tabula often were embodied as marked or grooved boards or tables. Figure 2.1a depicts the Salamis tablet, the earliest known example of such a board, dating to ca. 300 BC. In some instances, deeply grooved lines served as constraints for spherical tokens. This is illustrated by the Roman tabula of Figure 2.1b. A schematic illustration of these early configurations and interpretations appears as Figure 2.2. Descendants of these counting boards continued in active use through the Middle Ages, and remained in widespread use in Europe into the 17th century.

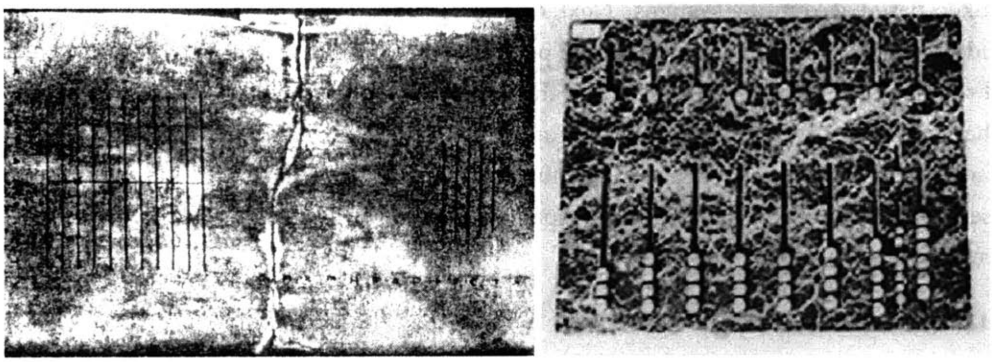


**Figure 2.1a,b: Salamis tablet (oldest known counting board); Roman tabula with spherical pebbles constrained within grooves**

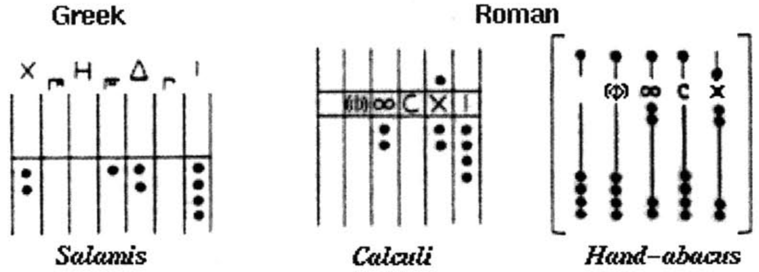


**Figure 2.2: Illustration of ancient counting board approaches** *(from [Fernandes 2000])*

The use of rods and beads within the abacus appeared ca. 1200 AD in China as the "suan pan," and was adopted in Japan as the "soroban" ca. 1600 AD. Examples of these devices appear in Figure 2.3. Instances of both devices continue in modern use, occasionally in hybrid electronic/mechanical forms (as illustrated in Figure 2.4). A Russian abacus tailored for the calculation of rubles and kopecs (the "Ščët") also dates to the 17th century AD. Interestingly, a related abacus form of Aztec origins (the "nepohualtzitzin"), composed of kernels of maize threaded through strings mounted upon a wooden frame, appears to have been used ca. 900-1000 AD.

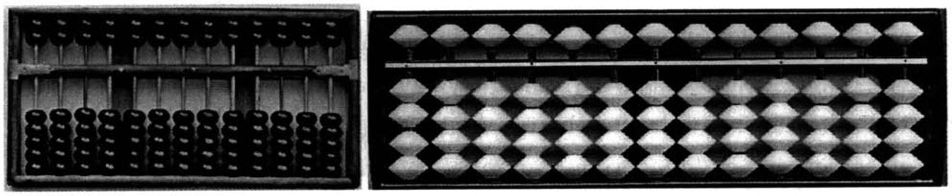


**Figure 2.3: Chinese suan-pan and Japanese soroban** *(figures at different scales)*

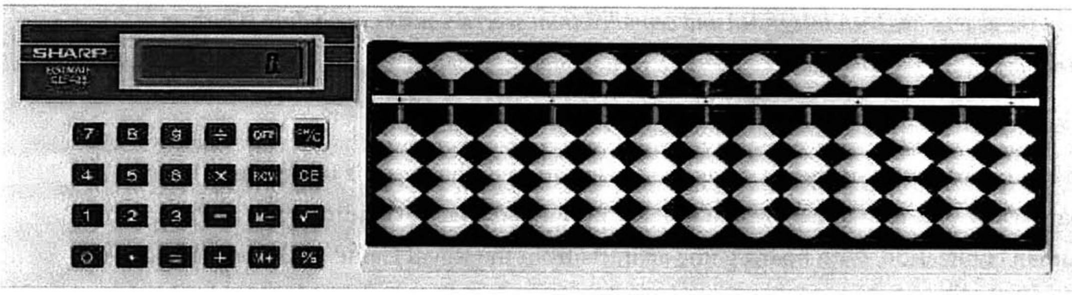


**Figure 2.4: Combination electronic calculator and soroban (Sharp ELSI MATE EL-428)**

The evolution of the abacus has several kinds of implications for tangible interfaces. For one, many historians of computing begin with introducing the abacus as mankind's first computational device. From the perspective of the thesis, the abacus is especially notable for representing both information and the process of calculation in physically embodied form, making no distinction between the dual aspects of representation and control.

The abacus also has important implications for what it was first used to represent, and how it encodes these representations. In the earliest forms of the counting board, physical tokens are believe not to have represented abstract numbers, but real entities within the world – cows, goats, bushels of grain. This will be elaborated in §2.1.3. While evolutions of the abacus introduced the critical element of abstraction (especially for representing large numbers), this symbolic relationship between physical artifacts and specific entities in the world is partly inspirational to the work of this thesis.

Secondly, it is important to note that the abacus represents information not just as discrete physical beads, but also through the spatial structuring and configuration of these elements within the "reference frame" of the counting board or rods. While the pragmatics of mobility and managing numerous physical elements eventually pushed the abacus to a system of captive beads, abacus tokens remained discrete and spatially reconfigurable for much of the device's history. As evidenced by the deeply grooved counting board of Figure 2.1b, some abacus devices closely approximated the interpretive constraint approach that is the focus of the thesis.

Thirdly, it is worth noting that only the discrete state of abacus beads (whether they are pushed to the upper or lower side of the abacus frame) is interpreted by its users. The thesis makes use of both the discrete positions of tokens (both with respect to constraints and to each other) and also to the continuous position of tokens within constraints. The continuous mapping is more meaningful within tangible interfaces because of the active interpretation and feedback of token states provided by TUIs' underlying computers.

These properties, combined with the experiences of Ishii described in [Ishii and Ullmer 1997], have made the abacus a strong inspiration from the earliest days of the thesis. In the middle stages of the thesis, these grew into attempts to directly apply variations of the counting board and abacus forms to the representation and manipulation of digital information. While these efforts did not reach fruition, aspects of the abacus design and function continue to strongly influence the thesis.

*(The history of the abacus has been integrated and compiled from several sources, including [Durham 2002a,b, Fernandes 2001, Lütjens 2002, Tomoe 2002, and Britannica 2000a])*

### 2.1.2 Board games

The second kind of physical artifact that has been inspiration to the thesis is the space of board games. Board, card, and tile games present a richly populated space extending back to the dawn of human civilization, with board game artifacts from the Royal Game of Ur dating to ca. 2500-3000BC [Bell 1979; Masters 2002]. Prototypical instances such as chess and mancala clearly illustrate systems of physical objects – i.e., the playing pieces, boards, cards, and counters – unioned with the abstract rules and relationships these objects symbolically represent. Viewing examples such as in Figure 2.5, and imagining these physical tokens to digitally represent people, places, devices, data structured, and software, provides a stimulating point of departure for envisioning potential tangible interface.

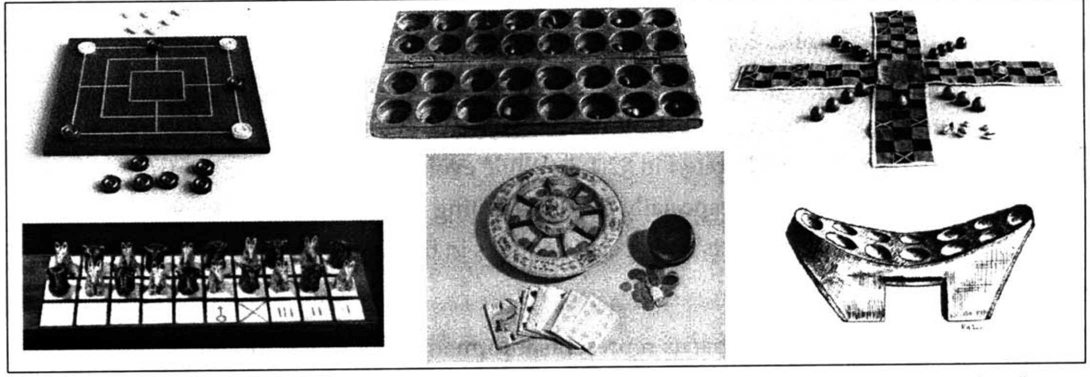


**Figure 2.5: Example board games** *(Nine Men Morris; Mancala; Pachisi; Game of Thirty; Pope Joan; Awari)*

The chess and mancala examples also begin to suggest relationships between the physical form and cognitive "function" of game artifacts. The Oxford Guide to Card Games notes that "what you play with governs what you play, as you will soon discover if you try playing football with a shuttlecock or ping-pong with a puck" [Parlett 1990]. Clearly, one would be hard pressed to play mancala with the physical "equipment" of chess, or vice versa. But why is this necessarily so?

In the athletic games of the Oxford quotation, the laws of physics that govern how physical form resolves into mechanical behavior are readily identified. In the case of non-athletic games, however, the physics of chess and poker kings and queens are more complex to resolve. If poker cards are rendered in wood or steel, the game likely can go on (although shuffling and other aspects of play may be impeded). But if the cards are transformed into (say) labeled balls, it is no longer clear whether the game of poker is still possible.

I discuss the relationships within non-athletic games between physical form and "cognitive function" in [Ullmer 2000a], focusing on cognitive science ideas about external representation and diagrammatic representation. These related areas are briefly reviewed in §2.2. For the purposes of this thesis, several points are worth summarizing.

First, board games offer compelling examples for how abstract rules and relationships can be encoded within systems of physical objects. For example, the game of Monopoly™ utilizes distinct physical representations for people (player tokens), physical entities (house & hotel tokens), properties (printed upon the board), money, actions (through several kinds of cards), and elements of chance (the dice). These roles are encoded within diverse physical tokens, cards, and the playing board itself, each with different physical properties for manipulation and use. Some elements of the game engender information hiding and privacy (esp. one-sided cards), while others facilitate shared state (esp. the tokens and board). Some representations are borrowed from other contexts (e.g., paper money and dice) while others are original to the game. Also, the game affords interaction not only between people and information, but also between multiple people, in a compelling and engaging fashion.

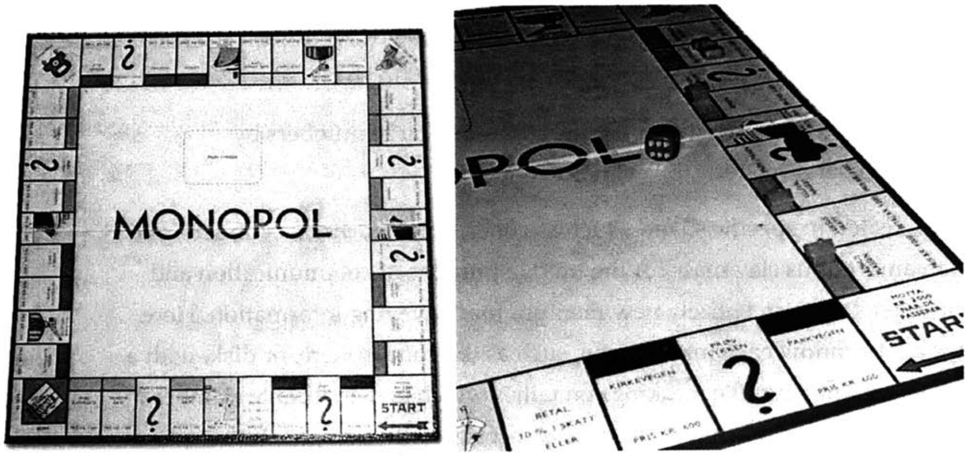


**Figure 2.6: Monopoly™ game, overview and in use**

Secondly, board games can suggest specific physical structures and actions that can be employed within tangible interfaces. The "rack" structure of the thesis was inspired in part by two such examples: the word blocks and Scrabble™ tile rack of Figure 2.7. In both cases, a series of physical tokens are constrained within a linear rack constraint to facilitate the composition of words or sentences. While in these examples configurations of objects are interpreted only within the mind of the human user, these examples lend themselves well to the variety of digital interpretations developed within this thesis, as well as to other possible variations.

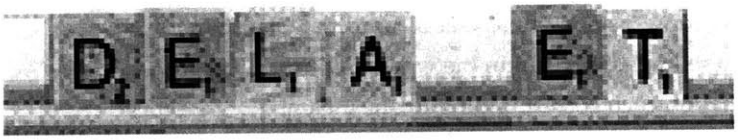


**Figure 2.7a,b: Word blocks and Scrabble™ tile rack examples**

### *2.1.3 Accounting with clay tokens*

Both board games and the abacus are familiar artifacts that continue into present-day use (at least within some circles). Late in the thesis, I learned that these artifacts – and by some well-established accounts, written language itself – grew from a much older language of physical tokens that flourished between 5,000 and 10,000 years ago. While this example was not directly "motivational" to the thesis, the "language" of these artifacts is closely relevant, and merits discussion in this section.

Clay tokens were used as the basis of the first widely used communication system external to the human body. They were first developed in Mesopotamia ca. 8,000 BC, and continued in use until ca. 3,000 BC in an area ranging from the Mediterranean to Turkmenistan (to the north of Afghanistan) [Schmandt-Besserat 1996; Streeck and Cole 1991]. To place this timeframe in perspective, wheels are thought to have been first developed ca. 3,500 BC; writing, ca. 3,200 BC; and the first cities (Ur and Uruk), ca. 5,000 to 3,500 BC [Britannica 2002d,b,c].

Writing on the function of these tokens (which have been found in large numbers by archeologists), Schmandt-Besserat relates:

> Tokens were artifacts created in specific shapes, such as cones, spheres, disks, cylinders, and tetrahedrons from an amorphous clay mass for the unique purpose of communication and record keeping. The tokens were an entirely new medium for conveying information. Here the conceptual leap was to endow each token shape, such as the cone, sphere, or disk, with a specific meaning. Consequently, unlike markings on tallies which had an infinite number of possible interpretations, each clay token was itself a distinct sign with a single, discrete, and unequivocal significance. While tallies were meaningless out of context, tokens could always be understood by anyone initiated into the system. [Schmandt-Besserat 1997, p.93]

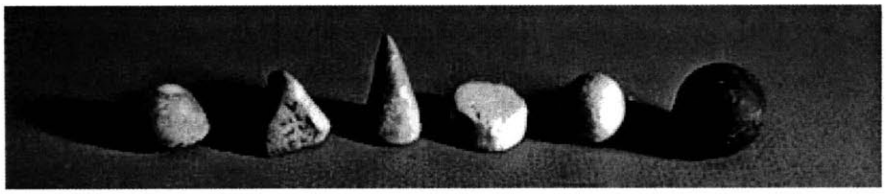


**Figure 2.8: Clay tokens representing particular kinds of animals and other physical entities,** used as the basis of early Mesopotamian accounting systems between ca. 8000-3000 BC

These tokens served not only as an accounting system, but also as a coordinated system for categorizing and communicating certain relationships within the world. The tokens were formed into a limited set of recurrent shapes, each corresponding to one meaning or category (e.g., "sheep" or "container"). In their earliest incarnations, tokens were simple geometric shapes. In later variations, they were inscribed with markings such as lines, dots, and circles.

Several additional observations about the clay tokens are worth mention. One such observation relates to the potential religious or spiritual significance that may associate certain forms of representation. Speaking of the possibilities of taboos that might have discouraged literal representations of objects in the world, Harris writes:

> ...The existence of such taboos could explain, in fact, the earliest adoption of 'arbitrary' symbols or tokens... perhaps a compromise between the significance of the emblem and the insignificance of the pebble. However... we do not know what symbolic or emblematic values may have been originally attached to shapes and configurations which to a modern eye appear to be simply 'geometric'. [Harris 1986]

These observations relate to the discussions of anthropology and magic that will be elaborated in §2.3.2.

Another observation relates to a progressive abstraction in token representation that occurred during the token's long evolution. Writing of [Schmandt-Besserat 1997], Wachtel writes:

> Schmandt-Besserat traces the development of abstract counting (the use of signs to represent the concepts of "oneness," "twoness," "threeness," etc., separate from the actual things being counted). Consequently, a sign that originally stood for "one jar of oil" was replaced by two signs, one for the quantity, one for the jar of oil. Further abstraction led to the use of three signs, one for the quantity, one for the jar, and a third for the oil. [Wachtel 1999]

These comments suggest a comparable sequence of abstraction that might be applicable for tangible interfaces. However, Wachtel goes on to speak more of the limitations of pictographic systems for expressing more complex concepts, and related arguments may also apply to tangible interfaces.

Finally, it is interesting to question what became of these early token "languages." Some have suggested that ideomorphic tokens were used in conjunction with early "proto-abacus" [Chizlett 1999]. Perhaps continuing abstractions in form lead to the use of abstract counters observed with the abacus, as well as the contemporaneous evolution of board games. While Hockett suggests some of the useful properties of the token systems (recounted in Appendix C), Schmandt-Besserat also describes some of the pragmatic limitations of the token systems:

> ...Three-dimensionality gave the [token system] the advantage of being tangible and easy to manipulate. On the other hand, the volume of the tokens constituted a major shortcoming. Although they were small, the counters were also cumbersome when used in large quantities. Consequently, as is illustrated by the small number of tokens held in each [clay] envelope, the system was restricted to keeping track of small amount of goods. The tokens were also difficult to use for permanent records, since a group of small objects can easily be separated and can hardly be kept in a particular order for any length of time. Finally, the system was inefficient because each commodity was expressed by a special token and thus required an ever-growing repertory of counters. In short, because the token system consisted of loose, three-dimensional counters, it was sufficient to record transactions dealing with small quantities of various goods, but ill-suited for communicating more complex messages. [1997, p. 98]

These observations are similar to those evoked by Swift with the object-languages of his "Sages of Lagado" [Swift 1726]. The implications of these analyses for tangible interfaces are explored in Chapter 7 as part of a broader discussion of "what should be physical, and what digital."

## 2.2 Perspectives from psychology and cognitive science

Psychology and cognitive science offer perhaps the broadest area of scientific study related to the use of tangible interfaces. At some level, this is closely in keeping with the broader area of human-computer interaction, which also finds specialists from human factors, psychology, and cognitive science among its earliest scientific investigators. Simultaneously, tangible interfaces involve a far longer history and broader range of modalities for engagement between people and computation than graphical interfaces. These have led to the relevance of an even broader range of subdisciplines.

The doctoral dissertations of Fitzmaurice [1996] and Hinckley [1996] have offered detailed and perceptive reviews of the psychology and cognitive science literature with respect to the "input"-oriented aspects of graspable interfaces and two-handed interaction, as well as contributing new studies in these areas. Instead of duplicating these efforts, this section will focus on the representational aspects of tangible interfaces – specifically, the topics of external representation, diagrammatic representation, affordances, and distributed cognition – followed by a briefer positioning of other relevant areas of psychology and cognitive science.

### *2.2.1 External representations*

In recent years cognitive scientists have approached a growing consensus that the process of cognition lies not only in the human mind, but also within the physical world. Researchers including Norman [1993], Zhang [1994], and Scaife and Rogers [1996] discuss cognition in terms of internal and external representations. Internal representations are variations upon traditional "mental models," while external representations are "knowledge and structure in the environment, as physical symbols, objects, or dimensions, and as external rules, constraints, or relations embedded in physical configurations" [Zhang 1994].

In a widely known study into the cognitive roles of external representations, Zhang and Norman studied the games of Tic-Tac-Toe and the Towers of Hanoi. In this work, game isomorphs were developed that preserved the games' abstract rules and structure, while mapping these abstractions to a variety of alternate external representations. In experimental tests of these isomorphs, Zhang and Norman found with statistical significance that each increase in externally represented information yielded successive improvements in solution times, solution steps, and error rates [1994].

Based upon these results, Zhang and Norman asserted that "the physical structures in external representations constrain the range of possible cognitive actions in the sense that some actions are allowed and others prohibited" [Zhang 1994]. Zhang went on to conclude that "external representations are neither mere inputs and stimuli to nor mere memory aids to the internal mind. They are intrinsic components of many cognitive tasks; they guide, constrain, and even determine cognitive behavior" [Zhang 1997].

These observations and experimental results are closely in keeping with the thesis' use of interpretive constraints. Zhang and Norman's results support the assertion that these physical structures can serve not just as mechanical aids, but also as cognitive aids for structuring and guiding the interaction task.

An interesting distinction between the Towers of Hanoi and Tic-Tac-Toe is that Tic-Tac-Toe is often written in purely visual form. Once inscribed, the game's crosses and circles are no longer physically manipulable. In contrast, the disks of the Towers of Hanoi (and the isomorphs explored by Zhang and Norman) have discrete physical embodiments, therefore remaining physically manipulable.

The different properties of written versus physically manipulable external representations appears to be largely without development in the academic literature. Confirming this in personal communications, Zhang said:

> The reason we used physical objects (instead of symbols/objects on computer screens) for the Tower of Hanoi study was primarily due to our belief that real physical/graspable objects were different from written symbols. However, ... I have not been aware of any extensive studies on this. [Zhang 1999]

### *2.2.2 Diagrammatic representation*

The area of diagrammatic representation offers a more specific investigation into the role of external representations upon cognition. This subdiscipline grew out of studies upon how people engage with diagrams upon paper and other physical media. In recent years, it has frequently been studied in conjunction with the development of visual languages and visual programming languages for human-computer interaction. While this area of work has historically focused on visual notations, many of its lessons are relevant to the kinds of physical descriptions employed by tangible interfaces.

In a seminal paper by Larkin and Simon titled "Why a Diagram is (Sometimes) Worth Ten Thousand Words," the authors relate:

> The advantages of diagrams, in our view, are computational. That is, diagrams can be better representations not because they contain more information, but because indexing [such as spatial juxtaposition of related elements]... can support extremely useful and efficient computational properties. [Larkin and Simon 1987]

Lewis and Toth present an interesting interpretation of this conclusion:

> The primary advantage of visual representations such as diagrams is attributed to apprehension of dynamics rather than the explicit representation of state (Larkin and Simon, 1987), although the model accounts for both. [Lewis and Toth 1992]

These remarks have several implications for tangible interfaces. Among others, they suggest that even among TUIs and GUIs that utilize similar visual representations, tangible interfaces may draw strength from their strong indications of potential dynamics. This might hold special relevance for the physically embodied structures of interpretative constraints, which not only contribute to the "apprehension of dynamics," but also enforce these mechanically, contributing to learning effects, muscle memory, and the development of physical skills.

Larkin and Simon offer an important tempering remark to their assessment of diagrams, noting:

> ...none of these points insure that an arbitrary diagram is worth 10,000 of any set of words. To be useful a diagram must be constructed to take advantage of these features. The possibility of placing several elements at the same or adjacent locations means that information needed for future inference can be grouped together. It does not ensure that a particular diagram does group such information together. [1987]

Petre elaborates upon this point, saying:

> The strength of graphical representations... is that they complement perceptually something also expressed symbolically. For instance, when functionally-related components are placed close together, which is typical practice in electronics schematics, an analog mapping is being used to supply extra information over and above the information explicitly represented by the components and their connections. Expert designers regard this 'secondary notation' as being crucial to comprehensibility.
>
> ...Much of what contributes to the comprehensibility of a graphical representation isn't part of the formal programming notation but a 'secondary notation' of layout, typographic cues, and graphical enhancements that is subject to individual skill." [Petre 1995]

These comments support the intuition that the representational strength of diagrams and tangible interfaces is not inherent in the use of physical or spatial representation alone, but rather a product of good design. Also, Petre's notion of secondary notations seems to share conceptual ground with Larkin and Simon's emphasis upon the computational aspects of representation, with salience arising through higher-order patterns and relations of underlying elements. Taken together, tangible interfaces can be expected to succeed (or fail) largely on the basis of good design, both as a product and as evidenced by the patterns, tensions, and interplay of "secondary notation" they evoke.

### 2.2.3 Affordances

Ideas about "affordances" by Gibson, Norman, and others have long been of interest to the HCI community, and hold special relevance to tangible interface design. The "affordance" term was coined by the perceptual psychology J. J. Gibson, and refers to the "complementarity of the animal and the environment" [Gibson 1979]. More specifically from the standpoint of tangible interfaces, Gibson writes:

> The affordances of what we loosely call *objects* are extremely various... Some are graspable and other[s] not. To be graspable, an object must have opposite surfaces separated by a distance less than the span of the hand. A five-inch cube can be grasped, but a ten-inch cube cannot. [Gibson 1979, p.133]

This observation has a number of specific implications for tangible interfaces. As one interesting instance, a number of tangible interfaces have converged on "modes" of cubical or rectangular objects of 10cm or 5cm per side. For instance, systems by Frazer [1983], Anagnostou et al. [1989], Suzuki and Kato [1993], and Shießl [2001] all independently converged upon cubes of roughly 10cm/side (see Figure 2.9).

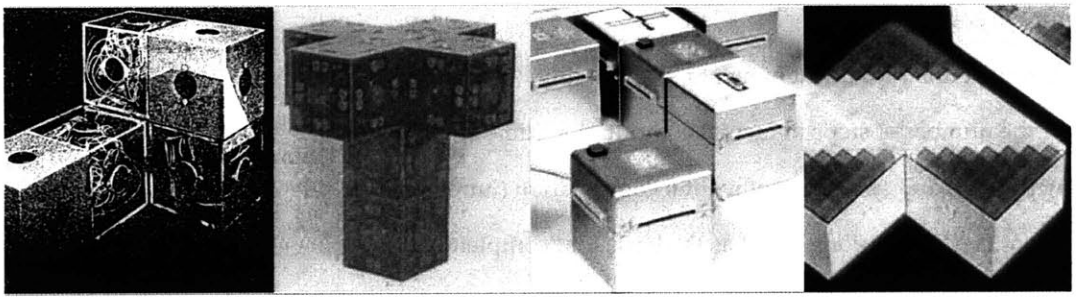


**Figure 2.9: Cubes of Frazer [1982], Anagnostou et al. [1989], Suzuki and Kato [1993], Shießl [2001]**

Similarly, in the mediaBlocks project of this thesis, a face size of 5cm square was chosen both on the basis of design iterations (see Figure 4.12), and on the size of 35mm slides. It was soon discovered that this 5cm size not only afforded use of the mediaBlocks together with a range of racks and boxes designed for 35mm slides, but also with a wide array of other drawers and other organizing structures commonly available toward a range of intended purposes.

These sizes seem to reflect the anatomy of the human hand. Napier [1956], Cutkosky and Howe [1990], and MacKenzie and Iberall [1994, pp.24-28] classify grasping postures of the human hand into "precision" and "power" postures. The broad space of precision postures and the "heavy wrap, large diameter" posture (#1) from Cutkosky and Howe [1990] seem to correspond with the above 5cm and 10cm diameter forms.

In addition to physical size, physical shape and structure can also offer important affordances. For example, regularly shaped physical blocks afford stacking, as illustrated by Figure 2.9a,b. The token+constraints approach also builds upon ideas about affordances, using mechanically interacting elements to physically suggest how physical objects should be combined and constrain how they can be manipulated.

### *2.2.4 Distributed cognition*

The topic of external representation is often discussed in the broader context of "distributed cognition." The distributed cognition approach "explores how cognitive activity is distributed across internal human minds, external cognitive artifacts, and groups of people, and across space and time" [Zhang 1997].

Several elements from the work of Hutchins and Kirsh help to illustrate the relevance of this approach for tangible interfaces. The relevance of Kirsh's work to graspable interfaces, and by extension to tangible interfaces, is well summarized by Fitzmaurice [1996, pp. 19-23]. One important aspect concerns the relationship between physical manipulation and exploration with the hands [Klatzky and Lederman 1990]. People's manipulation of objects can be divided into of "exploratory" and "performatory" actions [Gibson 1962], or alternately "epistemic" and "pragmatic" actions [Kirsh and Maglio 1994].

Epistemic actions are performed to uncover information that is hidden or hard to compute mentally. Examples include the use of fingers while counting, or the tentative manipulations of pieces in playing games such as chess. As reported by Kirsh and Maglio [1994], epistemic actions can improve cognition by

- Reducing the memory involved in mental computation (space complexity)
- Reducing the number of steps in mental computation (time complexity), or
- Reducing the probability of error of mental computation (unreliability).

Epistemic actions can also be considered as the basis for "complementary strategies" for cognition. These are described by Kirsh as:

> ... any organizing activity which recruits external elements to reduce cognitive loads. Typical organizing activities include positioning, arranging the position and orientation of nearby objects, writing things down, manipulating counters, rulers or other artifacts that can encode the state of a process or simplify perception. [Kirsh 1995]

These analyses help illustrate how the physical artifacts of TUIs can serve as more than functional "pragmatics" for steering a man/machine interface. More broadly, these artifacts fill the role of "tools for thought" that support cognitive roles extending beyond the interface's explicit digital functionality. Kirsh's distinction between pragmatics and epistemics also relates to consideration of "in-band" vs. "out-of-band" reconfigurations of TUI elements. "In-band" manipulations are sensed and interpreted by the computational system, while "out-of-band" manipulations serve epistemic roles. These ideas will be developed further later in the thesis.

In another thread of work upon distributed cognition, Hutchins has analyzed observations of individual and group activity aboard naval vessels [1995]. In a section titled "Communication in a shared world," he presents an example analysis of coordination between seven people on the bridge of a ship:

> The plotter's use of his finger in locating the bearing in the bearing record log is very interesting. Because the bearing log is a memory for the observed bearings in this distributed cognitive system, the plotter's action is part of a memory-retrieval event that is internal to the system but directly observable. From the perspective of the individual, the technology in use here externalizes certain cognitive processes. This permits some aspects of those processes to be observed by other members of the team. Because of this, the chief's pointing can be both a part of his private cognitive processing and an element of communication to the recorder [another person on the ship's bridge]. Some kinds of media support this sort of externalization of function better than others. [Hutchins 1995, p. 236]

Here, Hutchins observes how the manipulation of physical artifacts, and the use of physical gestures within these contexts, can serve as critical mechanisms not only for individual cognition, but also for facilitating group communications. As observed in work such as [Cohen et al. 1999] and [Ishii et al. 2002], tangible interfaces' support for group communications appears to be one of their clearest and most compelling virtues. This aspect has been the focus of study by Hornecker (e.g., [2002]), and will be considered further in §2.7.

### *2.2.5 Other psychological and cognitive science perspectives*

A broad variety of other relevant literature exists in the areas of psychology and cognitive science. From a motor psychology perspective, Guiard is widely cited for his studies of human skilled bimanual (two-handed) action [Guiard 1987]. These studies highlight the ways in which the two hands are used in asymmetric, complementary fashions, with the non-dominant hand often establishing a reference frame within which the dominant frame operates. Hinckley extends these ideas in his discussion of bimanual frames of reference [Hinckley 1996]. Of particular relevance to this thesis, Hinckley observes:

> When physical constraints guide… tool placement, this fundamentally changes the type of motor control required. The task is tremendously simplified for both hands, and reversing roles of the hands is no longer an important factor. [Hinckley 1996]

Other studies and analyses discussing two-handed manipulation include [Fitzmaurice 1996, Balakrishan and Hinckley 2000, Leganchuk et al. 1998].

The subfield of spatial representation considers how people establish cognitive representations of spatial relationships. Ideas about "spatial prepositions" and object/reference frame systems are discussed later in this chapter within §2.6.3. Additional literature on this area appears in sources including [Eilan et al. 1993].

Activity theory is another approach that has recently been used to describe graspable and tangible interfaces. This approach offers additional perspectives upon how internal mental activity is "exteriorized" in the form of physical artifacts and tools [Fjeld et al. 2002]. Green's "cognitive dimensions of notations" model [Green 2000] is also of relevance, and will be discussed briefly in the next chapter within the context of visual languages.

## 2.3 Perspectives from other social sciences

### *2.3.1 Semiotics*

The discipline of semiotics – the study of signs and symbols – is concerned in part with the symbolic role of physical objects. A relatively large body of work has considered semiotic analyses of GUI "icons." For example, Familant and Detweiler discuss seven attempts at taxonomies for GUI icons [1993]. Given the richer material and contextual attributes of physical artifacts over their graphical counterparts, semiotics might hold even stronger relevance for the consideration of TUIs.

Many taxonomies for GUI icons have been grounded upon the discipline of semiotics – in particular, the Peircian notion of signs and the trichotomy of icons, symbols, and indexicals. Familant and Detweiler note that "according to Peirce, a *sign* 'is something which stands to somebody for something in some respect or capacity.' ... For Peirce, an icon is a sign that shares characteristics with the objects to which it refers... A symbol stands in an essentially arbitrary relationship to the thing it signifies" [1993]. Alternately expressed, the physical or graphical forms of *iconic* signs share representational properties in common with the objects to which they refer. In contrast, *symbolic* signs need not share such visual or physical references.

It is important to note that the "symbolic" vs. "iconic" distinction is related, but not equivalent, to the issue of "abstract" vs. "literally representational" forms. For example, Gorbet discusses the example of abstraction in comics, where the representation of a character may range from a photograph (uniquely representational) to a "smiley face" (minimally representational) [Gorbet 1998, McCloud 1993]. For Peirce, these continuums of representations are all instances of *iconic* reference. However, if a person is represented with the form of an apple or a geometrical cube, a *symbolic* reference is being used. Both iconic and symbolic forms of representation have been widely used within tangible interfaces.

Additionally, semioticians Krampen [1986, 1979], Rossi-Landi [1975], Prieto [1975], Moles [1972], Boudon [1969], and von Uexkull [1970] have considered the relation of physical tools to human language, grammars, and semantics, which has potential relevance to the characterization of tangible interfaces. At the same time, in considering the semiotics of computer science, Anderson offers the tempering note that semiotics has traditionally been situated as the study of what already exists, rather than as a design perspective for what has yet to be created [1997].

### 2.3.2 Anthropology

Another broadly relevant perspective comes from the field of anthropology, which is concerned with physical artifacts as reflections upon human society and cultural evolution. Some of the artifacts discussed earlier in the thesis have been the focus of anthropological study. For instance, board games have been studied from an anthropological perspective (among other contexts) [de Voogt 1998, Townshend 1986], as have the clay accounting tokens of Schmandt-Besserat [1996]. Another relevant artifact is the Lukasa "memory board," a mnemonic device once used by the Luba people of Zaire (Figure 2.10). The Lukasa were hand-held wooden objects studded with beads and pins, and used to teach lore about cultural heroes, clan migrations, and the like. The kinds of information encoded within such boards included journeys, kings and courtiers, genealogies, and lists of clans [Roberts 1994].

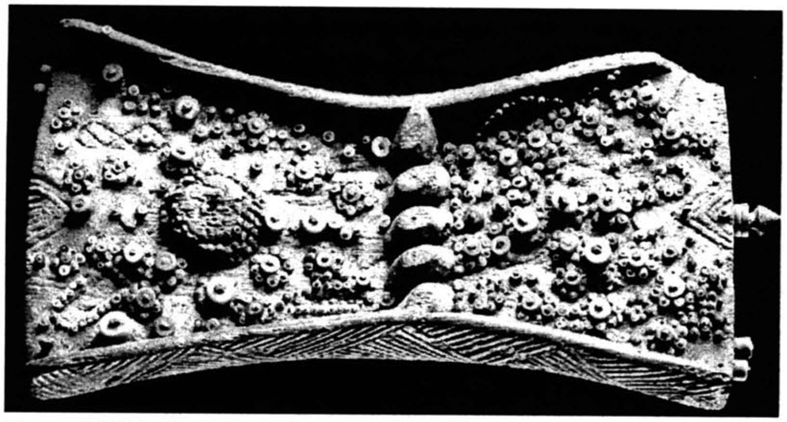


**Figure 2.10: Luba Lukasa "memory board"**

In more recent history, map rooms, "war rooms," and control rooms offer other examples of symbolic and iconic uses for physical artifacts. "In/out" boards, magnet boards, and LEGO boards are sometimes used with reconfigurable tokens for groups to collaboratively track and explore time-evolving processes. In domestic contexts, people use souvenirs and heirlooms as representations of personal histories [Csikszentmihalyi and Rochberg-Halton 1981, Gonzalez 1995].

The study of magic, magical artifacts, and associated cultural beliefs is another area of anthropological concern that holds relevance for tangible interfaces. The relations between magic and tangible interfaces have been discussed at length by both Hadis [1997] and Svanæs and Verplank [2000].

The topic of magic can be approached from several quite different perspectives. One of these relates to stage magic, slight-of-hand, and illusion, which have been considered as metaphors for user interface by works such as [Tognazzini 1993] and [Lokuge 1995]. Another perspective is that of magic as a system of belief, usually relating to the supernatural.

The magic of the supernatural or phenomena "beyond the laws of physics" is not new as a metaphor. Perhaps the most common invocation is the well-known "law" of science fiction author Arthur C. Clarke that "any sufficiently advanced technology is indistinguishable from magic." Speaking of the relationship between magic and technology, Hadis says:

> Magic is "what happens" when technology is sufficiently advanced that the chains of physical causation... become hidden from or incomprehensible to the user. ... One is more inclined to call a technology "magical" when it shatters our fundamental perceptions of space, time, or physical causation. [Hadis 1997]

One of the classic descriptive works of magic, *The Golden Bough* [Frazer 1890], attempted to systematize postulates of magic throughout different cultures and societies. In this text, Frazer postulated two "laws" of magic: the "law of similarity" and "law of contact (or contagion)." The law of similarity proposes that "like produces like, or that an effect resembles its cause."

The law of contagion asserts that objects "once in physical contact with each other later continue to act on each other at a distance." [Frazer 1890]

These two "laws" correspond loosely with common tangible interface approaches. The "law of similarity" overlaps broadly with TUI uses of physically representation artifacts, where physical manipulation of the artifact invokes corresponding digital operations. The neurosurgical props of Hinckley et al. [1994] and the bottles of Ishii et al. [2000] offer quite different illustrations of this "magical law." Alternately, a number of different tangible interfaces have explored the use of paired objects at a distance, where physical actions upon one object cause a corresponding effect upon the remote peer. Examples of such systems include InTouch [Brave et al. 1998], Personal Ambient Displays [Wisneski 1999], Hummingbird [Holmquist et al. 1999], and the White Stone [Tollmar et al. 2000]. Many of these systems implicitly or explicitly invoke the "law of contagion."

The presence of common magical "laws" across many different cultures suggests that deeper psychological predispositions may be at work. In this sense, concepts from magical traditions may hold insights and relevance for user interface design. At the same time, the "opacity" of magic as a user interface metaphor – the masking from user knowledge of the internal function or assumptions of the interface – place it partially in opposition to perspectives such as education, which generally seek to foster transparency and comprehensibility within user interfaces.

## 2.4 Perspectives from design disciplines

The disciplines discussed in §2.2 and §2.3 are devoted primarily to the study of artifacts and phenomenon that already exist. In contrast, the thesis is concerned primarily with the design and realization of new artifacts and interactions.

The word "design" holds many meanings, with widely differing connotations in different professional contexts. [Webster 1999a] defines the word as:

*1: to create, fashion, execute, or construct according to plan*

*4a: to make a drawing, pattern, or sketch of; b: to draw the plans for*

While this is a reasonable start, it leaves an industrial designer, mechanical engineer, electrical engineer, and software engineer with equal claim to the term for describing their work. In the following discussion, design will be considered as the expression of the physical, visual, and behavioral form of man-made artifacts for which the principle role is engagement with people, and more especially people qua people.

Where perspectives such as human factors are concerned with the mechanistic efficiency or safety of human engagement with physical artifacts, design perspectives draw from the quest for achieving and resolving a balance and tension between functional and aesthetic factors. Speaking of such issues, the Encyclopedia Britannica notes that "as with some other arts, the practice of architecture embraces both aesthetic and utilitarian ends that may be distinguished but not separated, and the relative weight given to each can vary widely from work to work"

[Britannica 2000]. This assessment is broadly descriptive of the role of design, and is specifically relevant to the design of tangible interfaces.

Clearly, the relevance of design to computing is far from new. Graphic design has long been an important component in the development of graphical user interfaces, and has played a major role in the growth of the World Wide Web. At the same time, the relevance of design disciplines to tangible interfaces is particularly strong.

Creators of graphical interfaces usually build upon standardized, commercially produced interface and computing hardware, programming environments, and user interface toolkits. In contrast, creators of tangible interfaces must play a much larger role in realizing both their physical and computational aspects, leaving less of a "safety net" to buffer from ill-conceived or ill-executed approaches. Moreover, tangible interfaces' cost and consumption of resources (by a variety of metrics) may be expected to exceed those of graphical interfaces. These factors suggest that tangible interfaces not only face new design and implementational challenges, but also must pass a higher bar than their graphical kin to merit adoption.

A number of different design disciplines have bearing upon tangible interfaces, including industrial design, graphic design, architecture, environmental design, and interaction design. Industrial design is prototypically concerned with the physical form of products intended for mass production and distribution. Graphic design relates to the visual form of media, interweaving typography, images, and other visual elements toward the ends of communication. Architecture relates to the realization of large-scale built and especially habitable physical forms, where environmental design is concerned more with the spaces within and between architectural structures. More recently, interaction design has developed as a discipline concerned with the interplay between physical and visual form and computationally mediated behaviors within interactive products.

Each of these disciplines has relevance to the broad design space of tangible interfaces, each offering different perspectives on physical mediums, scale, mobility, uniqueness of artifacts, cost, and modalities of engagement. From the narrower perspective of this thesis, the field of industrial design holds the most direct relevance. To illustrate the role and influence of industrial design, it is useful to consider a case example.

### *2.4.1 Evolution of the radio's physical form*

The discipline of industrial design has had a profound impact on the physical forms of many modern artifacts, with the automobile, telephone, and radio as especially well known examples. In the case of radio, Adrian Forty describes their physical evolution as passing through three stages of design [Forty 1986].

In the first stage, Forty says "the earliest sets... reflected the almost total pre-occupation of both manufactures and public with the technical properties of the apparatus" [1986, p.201]. Systems such as the Pye "unit system" receiver (Figure 2.11a, 1922) appear little evolved beyond their form upon the experimenter's workbench, exposing every element of the system to immediate access and manipulation (whether intentional or not). Systems such as the Burndept IV receiver (Figure 2.11b, 1924) began to enclose these internal elements, but still more closely resemble laboratory equipment than a "domesticated" appliance.

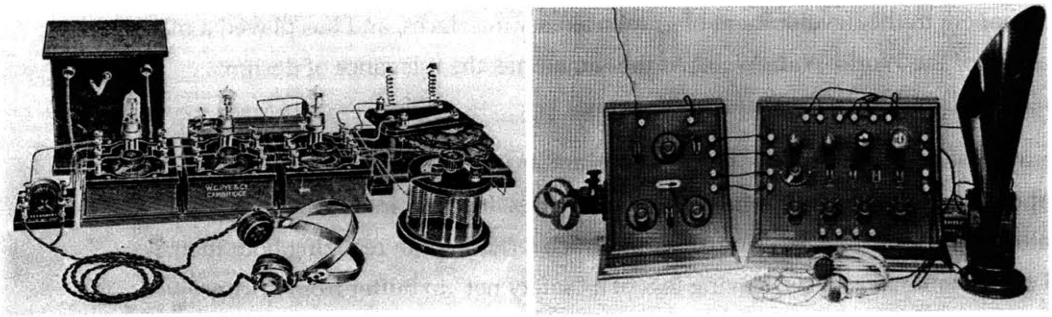


**Figure 2.11a,b: Pye "unit system" radio receiver, 1922; Burndept IV receiver, 1924 [Forty 1986]**

By the mid-1920s, the radio began to enter another stage in development:

> In the second stage of development, manufactures, unable to compete with each other by technical innovation, turned to other means, notably the design of cabinet, to which little thought had been given previously, except in the most expensive sets. The problem that faced manufacturers was what cabinets should look like. [Forty 1986, p202]

Perhaps the most common design approach in this stage was that of furniture. Often this took the form of wooden cabinets which often could not be recognized as radios when closed, as in the custom designed high-end set of Figure 2.12a. In some cases, this approach led to more radical disguises of the underlying technology, such as the radio "easy chair" of Figure 2.12b.

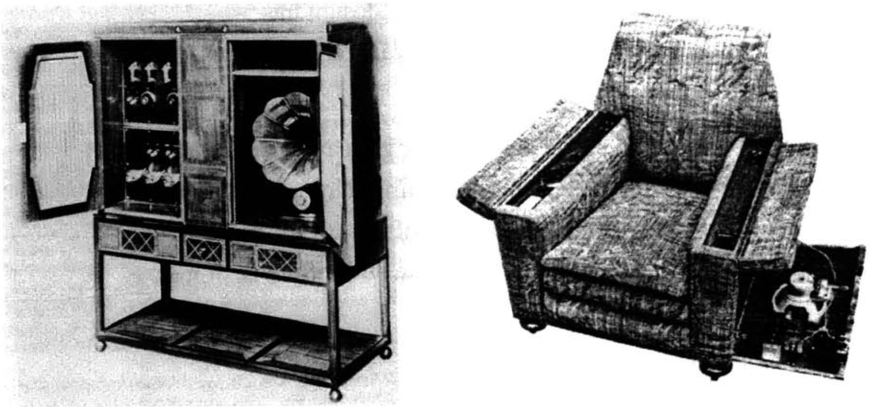


**Figure 2.12a,b: Radio designed by Sir Ambrose Heal, 1924; Radio "Easy Chair," 1933 [Forty 1986]**

In the third stage of design, radios began to realize new physical forms that were neither ad-hoc nor imitative, but rather reflected new design vocabularies for a new kind of artifact:

> The convention of housing wirelesses in cabinets that were identifiable as pieces of furniture gave radio an image which, while it might have been convenient, was not particularly true to its nature.... [With] the introduction of transistors... the majority of sets became portable, [and] furniture ceased to have any relevance to cabinet design.... [In successively emergent

> designs], designers were able to make [an] entirely novel way in which inanimate objects convey this message [communicating their function] without the assistance of words or pictures. [Forty 1986, p202]

Examples of this third stage approach can be seen in the designs of Figure 2.13, especially in the design of Coates.

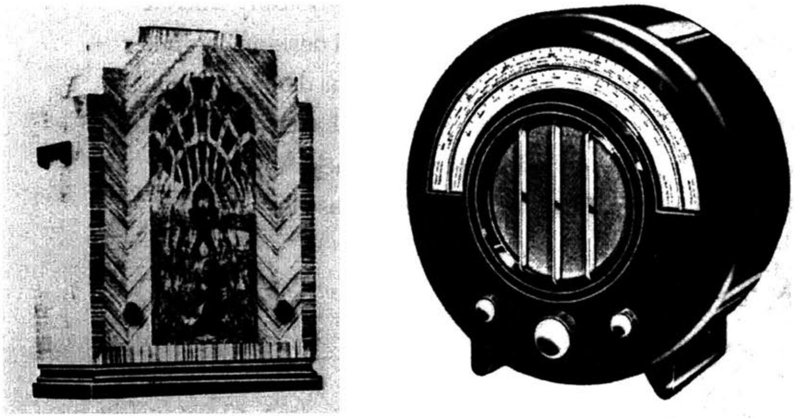


**Figure 2.13a,b: Radio cabinet design, 1932; Ekco AD65 receiver, 1934, Bakelite cabinet, designed by Wells Coates [Forty 1986]**

In these early stages of the radio's development, the engineering challenge lay with the reduction to practice of the underlying technology, toward the ends of fitting the resulting "radio" into a box. In contrast, from the perspective of industrial design, the problem was roughly the opposite – getting the radio *out* of the box (or the cabinet, in this case), and separating it from prior design vocabularies to find its own voice.

Returning to the present day, several very different kinds of trends may be observed. On the one hand, the radio as a distinct appliance has in many cases disappeared into highly integrated "music devices," or even into more broadly functional media and computing devices. Here, the radio may be physically externalized at best with a button for the "radio behavior," with tuning accomplished through multi-purpose "up/down" buttons or numeric keypads.

On the other hand, the advent of the Internet poses new opportunities and demands upon the future of radio that may drive new evolutions in physical form. Instead of radio's early identify as a dozen or two channels spanning 20 MHz of broadcast spectrum, Internet-born audio (whether of streaming or file-based content) may originate from any of a hundred million computers, each potentially bearing thousands of local offerings.

The breadth of this space makes navigation with traditional radio-style tuners implausible. Such Internet-based content is currently referenced with the URL-style addresses and hyperlinks of the Web, which is a plausible approach for use within desktop computers. By extension, a user might remotely log into a stand-alone "tuner" to register favorite "Internet radio stations" with the device. However, these kinds of solutions are consistent with Forty's first and second stages of radio design, and considerably removed from designs such as Coates' radio (Figure 2.13b) that reflect the properties of the new medium.

In contrast, tangible interfaces provide a promising direction by which online musical content might be physically embodied. The physical artifacts of TUIs are usually associated with digital media using the indirect reference of ID tags, rather than storing digital media locally, making them technically well-suited for referencing online content. Moreover, the physically discrete nature of TUI artifacts allows them to migrate gracefully between traditional desktop computer interfaces and tangible interfaces situated within the physical environment, as demonstrated with the mediaBlocks system of this thesis. A number of tangible interfaces have begun to demonstrate the "containment" and control of audio and musical content, including mediaBlocks, Oba's Environment Audio concept design [1991], and Ishii et al.'s musicBottles [2000] (Figure 2.14).

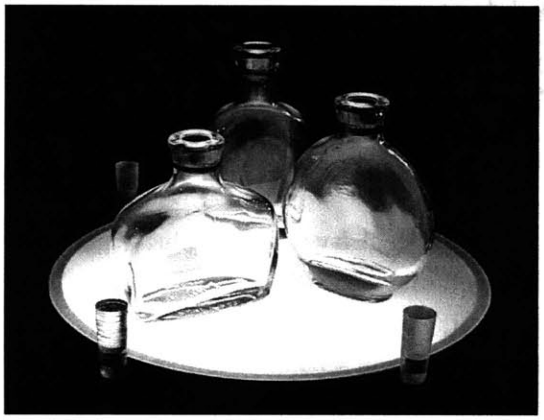


**Figure 2.14: musicBottles "jazz trio" [Ishii et al. 1999]**

### *2.4.2 Other perspectives from art and design*

In the field of industrial design, the literature of product semantics considers in detail the representation of interface semantics within designed physical forms [Vihma 1990]. Relationships between product semantics efforts from the Cranbrook School and early work in tangible interfaces are also discussed in [Gorbet 1998]. From the perspective of the art community, Duchamp's "ready-mades," combining pre-existing, mass-produced physical objects with provocative titles and conceptual positionings, also speak to tangible interfaces' symbolic use of physical objects.

## 2.5 Models from human-computer interaction

The broad discipline of human-computer interaction has also contributed a number of conceptual models that are relevant to tangible interface design. While a broad number of HCI perspectives hold relevance, the "direct manipulation" concept and MVC model have special applicability. Several other outside models specific to graspable and tangible interfaces will be considered in §2.7 and §3.7.

### 2.5.1 *Direct manipulation*

While posed in the context of graphical interfaces, Shneiderman's three principles of "direct manipulation" are also directly applicable to tangible interfaces [Shneiderman 1983]. These principles describe interfaces that provide:

1) Continuous representation of the object of interest
2) Physical actions or labeled button presses instead of complex syntax
3) Rapid incremental reversible operations whose impact on the object of interest is immediately visible

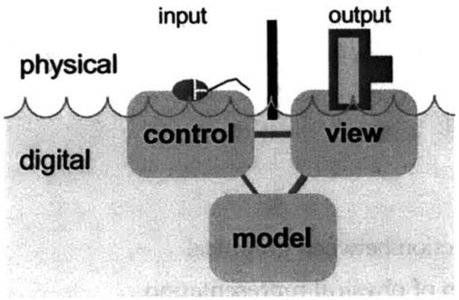
The first principle – "continuous representation of the object of interest" – knits especially well with the persistent nature of TUI tangibles. The second principle also resonates specifically with the thesis approach, in terms of the relationship between physical actions, interpretive constraints, and computational syntax.

For these reasons, the sizable literature relating to direct manipulation, and associated analyses of topics such as perceptual distance, are broadly relevant to TUI design (cf. [Frohlich 1997, Hutchins et al. 1986]). As with other direct manipulation interfaces, TUIs can be said to cultivate tool-like, rather than language-like, modalities of interaction [Smith 1989]. At the same time, tangible interfaces are also subject to some of the criticisms that have been directed at direct manipulation approaches, such as those discussed in [Gentner and Nielsen 1996, Frohlich 1997].

### 2.5.2 *MVC: Model-View-Controller*

Traditional computer interfaces frame human interaction in terms of "input" and "output." Computer output is delivered in the form of "digital representations" (esp., screen-based graphics and text), while computer input is obtained from control "peripherals" such as the keyboard and mouse.

The relationship between these components is illustrated by the "model-view-controller" or "MVC" archetype (Figure 2.15). MVC highlights the GUI's strong separation between the digital representation (or view) provided by the graphical display, and the control capacity mediated by the GUI's mouse and keyboard.



interaction model of GUI:
MVC model (Smalltalk-80)

**Figure 2.15: Illustration of MVC (model-view-control) model**

In its original formulation, MVC served as a technical model for GUI software design, developed in conjunction with the Smalltalk-80 programming language [Burbeck 1987]. However, the MVC model also provides a tool for studying the conceptual architecture of graphical interfaces, and for relating this to the tangible interface approach. While alternate interaction models such as PAC [Calvary 1997] may also hold relevance, MVC is instructive for its exposure of the view/control distinction.

### 2.5.3 *Other HCI models and perspectives*

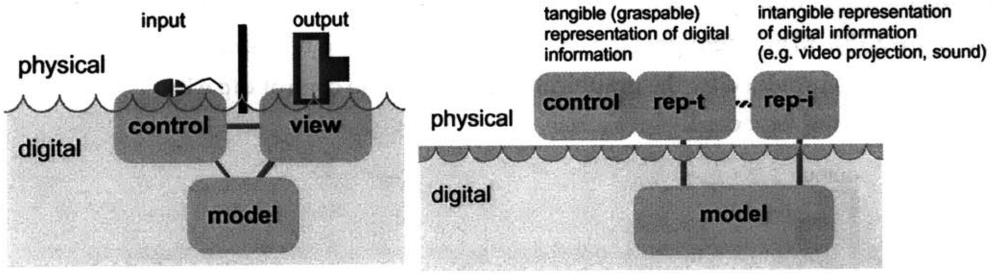
Many other models and perspectives from HCI also hold relevance to the study of tangible interfaces, including PAC (Presentation, Abstraction, and Control) [Calvary 1997] and instrumental interaction [Beaudouin 2000]. The whole sub-disciplines of computer-supported cooperative work (CSCW) and computer-supported cooperative learning (CSCL) are also broadly relevant, including work on topics such as tradeoffs between design for individual and group activities [Gutwin and Greenberg 1998].

## 2.6 Conceptual models within this thesis

In addition to the physical properties discussed in §1.2, the thesis presents several other conceptual models for tangible interfaces. The first of these, MCRit, draws from the MVC approach discussed in §2.5.2. The second develops the "tokens and reference frames" concept in the context of tangible interfaces and interpretive constraints.

### 2.6.1 *MCRit: Model-Controller-Representations (intangible and tangible)*

Drawing from the MVC approach, Ishii and I have developed an interaction model for tangible interfaces called "MCRit," for "model-control-representation (intangible and tangible)." This model is illustrated in Figure 2.16b. We carry over the "model" and "control" elements from MVC, while dividing the "view" element into two subcomponents: *tangible representations* ("rep-t") and *intangible representations* ("rep-i"). (These terms were introduced in §1.6.)



(a) interaction model of GUI: MVC model (Smalltalk-80)

(b) interaction model of TUI: MCRit model

**Figure 2.16a,b: GUI and TUI interaction models**

Where the MVC model of Figure 2.16a illustrates the GUI's distinction between graphical representation and control, MCRit highlights the TUI's integration of physical representation and control. This integration is present not only at a conceptual level, but also in physical point of fact – TUI artifacts *physically embody* both the control pathway, as well as a central representational (information-bearing) aspect of the interface.

This thesis concentrates upon the space of interfaces where each element of MCRit is clearly present, with an emphasis on the role of physical representation. These overlap closely with the five physical properties of tangible interfaces described in §1.2. As with these earlier properties, it is again worth noting that a series of interesting interaction regimes are highlighted by relaxing these expectations. For instance, if the *control* expectations are relaxed (corresponding to the earlier property of "physically manipulable"), the space of "ambient media" is highlighted [Ishii and Ullmer 1997, Wisneski et al. 1998]. These alternate regimes will be considered in the discussion chapter.

### 2.6.2 Key characteristics from the MCRit model

The MCRit interaction model provides a tool for examining several important properties of tangible interfaces. In particular, it is useful to consider the three relationships shared by the tangible representations ("rep-t") of TUIs.

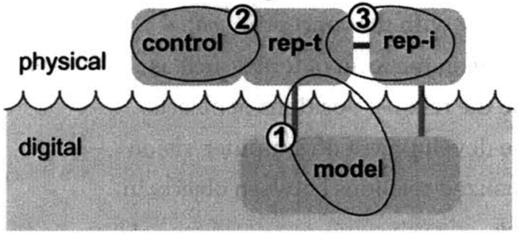


**Figure 2.17: Key characteristics of tangible interfaces**

As illustrated in Figure 2.17, the MCRit model highlights three key characteristics of tangible interfaces.

1) Tangible representations are computationally coupled to underlying digital information

   The central characteristic of tangible interfaces lies in the coupling of tangible representations to underlying digital information and computational models.

2) Tangible representations embody mechanisms for interactive control

   The physical representations of TUIs also function as interactive physical controls. The physical movement and rotation of these artifacts, their insertion or attachment to each other, and other manipulations of these physical representations serve as tangible interfaces' primary means for control.

3) Tangible representations are perceptually coupled to intangible representations

   Tangible interfaces rely upon a balance between tangible and intangible representations. While embodied physical elements play a central, defining role in the representation and control of TUIs, intangible representations – especially, graphics and audio – often present much of the dynamic information processed by the underlying computational system.

Where the above three characteristics refer directly to our MCRit model, a fourth TUI characteristic is also significant.

4) The physical state of interface artifacts partially embodies the digital state of the system.

As illustrated in Figure 2.17, the MCRit model does not specify whether a TUI's physical representations are composed of one or many physical artifacts. In practice, tangible interfaces are generally built from *systems* of physical artifacts. Taken together, these collections of objects have several important properties. As physical elements, TUI artifacts are *persistent* – they cannot be spontaneously called into or banished from existence. Where a GUI window can be destroyed or duplicated at the touch of a button, the same physics do not hold true for physical objects. TUI artifacts also often express *physical state*, with their physical configurations tightly coupled to the digital state of the systems they represent.

### 2.6.3 Tokens and reference frames

Another model referenced earlier in the thesis is the concept of "tokens and reference frames." This concept overlaps with ideas about spatial languages developed within the disciplines of linguistics, psychology, and artificial intelligence, and specifically with the concept of *spatial prepositions*. Spatial prepositions are familiar in natural language in the form of terms such as "in," "behind," and "above," which are often used to describe the relative positions of objects. The topics has perhaps met with the most intense study in the development of computer vision techniques for robotics systems, where the derivation of formalized relations between objects in a scene is important for AI reasoning and planning algorithms.

In a widely cited literature survey and analysis, Retz-Schmidt uses the terms "primary object," "reference object," and "reference frame" for her discussion of spatial prepositions [1988]. In this thesis, the "primary objects" are the physical tokens used to represent digital information. The "reference objects" are the interpretive constraints (such as racks) that are used to frame compositions of tokens. These interpretive constraints establish the reference frames within which tokens are interpreted (an interpretation that Retz-Schmidt describes as the "intrinsic use" of spatial prepositions [1988]).

When approached from this perspective, the physical relationships of tokens within interpretive constraints can be expressed as the combination of several basic properties. Specifically, the physical relationships of Figure 1.5 (reproduced below as Figure 2.18) can be described in terms of the relative and absolute positions of tokens with respect to the interpretive constraint and to each other.

| Physical relationships | Interaction Event | Digital interpretations |
|---|---|---|
| Presence | Add/Remove | Logical assertion; activation; binding |
| Position | Move | Geometric; Indexing; Scalar |
| Sequence | Order change | Sequencing; Query ordering |
| Proximity | Prox. change | Relationship strength (e.g., fuzzy set) |
| ├Connection | Connect/Discon. | Logical flow; scope of influence |
| └Adjacency | Adjacent/NAdj. | Booleans; Axes; other paired relations |

**Figure 2.18: Grammars for mapping physical relationships to digital interpretations**

More carefully stated, the physical relationships of tokens for the grammars of Figure 2.18 can be understood in terms of four basic relationships:

a) *Absolute position* with respect to *constraint*

b) *Relative position* with respect to *constraint*

c) *Absolute position* with respect to *each other*

d) *Relative position* with respect to *each other*

Here, the term "position" is intended to be roughly synonymous with "configuration," and should not be seen as specific to linear constraints. These same concepts are believed to apply to rotational constraints, as well as to more complex mechanical constraints. Also, this use of "absolute" and "relative" is loosely related to the terms "quantitative" and "qualitative." Absolute positions are most often expressed numerically, while relative positions are generally described with spatial prepositions such as "next to" and "left of."

Several configurations illustrating examples of these relationships are shown in Figure 2.19. These are abstractly illustrated, and presented in the context of tokens on a linear interpretive constraint such as the rack.

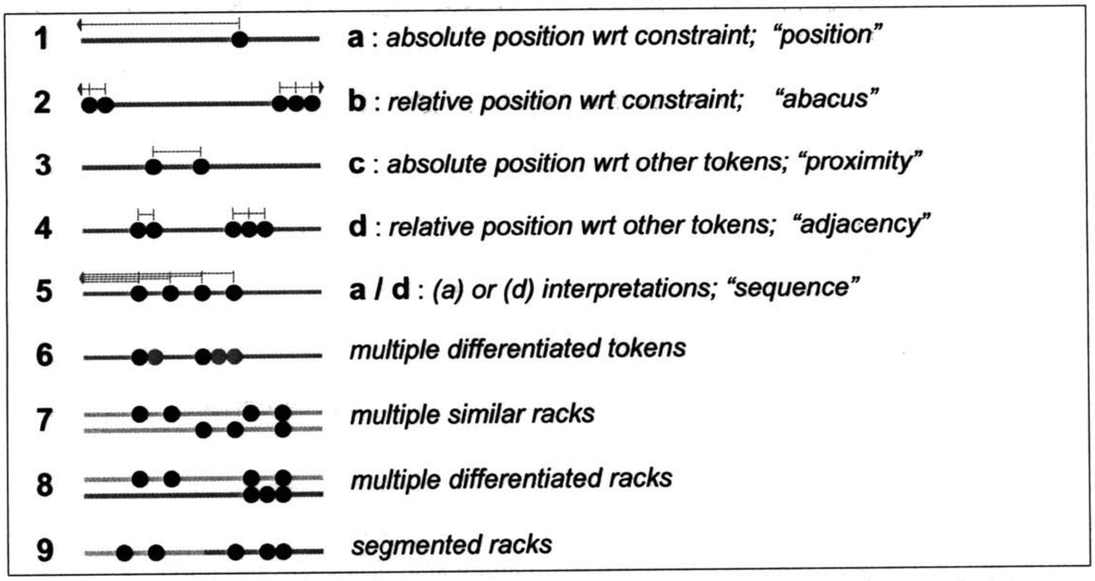


**Figure 2.19: Illustration of different relationships between tokens and interpretive constraints**

The relationships illustrated by examples 1, 4, and 5 are developed by the thesis' "position," "query," and "sequence" racks, respectively. Example 2 describes the kind of physical relationship found within the abacus. While this relationship has not been developed within published TUI research, it was explored in the thesis as an approach for navigating hierarchies, among other possible interpretations. Similarly, while the "proximity" relationship of example 3 has not been published, several observers of the thesis' query interfaces have suggested interpreting this as a kind of "fuzzy logic" relationship between query terms.

Some physical relationships can be described in multiple ways. For example, the "sequence" relationship of example 5 can be described by the absolute position of tokens with respect to the interpretive constraint (the token closest to the left margin is first in the sequence, followed by the next closest token, and so forth). Alternately, the sequence relationship can be described by the relative positions of tokens with respect to each other, using the "left of" spatial preposition.

The first five examples of Figure 2.19 illustrate physical relationships between a single kind of token on a single, undifferentiated rack. Alternately, examples 6-9 illustrate the use of different kinds of tokens, as well as different kinds and combinations of racks. These alternatives open the door to more expressive physical/digital grammars, and hold the potential for greatly extending the power of interpretive constraints. These possibilities will be considered in the discussion chapter (§7.1). Simultaneously, these multiple alternate interpretations again suggest the importance of distinguishing and selecting between multiple alternate mappings, which will be discussed further in §4.10.3.

## 2.7 Other conceptual models for graspable and tangible interfaces

Sections 1.2, 1.4, and 2.6 have presented properties and models that describe tangible interfaces in general, and the interpretive constraint approach of this thesis in particular. In addition, several alternative properties and models for graspable and tangible interfaces have been proposed.

Fitzmaurice offers five "core defining properties" for graspable interfaces [1996]:

1) Space-multiplex both input and output
2) Allow for a high degree of inter-device concurrency
3) Increase the use of strong specialized input devices
4) Have spatially-aware computational devices
5) Have high spatial reconfigurability of devices and device contexts

Where the properties of tangible interfaces described in §1.2 are motivated by concerns for physical embodiment and representation, Fitzmaurice's analysis is framed in terms of "input devices." The property of "spatial reconfigurability" is shared in common between the two approaches. Fitzmaurice's first two properties are helpful in explicitly identifying the importance of multiple physical devices that are simultaneously manipulable. These are also important properties for tangible interfaces, and are addressed in §1.2's discussion of the "physically discrete" and "physically manipulable" properties.

In more recent work, Hornecker has studied the role of graspable and tangible interfaces in cooperative (multi-user) usage contexts [2002]. Figure 2.20 presents an annotated illustration from this research. In the diagram's top tier, Hornecker lists two "key characteristics" of graspable interfaces, as identified by the doctoral work of Brauer [1996]. In the bottom tier, Hornecker identifies seven "positive effects" describing aspects of interaction with graspable (and tangible) interfaces that make them well suited for cooperative interactions. The middle tier describes several factors that support and enable these observed effects, and relates these to Brauer's key characteristics.

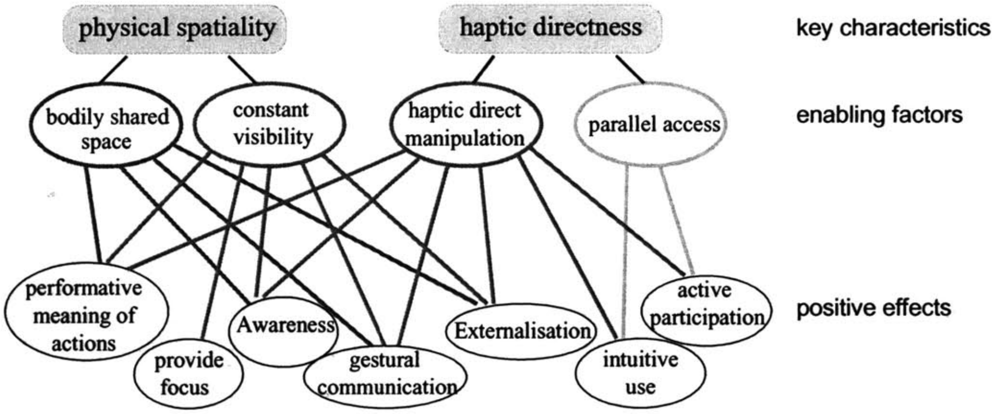


**Figure 2.20: Characteristics, factors, and effects relating to the cooperative use of graspable and tangible interfaces** *(from [Hornecker 2002], with annotations)*

Holmquist, Redström, and Ljungstrand [1999] and Underkoffler and Ishii [1999a] propose useful terminology that attempts to identify and generalize the functional building blocks of tangible interfaces. These will be considered in §3.3 after the introduction of related work.

## 2.8 Summary

The central role of physical artifacts throughout human culture – whether viewed from the eyes of science, technology, the humanities, art, or beyond – has given rise to many diverse approaches of relevance for tangible interfaces. This chapter has provided an overview of several perspectives from the social sciences, design, and human-computer interaction, albeit with a brevity enforced by the breadth of relevant literature. A number of other relevant areas, such as the discipline of education, remain outside the scope of this thesis.

## chapter 3

# related work

# 3 Related Work

This thesis has developed within the context of a growing wave of research into approaches for linking the physical and digital worlds. From the perspective of the human-computer interface community, this trend has been voiced largely within the past decade. Among published works, Weiser's vision of ubiquitous computing [Weiser 1991] and Wellner's discussion of the DigitalDesk (beginning with [Wellner 1991]) were among the earliest writings, each of seminal impact. The areas of ubiquitous computing, augmented reality, and computer-augmented environments brought continuing research efforts throughout the 1990s.

Simultaneously, many (perhaps even most) of the earliest tangible interfaces originated from outside of the human-computer interaction community. These include efforts from the communities of architecture [Aish 1979, 1984; Frazer 1982, 1995]; education [Perlman 1976, Suzuki and Kato 1993]; mechanical engineering [Anagnostou et al. 1989]; and product design [Oba 1990; Bishop 1992, Polynor 1995].

This chapter provides an overview of these and other related works. The chapter begins with a brief discussion of related subdisciplines of human-computer interaction, including ubiquitous computing, augmented reality, computer-augmented environments, and cooperative buildings.

The chapter continues with a discussion of individual systems sharing the TUI approach. Following the discussion of §1.3, the chapter first considers examples from the two most common tangible interface approaches – "interactive surfaces" and "constructive assemblies." The smaller set of systems sharing ground with the thesis' "tokens+constraints" approach is then considered.

As is common in new areas of research, tangible interfaces and their kin have been described using a wide range of terminology. Additionally, several different schemes for classifying the components of tangible interfaces have been proposed. The chapter concludes with a discussion of TUI terminology and classification schemes.

## 3.1 Related areas

As discussed in the introduction, the concern of tangible interfaces for developing ways to link the physical and digital worlds is broadly shared by several other major research themes. In a Danish language thesis, Svendsen presents several interesting illustrations of the relationships and interplay between many of these themes [2001]. His visual summary of individual themes (Figure 3.1) and broader categories of work (Figure 3.2) provide a useful pointer of departure for comparing and contrasting these different perspectives.

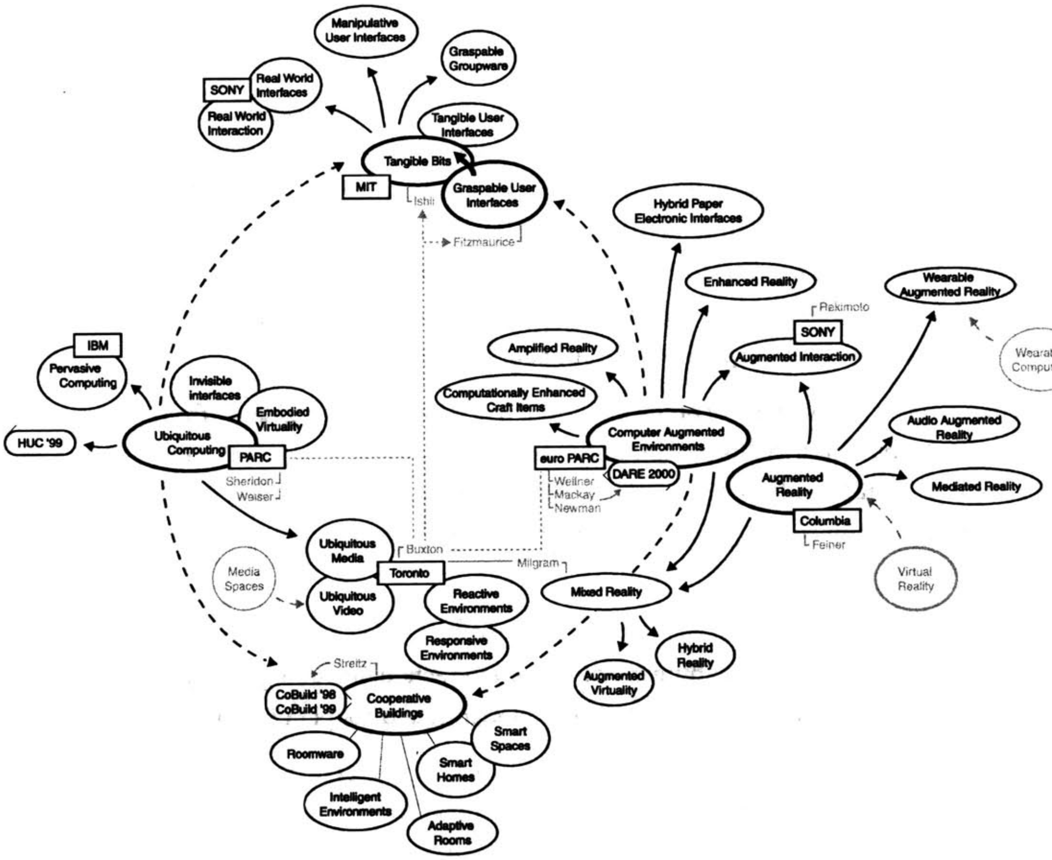


**Figure 3.1: Illustration of related user interface themes** *(from [Svendsen 2001, p. 13])*

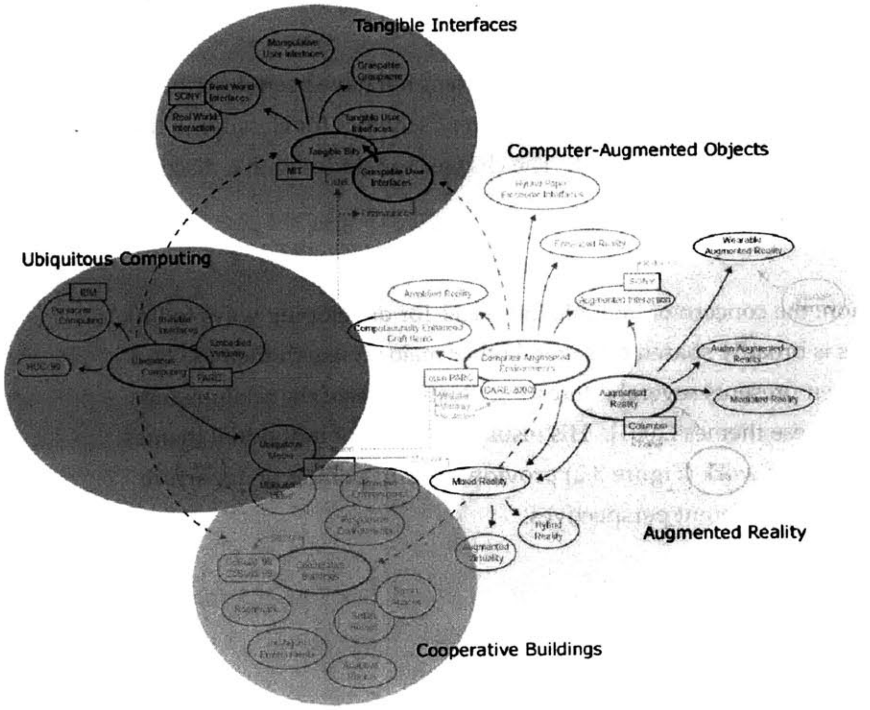


**Figure 3.2: Illustration of major related themes** *(from [Svendsen 2001, p. 21])*

Of course, it is difficult or impossible for such illustrations to be "comprehensive." Svendsen's figures draw primarily from work within the human-computer interaction community, and a number of pioneering systems, people, organizations, and early trends that will be discussed in the following pages also are not reflected in these figures. These include influential work from Bishop [1992], Oba [1990], and other product designers; efforts from the media art community, such as Naimark [1984] and Krueger [1985]; pioneering efforts such as by Perlman [1976], Aish [1979, 1984], Frazer [1983, 1995], and Anagnostou et al. [1989]; and the influence of organizations such as Interval Research (e.g., [Cohen et al. 1999, Singer et al. 1999]). However, Svendsen's illustrations are effective in communicating the identities and relationships among many of the major research themes from an HCI perspective that have developed over the last decade.

The themes highlighted in Figure 3.2 will be considered in the following paragraphs.

### 3.1.1 Ubiquitous computing

Among early visions for relating the physical and digital worlds, one of the most influential is Weiser's seminal paper introducing the area of ubiquitous computing [1991]. Weiser anticipated the migration of computing off of the desktop and into niche contexts within the physical environment. In recent years the term "pervasive computing" has also been used to describe a similar concept [Huang et al. 1999, Satyanarayanan 2001]. In many respects, Weiser's vision is already coming to pass, with the widespread integration of computing into handheld forms – as "personal digital assistants," high-end mobile telephones, digital music players, and many others – and the proliferation of computer projectors and wall-based applications as two particularly visible examples.

**Figure 3.3: Ubiquitous computing concepts: prototypes of tabs, pads, and boards [Weiser 1991]**

Research in ubiquitous and pervasive computing has often highlighted the "computing" component of these phrases. Many research efforts under these labels have focused on the development of new mobile devices, wireless networking, disconnected operation, and power management. The more interaction-oriented work under this label has frequently investigated issues of interaction with varying scales of visual real estate (from a few square centimeters to multiple square meters); with the software leveraging of contextual awareness and physically situated content; and with interaction between multiple colocated computing devices (e.g., handheld and wall computers) [Abowd and Mynatt 2001].

From a user interface, ubiquitous computing has most often followed the path of adapting graphical interface approaches to new physical devices and contexts. While this differs from the tangible interface approach, the more evolutionary trajectory gives it heightened practical relevance in the immediate term.

### *3.1.2 Augmented reality and computer-augmented environments*

One of the major themes in the history of human-computer interaction is the notion of "augmentation." The idea of augmenting people's thought processes through technological means was a central theme of Bush's profoundly influential article "As We May Think" [Bush 1945, Simpson et al. 1996], and remained an even more explicit agenda for the seminal work of Engelbart [1962].

As another interpretation of the "augmentation" term, the concept of augmented reality broadly relates to the fusion of physical world artifacts and computationally mediated augmentations. Sutherland's invention of the see-through head-mounted display [1968] is often identified as one of the first embodying instances. At present, the augmented reality term has several different interpretations. Many researchers consider augmented reality to be closely associated with the use of head-mounted displays as a path for virtually superimposing graphical objects and annotations over views of the surrounding physical world. Two early and influential examples of this approach are the KARMA system of Feiner et al. [1993, Figure 3.4a]; and the medical ultrasound work of Bajura, State, and others [Bajura et al. 1992; State et al. 1994, Figure 3.4b].

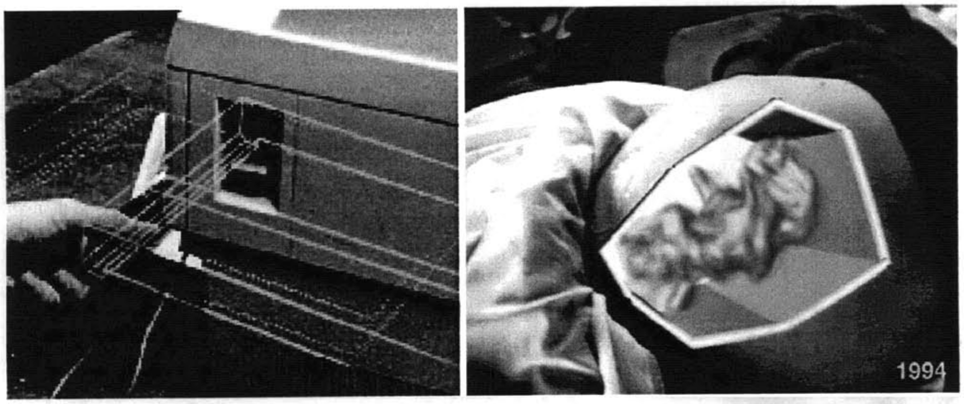


**Figure 3.4: KARMA, printer repair application** [Feiner et al. 1993]; **spatially-registered ultrasound imagery** [State et al. 1994]

Another approach to augmented reality seeks to projectively augment physical-world spaces through overhead projectors. One such stream works to augment pre-existing work practices in the physical world. Wellner's pioneering DigitalDesk supported augmented interaction with paper documents on a physical desktop, identifying and augmenting these with overhead cameras and projectors [1993, Figure 3.5a]. As another example, Mackay and Pagani applied these paper-based interaction techniques to systems for video storyboarding [1994, Figure 3.5a].

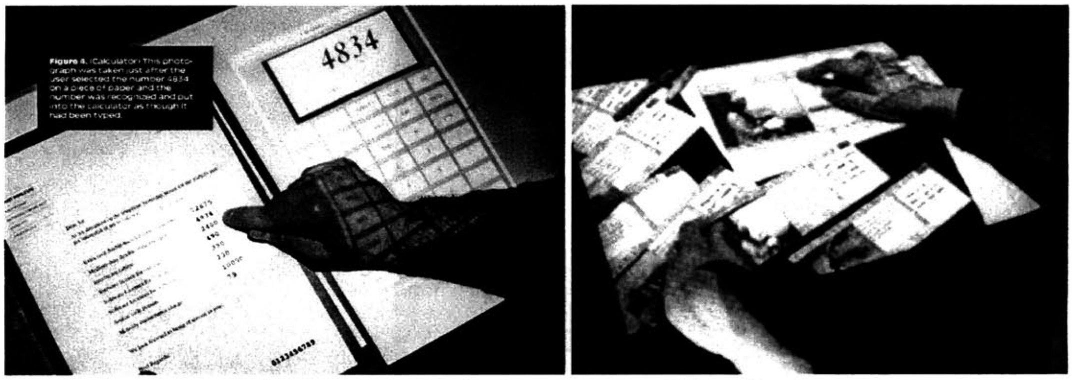


**Figure 3.5: DigitalDesk (calculator application) [Wellner 1993], Video Mosaic [Mackay and Pagani 1994]**

Another kind of projective approach seeks to allow surfaces of the physical environment to become extensions of traditional computing environments. Raskar et al.'s "Office of the Future" [1998, Figure 3.6a] presents one example, motivated largely from a technological perspective. Rekimoto and Saitoh's "Augmented Surfaces" research presented another compelling related example [1999, Figure 3.6b]. In this system, the GUI "drag and drop" facility of laptop computers was extended to allow graphical objects and applications to be seamlessly migrated between different interactive surfaces, as well as associated with physical objects.

**Figure 3.6: "Office of the Future" [Raskar et al. 1998]; Augmented Surfaces [Rekimoto and Saitoh 1999]**

Tangible augmented reality is a new stream of research that integrates some of the benefits of tangible interfaces and augmented reality systems [Kato et al. 2000]. In these systems, physical objects are used as tools and containers for physically interacting with AR-enhanced visuals (Figure 3.7). These approaches offer the benefits of stronger integration of visual and physical representations of digital information, at the cost of encumbrances posed by head-mounted displays (e.g., interference with eye gaze and other important cues in group settings).

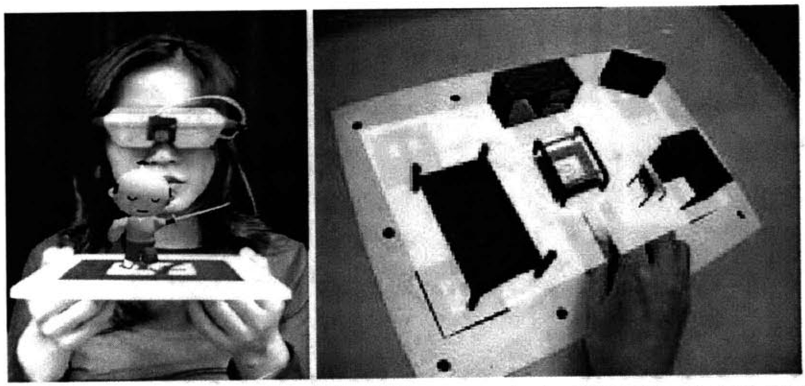


**Figure 3.7: Early examples of tangible augmented reality: use of card, paddle [Kato et al. 2000]**

### *3.1.3 Cooperative buildings*

The area of "cooperative buildings" provides another stream of related research. This work takes as a starting point the contexts of architectural space and co-located cooperative interaction. These contexts lead naturally to concerns about physical scale; integration of computational interfaces with walls, doors, furniture, and other elements of architectural space; and the nature of situated relationships and interactions within such spaces [Streitz et al. 2001]. Strong examples include the i-Land prototypes of Streitz et al. [1999, Figure 3.8]; the reactive environments of Cooperstock et al. [1995]; and interactive wall approaches such as by Guimbretiere et al. [2001]. A number of systems listed in the above sections as examples of augmented reality (e.g., [Raskar et al. 1998, Rekimoto and Saitoh 1999]) and ubiquitous computing can also be seen as relating to cooperative buildings.

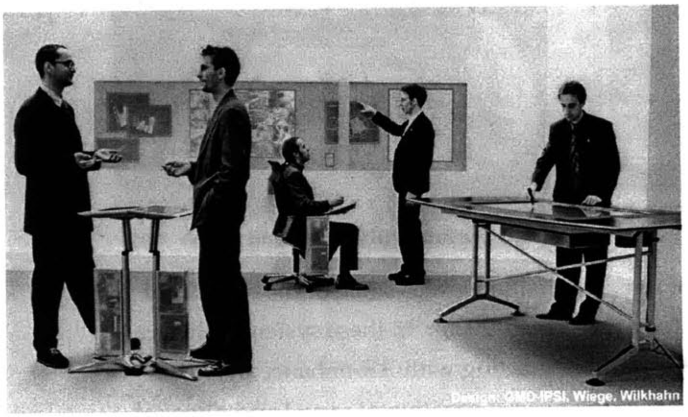


**Figure 3.8: i-Land prototypes [Streitz et al. 2001]**

## 3.2 Tangible interface instances

While the areas of ubiquitous computing, augmented reality, and cooperative buildings hold in common a concern for physically contextualized interaction, I believe they inhabit different conceptual and design spaces than tangible interfaces. In particular, where tangible interfaces are centrally concerned with the user interface properties of systems of representational physical artifacts, none of these alternate frameworks share this emphasis.

As introduced in §1.1, I believe that much of the current design space of tangible interfaces can be divided into three high-level approaches: interactive surfaces, constructive assemblies, and tokens+constraints. In the following pages, I present a literature survey that introduces and briefly discusses examples of each approach. While the thesis emphasis is upon systems utilizing tokens+constraints, a more inclusive review supports the thesis' consideration of the broader space of tangible interfaces.

## 3.3 Interactive surfaces

As discussed in the Introduction chapter, one popular paradigm for tangible interfaces is based upon "interactive surfaces," where physical objects are manipulated by users upon an augmented planar surface. The presence, identity, and configuration of these objects is then electronically tracked, computationally interpreted, and graphically mediated.

In the context of tangible interfaces, interactive surfaces have most frequently taken one of several major forms. Perhaps the most popular are "interactive workbenches," where objects are configured upon a near-horizontal workbench. A number of tangible interfaces have also been based upon "interactive walls." To counteract gravity, these systems often use magnets, sticky-notes, thumbtacks, or LEGO-like attachments to couple objects to the surface.

It should also be noted that the "interactive surface" term itself is not entirely specific to tangible interfaces. For example, a number of systems have developed interactive walls and whiteboards that use the human body or pen strokes as the primary medium of interaction. Examples include [Galloway and Rabinowitz 1980; Krueger 1984; Stefik et al. 1987; Weiser 1991; Pedersen et al. 1993; Ishii et al. 1994; Maes et al. 1996; Matsushita and Rekimoto 1997; Strickon and Paradiso 1998; Streitz et al. 1999; Mynatt et al. 1999; Guimbretiere et al. 2001]. For the most part, these works do not employ tangible interface techniques (with the work of [Ishii et al. 1994] as an important exception). Nonetheless, they may also broadly be considered as employing "interactive surfaces." References in the thesis to "interactive surfaces" will concern their use within tangible interfaces unless otherwise noted.

### *3.3.1 Interactive workbenches*

One of the earliest interactive workbench systems was the DigitalDesk of Wellner [1991]. This was discussed in §3.1.2 and pictured in Figure 3.5a. Another early system is the Bricks work of Fitzmaurice, Ishii, and Buxton [1995, Figure 3.10a]. The Bricks research, as a central example of the broader "graspable user interface" approach, involves placing one or more bricks – abstract physical blocks tracked with 6DOF (six degrees of freedom) – onto various screen-based virtual objects, b-spline control points, etc. Bricks can then be used to physically rotate, translate, or (using multiple bricks in combination) scale and deform the "attached" virtual entities by manipulating the proxying brick devices (Figure 3.9).

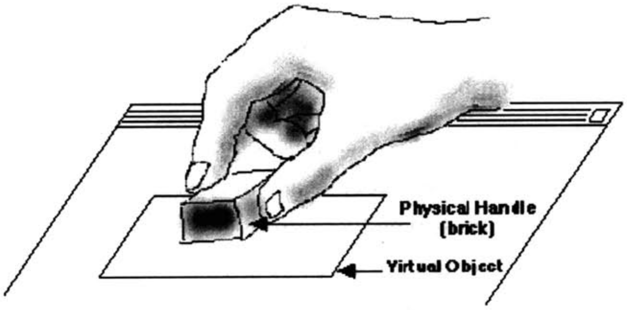


Figure 3.9: Bricks concept diagram [Fitzmaurice et al. 1995]

The Bricks system positions the role of its physical tokens as "handles" used for "grasping" graphical objects that remain within the graphical surface of its interactive workbench. Here, the physical tokens are disassociated from the "brick" whenever the brick is lifted from the workbench. This differs from the notion of TUI tangibles as persistent physical representations of digital information.

In addition, the Bricks "GraspDraw" prototype also illustrates a second behavior which is much closer to that of tangible interfaces. In particular, the "GraspDraw" drawing system introduces physical "tray" and "inkwell" devices (Figure 3.10b) that are used to bind tools and attributes (colors) to Bricks. These bindings persist until the tokens are explicitly rebound. However, the bindings are not active upon the workbench unless a button is pressed; the normal Brick behavior is as a handle for graphical objects.

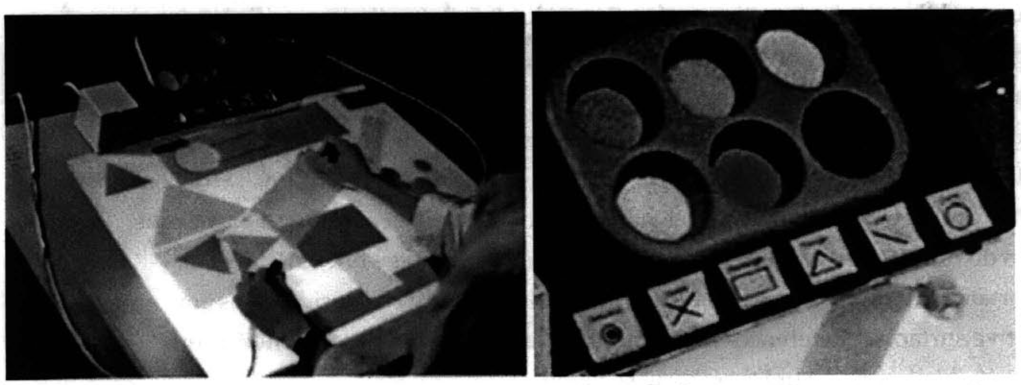


Figure 3.10: Bricks – GraspDraw prototype and tray+inkwell close-up [Fitzmaurice et al. 1995]

Fitzmaurice did not elaborate upon the tray and inkwell devices, and the different behaviors are described as different styles of binding (transitory and persistent). From the perspective of this thesis, the persistent bindings approximate a kind of "container" functionality (albeit for functional rather than data bindings). Moreover, the tray and inkwell each illustrate kinds of interpretive constraints, albeit without the "manipulate" phase (discussed in §1.1.2). The tray and inkwell were loosely inspirational to the thesis' further development of constraints.

The subject of my master's thesis, the metaDESK, was the first interactive workbench fleshing out the tangible interface approach [Ullmer and Ishii 1997, Ishii and Ullmer 1997, Ullmer 1997]. The main demonstration application, Tangible Geospace, supported interaction with a geographical space through the manipulation of several physical tokens (described as "physical icons" or "phicons") and supporting tools. These are illustrated in Figure 3.11. The building phicons were explicitly described as "containers for digital information." However, the highly representational tokens were more or less permanently bound to their geographical "contents," with little discussion of mechanisms for rebinding.

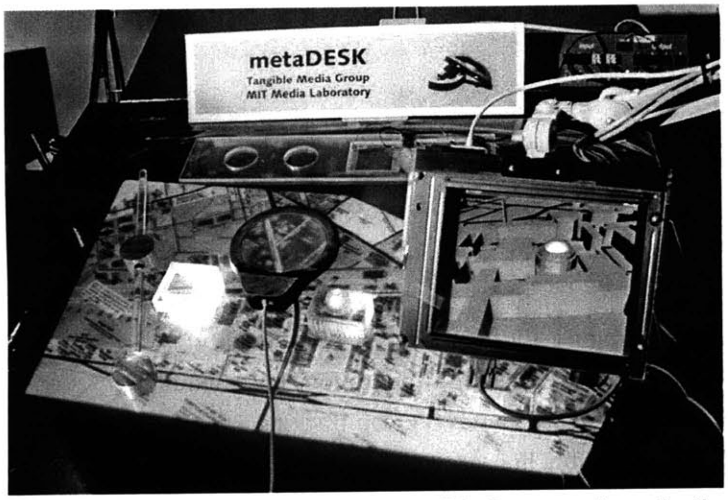


**Figure 3.11: The metaDESK, with "Tangible Geospace" application**

The building phicons were associated with multiple sets of digital information, representing alternative two- and three-dimensional representations of the associated geographical location. I did not explicitly discuss the manipulation of information aggregates within the Tangible Geospace application. However, an earlier application called "Tangible Infoscapes" did explore the physical representation and manipulation of information aggregates [Ullmer 1997].

In this prototype, business card-sized elements called "hypercards" represents collections of digital images (Figure 3.13). Two kinds of image collections were explored: sequences of key-frames from videos; and sequences of Japanese art images known as "renga." Two dimensional "digital shadows" of the hypercards were displayed on the metaDESK surface, with three dimensional "digital shadows" visible through the arm-mounted "active lens." An early prototype is shown in Figure 3.12. However, a meaningful application or mechanism for interacting with the hypercard contents was never fully developed.

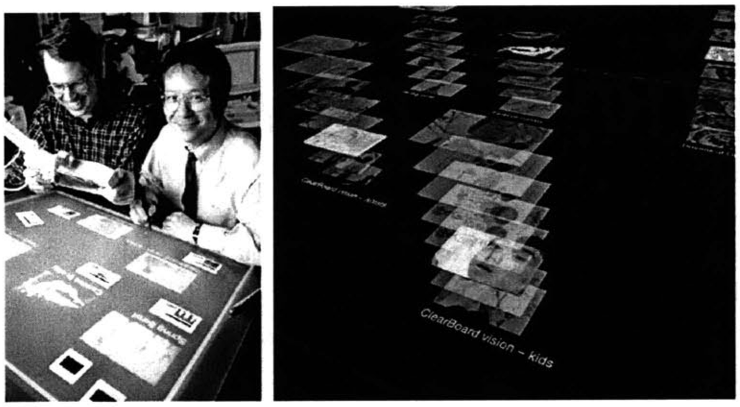


Figure 3.12: "Tangible Infoscapes:" early concepts for representing information aggregates


The Media Laboratory Tangible Media Group

SEAMLESS MEDIA
ClearBoard vision - kids
Hiroshi Ishii & Minoru Kobayashi
http://

Figure 3.13: "Hypercard" object used within Tangible Infoscapes

Another workbench approach from the same timeframe is illustrated by the "Real Reality" work of Schäfer, Brauer, and Bruns [1997]. This research developed applications for assembly line planning and other industrial contexts. The system used a novel grasp-tracking approach to manipulate both literal physical models of the assembly line, as well as physical representations of logical flows (Figure 3.14).

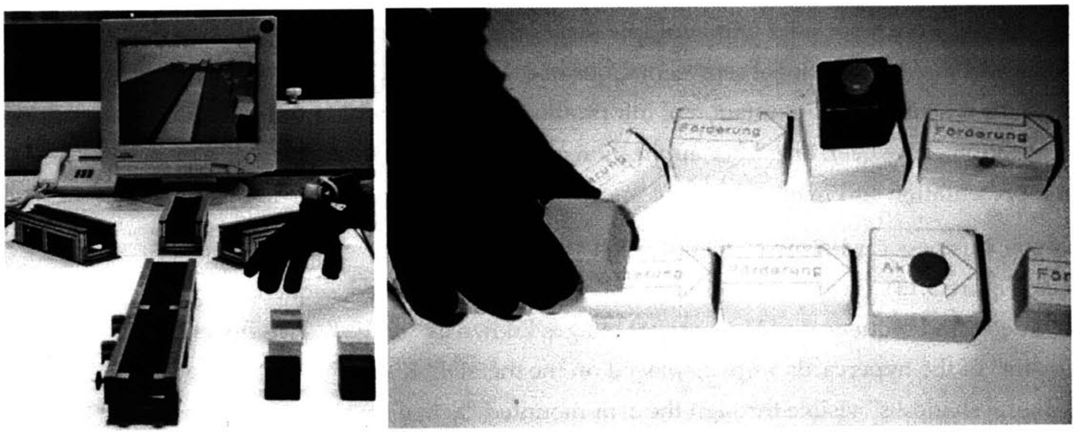


Figure 3.14: "Real reality" interface for assembly line planning [Schäfer et al. 1997]

The BUILD-IT system of Fjeld et al. [1998, 2002] developed another interactive workbench that made several new contributions. First, the system combined 2D and 3D views in a fashion better suited to collaborative work than the metaDESK's active lens (described as "above" and "side" views, respectively, in Figure 3.15b). Second, it developed a workbench layout facilitating the binding of brick elements to data and functional elements (Figure 3.15b). Third, the system developed several more realistic example applications (including floor planning and assembly line planning), and tested these with domain experts.

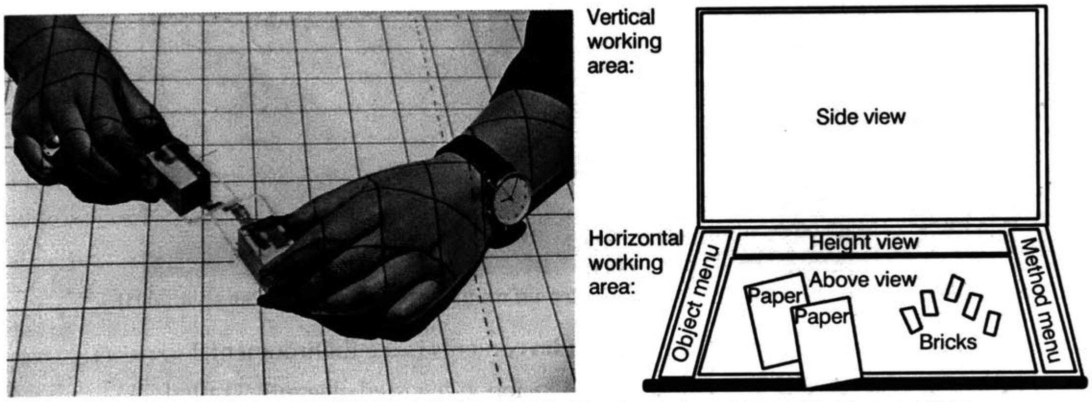


**Figure 3.15: BUILD-IT system (floor planning application); system layout [Fjeld et al. 1998]**

One of the most fully developed interactive workbench systems, and arguably the first system to achieve a "critical mass" of functionality, is the Urp urban planning system of Underkoffler [1999a]. The system supported shadow studies, reflection studies, wind simulations, zoning metrics, and other features useful for understanding implications of constructing a new building (Figure 3.16a). The system also saw several semesters of use within urban planning classrooms, providing valuable feedback on group usage contexts (Figure 3.16b).

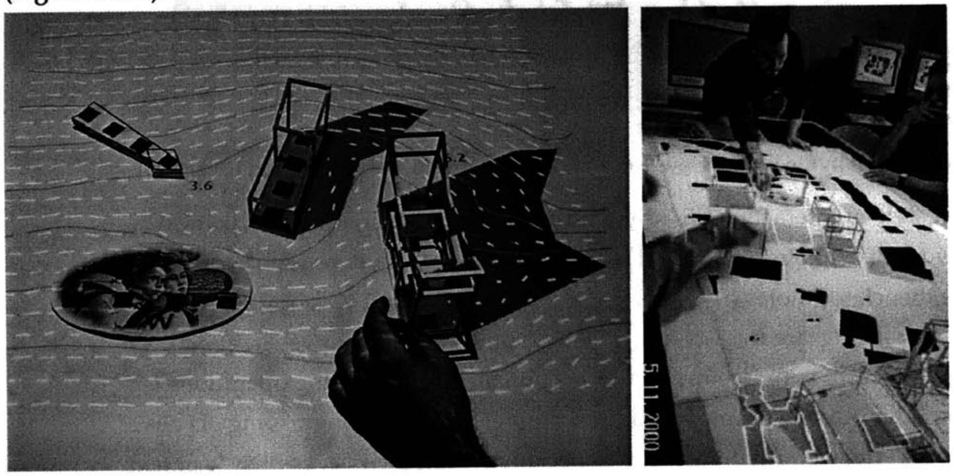


**Figure 3.16a, b: "Urp" urban planning simulator [Underkoffler and Ishii 1999]**

Interactive workbench systems have also been applied to more abstract problem domains for which inherently geometrical representations do not exist. For example, the Sensetable work of Patten et al. has developed systems for interacting with supply chain simulations, musical performances, and other application areas [2001, 2002; Figure 3.17]. This work has also made important advances in developing robust, high-performance sensing infrastructure based on electromagnetic tracking technologies.

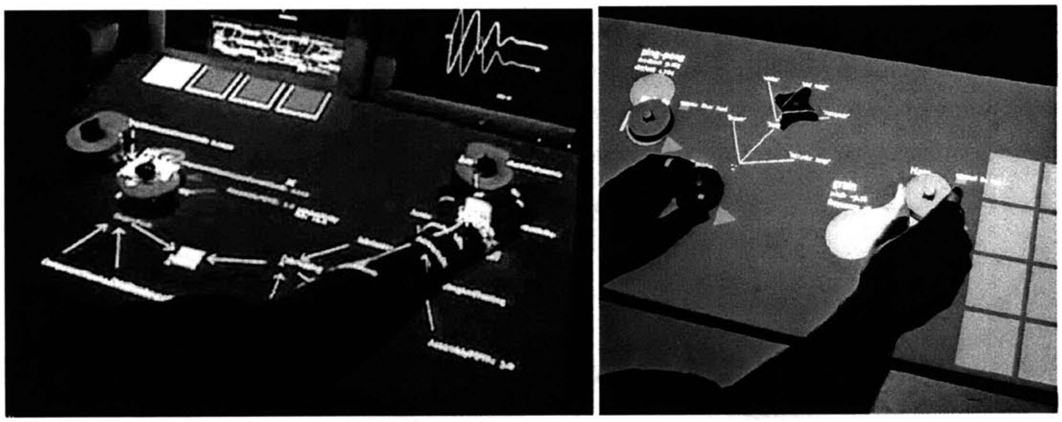


Figure 3.17: Sensetable platform: system dynamics, electronic music applications [Patten et al. 2001]

A much earlier tangible interface for interacting with audio contents is the Environmental Audio concept design of Oba [1990]. Oba's mockup and vision video used elegantly crafted wooden, metal, and plastic tokens as containers for ambient sounds from nature, urban spaces, and the electronic airwaves, respectively (Figure 3.18). This work remains one of the most provocative TUI examples for the use of physical materials and forms to evoke digital contents.

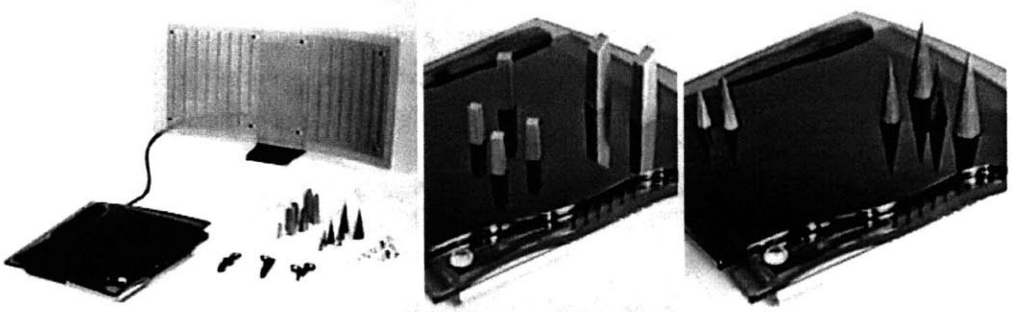


Figure 3.18: Environment audio system overview + containers for urban, forest sounds [Oba 1990]

Another quite different style of interaction broadly within the interactive workbench context is the Surface Drawing work of Schkolne et al. [2001]. This builds upon the stereo shutterglass-based Responsive Workbench technology [Krüger and Fröhlich 1994], and was originally developed as a purely gestural interface for "sculpting" 3D graphical forms. However, after finding it cumbersome to control both freeform sculpting and structured editing operations with the same glove-based input, the developers added physical "tongs" (for grasping 3D objects), "erasers," "magnets" (for deformations), etc. Especially within their "eyes-busy" context of interaction, they described these physical tools as much more effective than their earlier gestural approach, building in part on leveraging kinesthetic memory of tool locations.

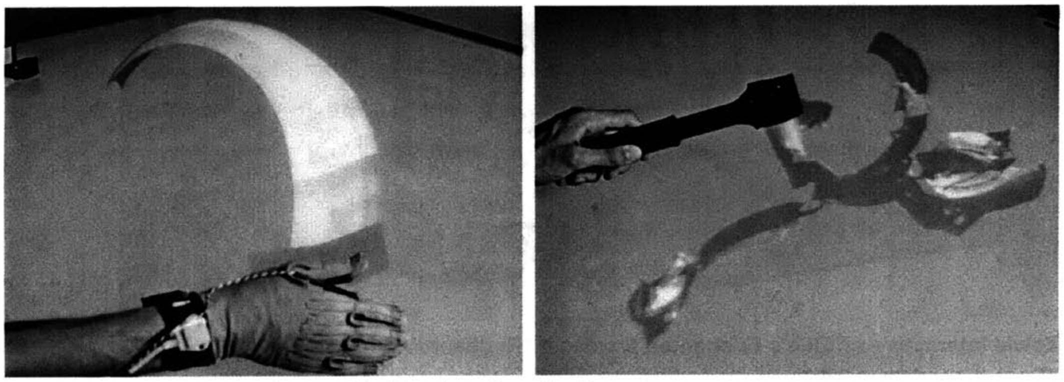


**Figure 3.19: Gloved hand, tongs in "Surface Drawing" system [Schkolne et al. 2001]**

While many interactive workbenches have used front- or back-projected graphics onto the horizontal work surface, several systems have used computer display screens as kinds of "magic mirrors." The Real Reality work of Schäfer et al. [1997, Figure 3.14] has provided one example. An earlier example is the Virtual Lego work of Small [1996, 1999]. Some of Small's systems implemented symmetric screen views of the workspace, with dynamic graphics providing additional augmenting information (e.g., the gear ratios visible within Figure 3.20a). As another example, physical manipulation of a LEGO helicopter allowed the navigation of a complex spatial scene, as well as dynamic spatial selection and application of material properties. Small also implemented a series of asymmetrical systems, where (e.g.) physical dials and tokens could be used to navigate large bodies of text (Figure 3.20b).

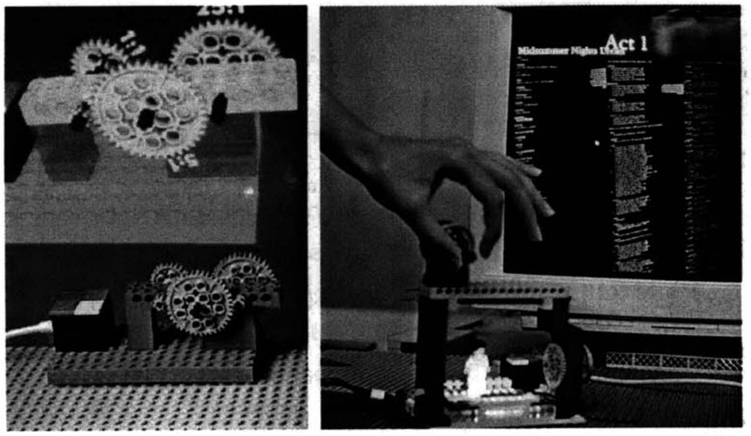


**Figure 3.20: Virtual Lego: symmetric representations + asymmetric navigation of text [Small 1996]**

A somewhat related approach was developed by the commercial products of Zowie Interactive. Zowie marketed two different playsets that used physical tokens to represent characters and artifacts (Figure 3.21). The placement and reconfiguration of these tokens within the playset was used to navigate and interact with various scenarios that were animated upon the screen.

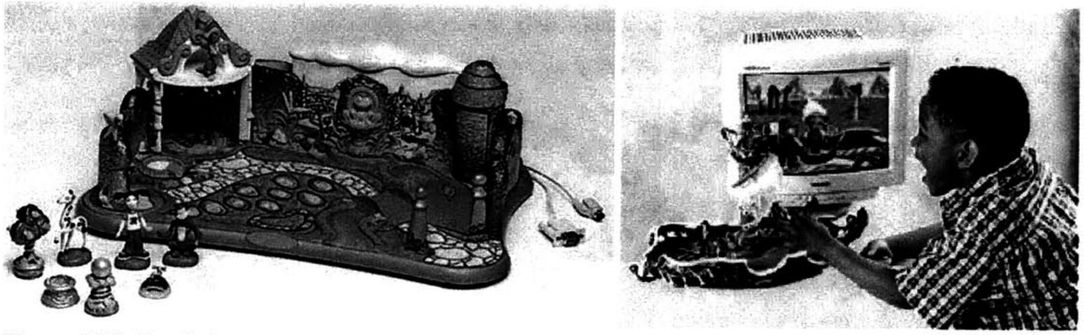


Figure 3.21: Zowie Interactive's "Ellie's Enchanted Garden," "Redbeard's Pirate Quest" products [Francetic and Shwe 2000]

Another early example of screen-based interaction complemented by TUI tangibles on a horizontal worksurface is Wacom tablet-based "character devices" of Fukuzaki [1993, cited in Fitzmaurice 1996]. This system used physical tokens with representational shapes and predefined functions to invoke and control operations such as erasing, color selection (invoked by an "ink pot"), and file storage (represented by an iconic "file cabinet") (Figure 3.22).

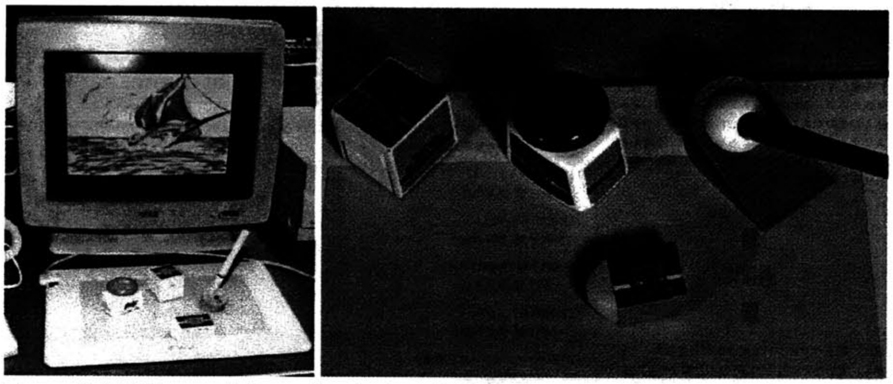


Figure 3.22: Wacom "character devices" [Fukuzaki 1993, Fitzmaurice 1996]

Finally, the influential neurosurgical props of Hinckley et al. [1994] made use of the computer screen as a semi-symmetrical display. Here, a physical doll's-head "prop" was used to orient and scale neurosurgical brain visualization, while cutting plane and trajectory props were manipulated with the second hand to operate upon brain data (Figure 3.23). However, this system did not make use a horizontal worksurface, and so may belong in a separate category including work such as the "surface drawing" system [Schkolnet et al. 2001].

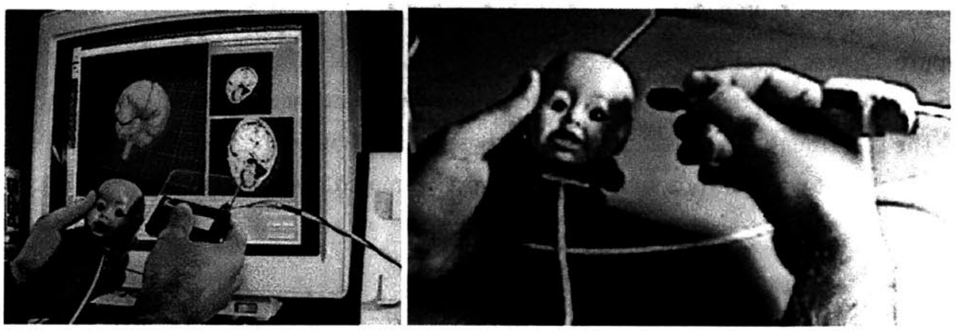


Figure 3.23: Passive props for neurosurgical visualization (cutting plane, probe) [Hinckley et al. 1994]

### 3.3.2 *Interactive walls*

Another major variation of interactive surfaces is the use of interactive walls as tangible interfaces. This approach finds roots in early interactive art such as Galloway and Rabinowitz's "Hole in Space" [1980] and Krueger's artificial realities [1984], as well as electronic whiteboards descending from [Stefik et al. 1987].

One of the earliest uses of physical objects as interface elements upon interactive walls was the use of sticky-notes (e.g., Post-It™ notes) in Ishii et al.'s ClearBoard [1994]. While the contents of these paper notes were not computationally interpreted, they played important interface roles in combination with pen strokes and eye gaze (Figure 3.24).

**Figure 3.24: Use of sticky-notes for problem solving with the ClearBoard [Ishii et al. 1994]**

Another early tangible interface employing interactive wall surfaces was the transBOARD [Ishii and Ullmer 1997]. This system combined a digital whiteboard with magnetically backed, business card-sized tokens called "hypercards." Each hypercard was encoded with a barcode, which could be scanned to copy the whiteboard's current contents "into" the card. These contents could later be accessed over the web, or potentially over the metaDESK (which used the same hypercard objects).

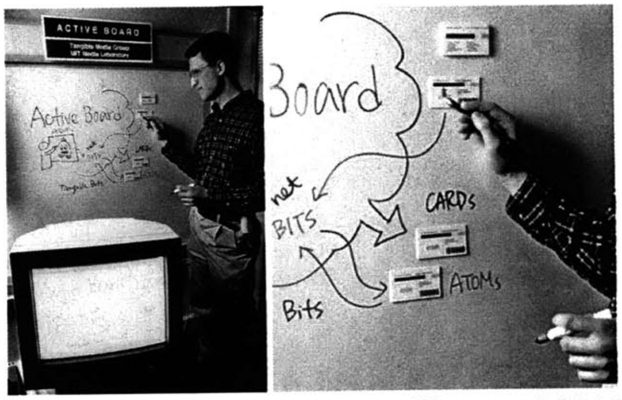
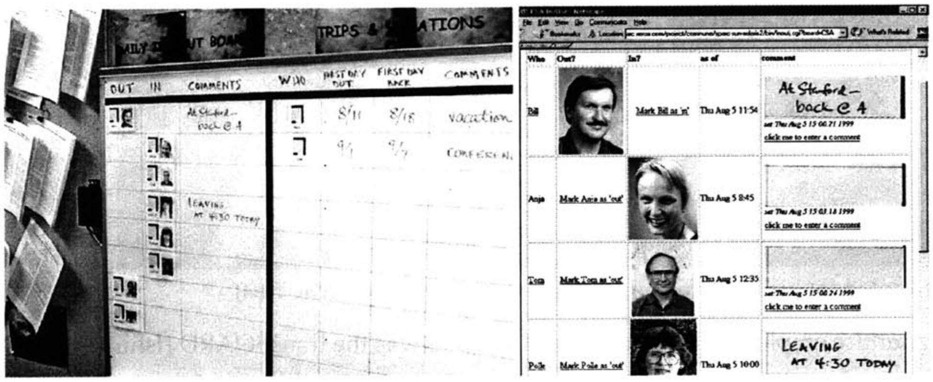


**Figure 3.25: transBOARD system and "hypercards" [Ishii and Ullmer 1997]**

The Collaborage system of Moran et al. [1999] pushed the transBOARD's broad approach much further. As one embodying example, a whiteboard-based "in/out" board used magnetically backed, glyph-tagged cards to represent people. The placement of these tags was monitored with a computer vision system, and the corresponding textual annotations were scanned and entered into a web-based "in/out" monitor (Figure 3.26). Like the transBOARD, the Collaborage board did not employ projection. Speaking of this design decision, the authors made the interesting observation:

> We also experimented with a video projector on the board, which has the advantage of very flexibly displaying information in a variety of ways. A projector might be appropriate in a limited work session; but we found it to be much too "intense" for a persistent wall, turning the wall into from a background information source to a center of attention.

Ishii and I have noticed similar issues in the different "quality of light" offered by projectors as opposed to (say) large plasma display screens. These observations bring to mind McLuhan's descriptions of interplay between "the medium and the message" [McLuhan 1964].



| Who | Out? | In? | as of | comment |
|---|---|---|---|---|
| Bill |  | Mark Bill as 'in' | Thu Aug 5 11:54 | At Stanford – back @ 4<br>click me to enter a comment |
| Anja | Mark Anja as 'out' |  | Thu Aug 5 8:45 | click me to enter a comment |
| Tom | Mark Tom as 'out' |  | Thu Aug 5 12:35 | click me to enter a comment |
| Polle | Mark Polle as 'out' |  | Thu Aug 5 10:00 | LEAVING AT 4:30 TODAY |

**Figure 3.26: Collaborage [Moran et al. 1999]**

Several more recent tangible interfaces have built upon other mediated uses of sticky-notes. In the Rasa system of McGee and Cohen [2001], handwritten sticky-note annotations are captured by a digitizer tablet, and the placement and contents of these sticky-notes are interpreted in combination with voice commands (Figure 3.27).

**Figure 3.27: Rasa [McGee and Cohen 2001]**

Interaction with the transBOARD, Collaborage, and Rasa tokens is augmented with audio feedback, and not directly mediated with a projective video display. In contrast, the Designer's Outpost of Klemmer et al. [2001, 2002] does make use of direct graphical mediation.

This system supports the usage of computationally-mediated sticky-notes within a web site planning task. Paper sticky-notes are used upon a rear-projection display (Figure 3.28). The notes are tracked by a rear vision system (building upon the metaDESK technique of [Ullmer and Ishii 1997]), and sticky-note contents are captured by a front-facing foveating camera.

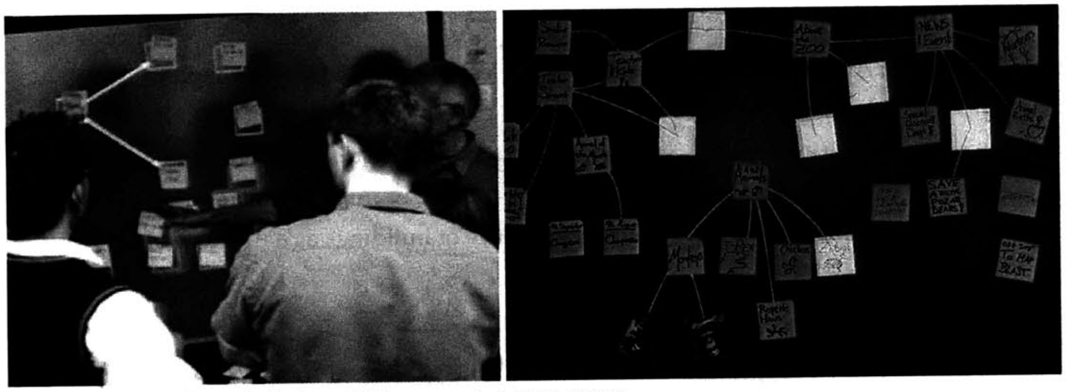


**Figure 3.28: Designer's Outpost [Klemmer et al. 2001, 2002]**

Another projectively augmented interactive wall is the Senseboard of Jacob et al. [2002]. They observed that in some relatively ordinary work contexts (e.g., conference session planning), the best of current GUI tools are passed over for simpler physical supports – even though GUIs provide much more extended functionality, and even when the task content is "born digital." The Senseboard TUI uses magnetically-backed tokens to support a variety of planning and organizing tasks such as the scheduling of conference presentations (Figure 3.29). Jacob et al. demonstrate its time-performance to be slightly better than comparable graphical and paper variations for a simplified organizational task. Equally important, their work supports the intuition that TUIs might find use in contexts where even the best GUIs go unused.

Like the "in/out" board of Collaborage, Senseboard's tokens are used within a grid layout, within which special ranges of cells can take on specific computational interpretations. In this respect, Senseboard and Collaborage illustrate a kind of hybrid between the interactive surface and token+constraint approaches, albeit without some of the mechanical enforcements present within the interfaces of this thesis.

**Figure 3.29: Senseboard system, close-up of grouping operation [Jacob et al. 2002]**

## 3.4 Constructive assemblies

Another major approach for tangible interfaces draws inspiration from building blocks and LEGO™. This approach has been employed by some of the earliest tangible interfaces, often toward the ends of providing modular, electronically instrumented artifacts for constructing models of physical-world systems.

Beginning in the late 1970's, Aish implemented a "building block system" (BBS) for modeling physical-world buildings [Aish 1979, Aish and Noakes 1984]. One interesting aspect of these systems is that they were used to explore not only the geometric structure of buildings, but also some of the more abstract resulting properties of such spaces. As one example, Aish and Noake's system interactively illustrated the thermal performance of buildings as a function of the changing geometry (Figure 3.30b).

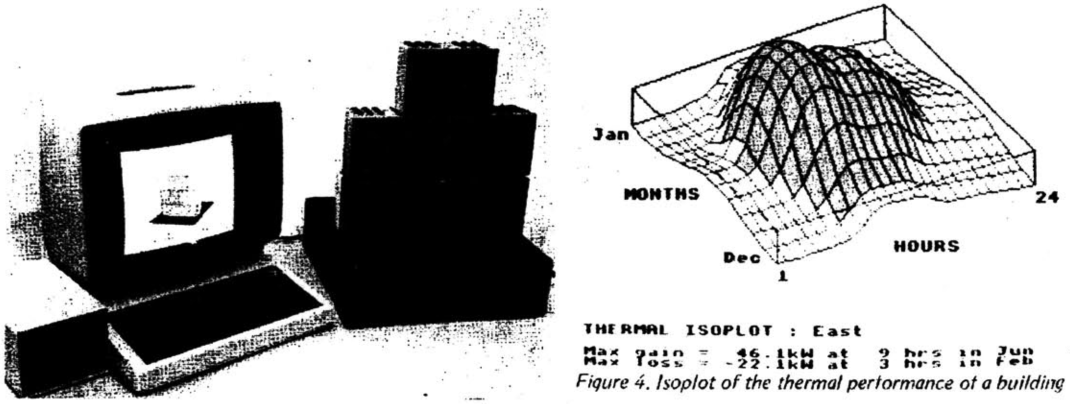


**Figure 3.30a,b: "Building blocks system" and example output (thermal isoplot) [Aish 1984]**

Beginning in the same timeframe, Frazer and his team built a wide variety of "intelligent modeling" systems for representing both physical buildings and also more abstract systems [Frazer 1982, 1994]. As one example, Frazer's Universal Constructor, a large system of modular interconnecting electronic cubes, supported three-dimensional constructions of physically reconfigurable cellular automata (Figure 3.31).

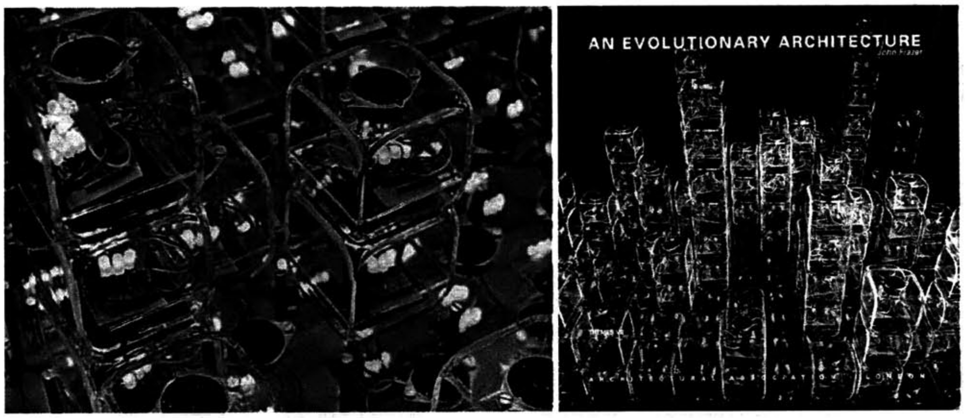


**Figure 3.31: "Intelligent modeling systems" [Frazer 1982, 1995]**

Several more recent tangible interfaces have also explored the encoding of cellular automata in modular physical elements. These include the Stackables of Kramer and Minar [1997]; the Tiles of Kramer [1998]; and Heaton's Peano [2000]. Each of these interfaces developed additional novel features, such as the Stackables' concept of a distributed display; the Tiles' use of mobile code; and Peano's conception as a touch-sensitive, painterly medium (Figure 3.32).

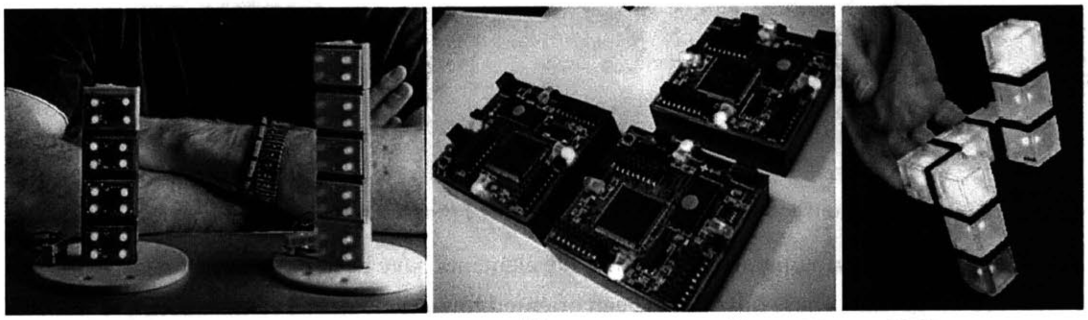


**Figure 3.32: Stackables [Kramer and Minar 1997]; Tiles [Kramer 1998]; Peano [Heaton 2000]**

Building from the mechanical engineering domain, the "geometry-defining processors" (or "GDP") of Anagnostou et al. supported the interactive physical construction of 3D fluid flow simulations [Anagnostou 1989, Figure 3.33]. A system of magnetically interlocking cubes, GDP was also intended to physically express the topology of an underlying parallel processing computation.

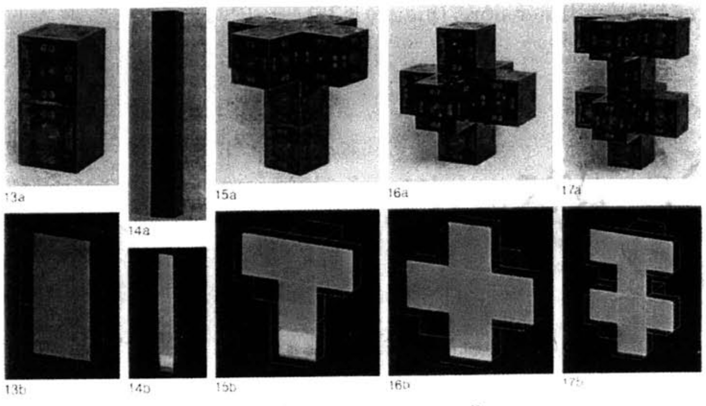


**Figure 3.33: "Geometry defining processors" [Anagnostou et al. 1989]**

Another related tangible interface is the Blocks system of Anderson et al. [2000]. This system used a series of blocks to physically describe different geometric structures (Figure 3.34a). These constructions were heuristically interpreted by the associated software to create graphical interpretations (e.g., Figure 3.34b) that could then be explored in various ways. The system also provided ways to scan and statically + dynamically interpret models constructed in clay, allowing another dimension of expression.

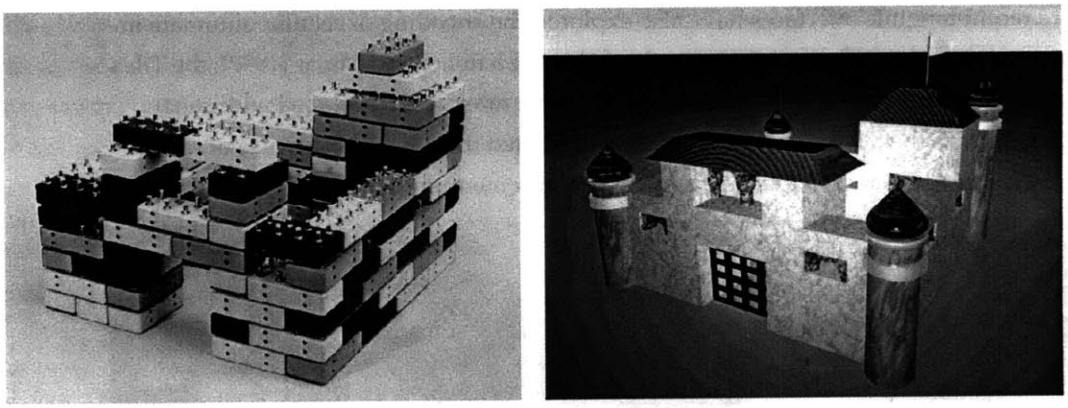


Figure 3.34: Blocks ("tangible interface + graphical interpretation") [Anderson et al. 2000]

Several tangible interfaces utilizing modular, constructive elements have been developed for describing software programs. These have often been oriented toward elementary education, offering concrete representations for abstract concepts that children may not be developmentally prepared to address in textual form.

One such design proposal was the "solid programming" concept of Maeda and McGee [1993]. This proposal articulated an approach by which a series of blocks and tags could be used to physically describe the fuzzy-logic response of a robotic device to a variety of input conditions. The design mapped 'AND,' 'OR,' and chained inferences to the Y, Z, and X dimensions, respectively, making effective use of all three spatial dimensions (Figure 3.35).

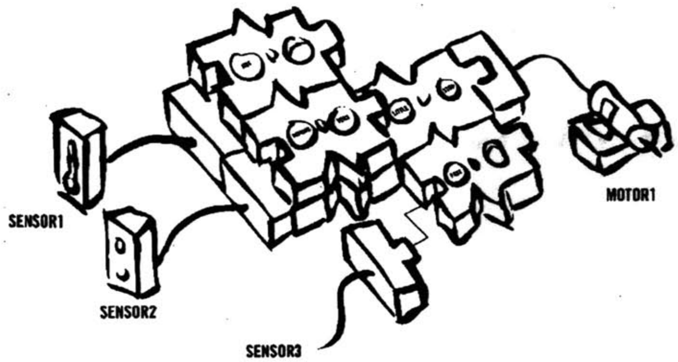


Figure 3.35: "Solid programming" design for mechatronic control [Maeda and McGee 1993]

A related system is AlgoBlock, a system of cubical aluminum blocks that dock with each other on a table [Suzuki and Kato 1993]. AlgoBlock was used to physically express a LOGO-like language. Each AlgoBlock represented a command, and offered control of associated parameters through knobs and levers permanently embedded within each block (Figure 3.36). AlgoBlocks also contained lighted buttons to trigger the execution of each physically embodied command. This execution would propagate onwards to other connected blocks, with the lights glowing to indicate the program execution's progression and evolving state.

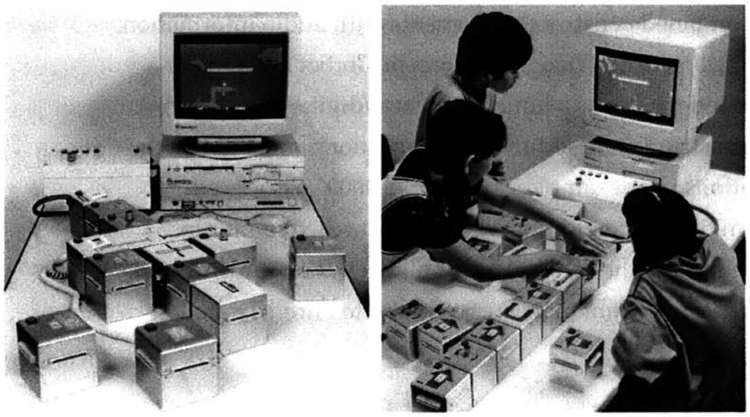


**Figure 3.36: AlgoBlocks [Suzuki and Kato 1995]**

Another tangible interface for expressing programs is the Programming Bricks system of McNerney [2000]. This interface used stackable, electronically active, parameterized LEGO bricks to express software rules in a simple functional language (Figure 3.37).

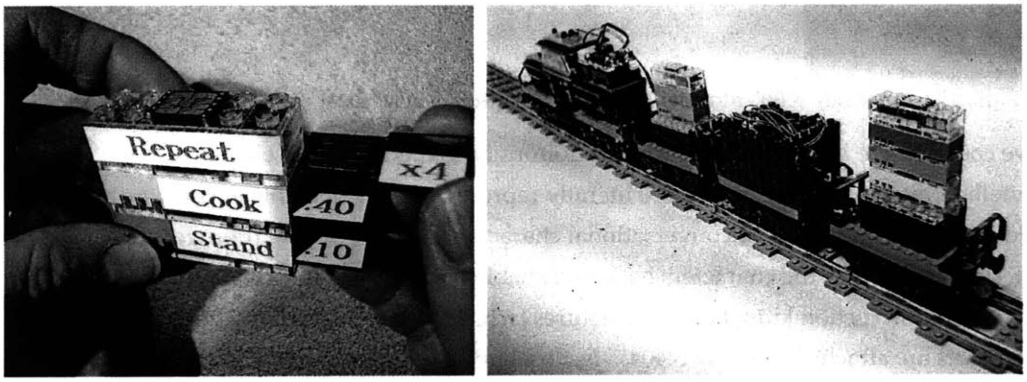


**Figure 3.37: Tangible programming bricks; a train carrying "tangible rules" [McNerney 2000]**

The Active Cubes of Kitamura et al. offer somewhat more abstract support for programming [2001]. Each cube is uniquely identified and can be bound to digital behaviors, with the cube's aggregate 3D configuration sensed in realtime (Figure 3.38). Active Cubes are embedded with a variety of sensors and actuators, allowing for the construction of modular, responsive structures.

**Figure 3.38: Active Cubes [Kitamura et al. 2001]**

A number of earlier tangible interfaces combined constructive elements with audio information, often in the context of storytelling applications. The Triangles system of Gorbet and Orth is a system of triangular electronic tiles intended for use as a kind of physical/digital interaction toolkit [1998a]. Triangles were used to construct several different applications, most of which related to audio and storytelling applications. For example, "Cinderella 2000" associated Triangles with characters and places from the Cinderella story in a kind of reactive "audio comic book" (Figure 3.39). Another application, the "Digital Veil," allowed the capturing and navigation of audio recordings in constellations of uniquely surfaced and textured Triangles.

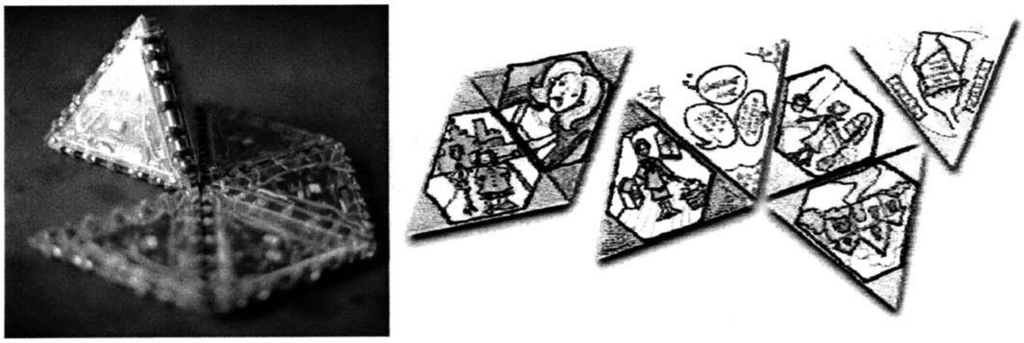


**Figure 3.39: Triangles, "Cinderella 2000" storytelling prototype** [Gorbet et al. 1998]

Where the above constructive interfaces have utilized uniformly shaped modular elements, a number of storytelling interfaces have used more literally representational elements to construct physical/digital representations of conversational characters. In LEGOHead [Borovoy 1995] and SAGE [Umaschi 1997], the characters have detachable body parts and clothing which act as "computational construction kit(s) to build creatures [which] behave differently depending on how these parts are attached" (Figure 3.40a). In Rosebud [Glos 1997], electronically instrumented stuffed animals are used as interactive containers for narratives by their owners. The TellTale system uses the modular segments of a "caterpillar" construction to capture, structure, rearrange, and replay segments of an audio story [Ananny 2001, Figure 3.40b].

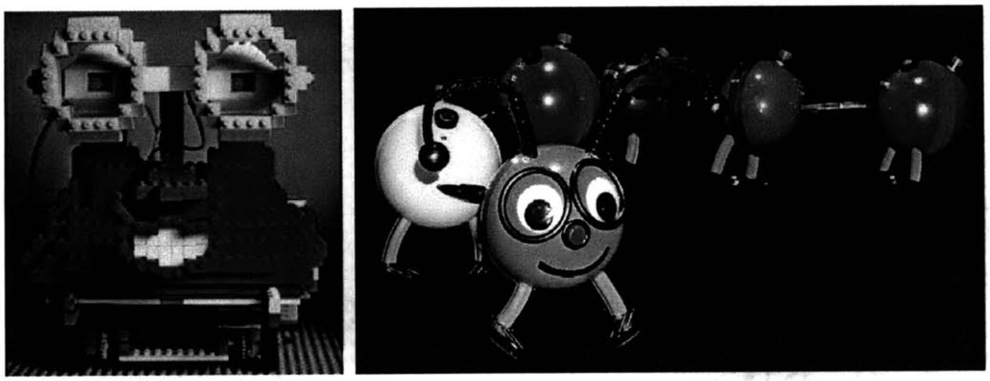


**Figure 3.40: LEGOHead** [Borovoy 1995]; **TellTale** [Ananny 2001]

A very different style of "constructive assembly" approach was illustrated by the TouchCounters work of Yarin and Ishii [1999, Yarin 1999]. Driven by a vision of "distributed visualizations," Yarin used low resolution, high brightness displays upon individual storage boxes to participate as fragments of a larger environmental display (Figure 3.41). One major difference of this work from other constructive assemblies in that the adding and removing of elements was a natural product of box usage, and did not serve a TUI interaction role per se. However, the concept of distributed visualizations – also illustrated by the Stackables [Kramer and Minar 1997, Figure 3.32], Tiles [Kramer 1998], Peano [Heaton 2000], and the Universal Contructor [Frazer 1995, Figure 3.31] – shares ground and suggests promising extensions to the constructive assembly approach. The TouchCounters system also has strong resonances with the thesis' query interfaces approach, a connection that is discussed in §5.3.5.

**Figure 3.41: TouchCounters [Yarin and Ishii 1999]**

A final system, the Navigational Blocks of Camarata et al. [2002], is closely related to the tangible query interfaces of this thesis. The system uses physical cubes to represent major categories of a historical application. Each face of these cubes is bound to a different instance of this category, using the FlipBrick/ToolStone concepts of [Fitzmaurice 1996] and [Rekimoto and Sciammerella 2000]. Placing a block on an active surface retrieves records relating to the cube's face element (Figure 3.42). Translating the block scrolls through multiple elements. Placing two blocks on the surface expresses a Boolean 'AND' combination of their contents. The interaction most consistent with constructive techniques involves bringing two blocks together. "Related" blocks physically attract, while "unrelated" blocks physically repel.

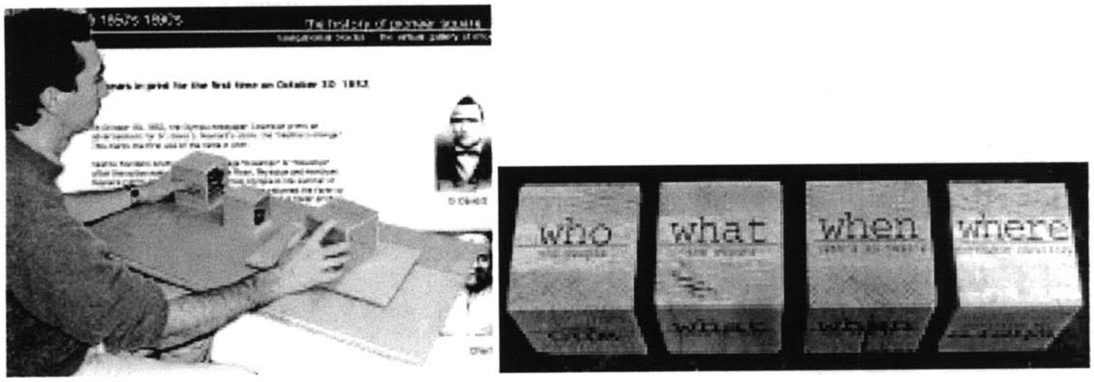


**Figure 3.42: Navigational Blocks [Camarata et al. 2002]**

## 3.5 Tokens+constraints

As introduced in §1.1.2, the systems of this thesis focus upon a TUI approach that I describe as "tokens+constraints" (or "physically constrained tokens"). While this approach has fewer embodying examples than the "interactive surface" or "constructive assembly" approaches, it does include systems created prior to the work of this thesis.

As discussed in §1.7, the thesis contributions in this area include:

- Identification and articulation of both tangible interfaces and the tokens+constraints approach;
- Creation of among the first tangible interfaces to employ both the "associate" and "manipulate" phases of interaction, as introduced within §1.1.2, and the most aggressive applications of the "manipulate" phase to date; and
- Creation of the first tangible interfaces to support the physical manipulation of aggregates of digital information. In so doing, the thesis suggests how tangible interfaces can be applied to a much larger space of use.

Perhaps the earliest example of the tokens+constraints approach, and one of the earliest tangible interfaces currently known, is the Slot Machine of Perlman [1976]. Its sister interface and predecessor, the "Button Box," was cited by Smith as one of the inspirations for the GUI "icons" context [Smith 1976].

The slot machine provided an interface for controlling Logo's robotic and screen-based "Turtle." In this interface, sequences of physical "action," "number," "variable," and "conditional" cards were configured within horizontal slots to construct Logo programs. Multiple cards could be stacked upon one another to create composite commands. E.g., the number card for "4" could be stacked upon the "move forward" action card to express "move forward 4." A height-based hierarchy existed between the different card types, allowing all of the cards with individual stacks to remain visible (Figure 3.43). The Slot Machine provided a fairly sophisticated level of programmatic control, including support for concepts such as recursion that have not been repeated in other known tangible interfaces to date.

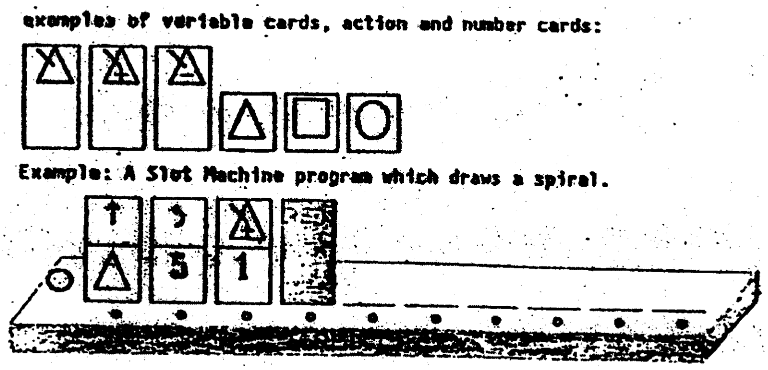


Figure 3.43: "Slot machine," recursive programming example [Perlman 1976]

The Slot Machine was based upon a series of several long rectangular "rows," which represent procedures composed of a series of card-based commands. Each row was rendered in a different color, and was divided into a series of slots into which cards could be placed. For instance, Figure 3.43 illustrates the "red row," in which four slots are occupied. The program of Figure 3.43 uses seven cards to express a recursive program for drawing a spiral:

1. Go forward for "triangle" time-units *("triangle" is a variable).*
2. Rotate left for 5 time-units.
3. Increment the "triangle" variable by 1.
4. Evaluate the procedure in the "red" row *(in this case, a recursive self-reference).*

As another variation, Figure 3.44 illustrates a program that "has the turtle toot and walk in a different direction when it hits something."

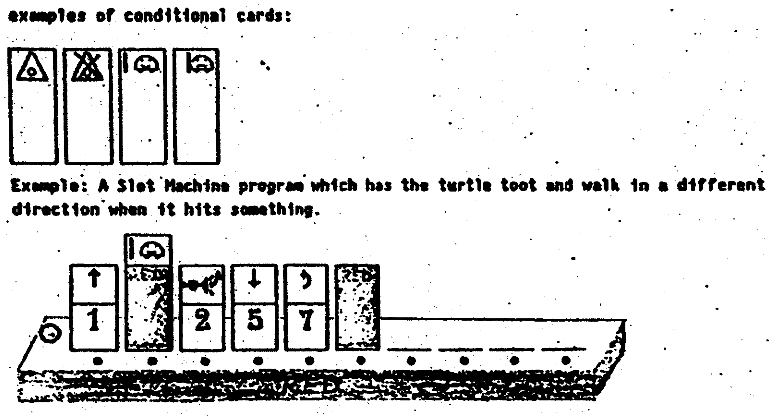


**Figure 3.44 "Slot machine," recursive/loop + conditional example [Perlman 1976]**

The Slot Machine illustrates how relatively complex concepts and behaviors can be expressed in tangible form. However, it also hints at some of the scalability limitations of tangible interfaces, and speaks less directly to how tangible interfaces might be applied to "grown-up" application contexts. The slot machine also relies heavily on the symbolic language printed upon the cards. While a powerful approach that has been adopted by recent TUIs such as Nelson et al.'s Paper Palette [1999] and DataTiles [Rekimoto et al. 2001], the slot machine makes somewhat more limited use of physical manipulation than many TUIs. For example, from the "tokens + constraints" perspective, the slot machine makes strong use of the "associate" phase, but does not support a "manipulate" phase. Alternately expressed, a card may enter or exit a slot, but no further physical manipulation of the card is supported once it is within the slot.

Another early token+constraint system is the LegoWall interface of Molenbach (as described in [Fitzmaurice 1996]). The LegoWall system implemented a wall-based matrix of electronically sensed LEGO bricks, which was applied to a ship scheduling application (Figure 3.45). The axes of the matrix were mapped to time of day and different shipping ports. LEGO objects representing different ships could be plugged into grid locations corresponding to scheduled arrival dates, or attached to cells allowing the display and printing of information about these ships.

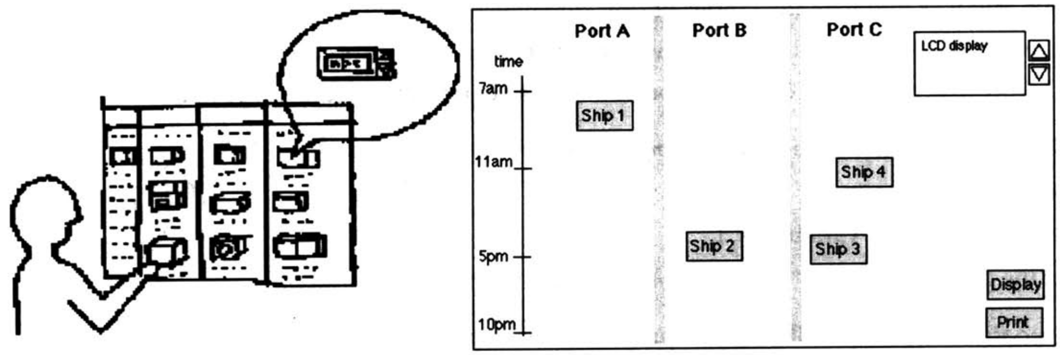


**Figure 3.45: "LegoWall" (described in [Fitzmaurice 1996])**

As illustrated by Figure 3.45b, the different port columns appear to have served as kinds of "constraints," while vertical movement of ship tokens within these constraints mapped to scheduling within time. While the LEGO™ grid attachment means the ship tokens cannot be moved continuously, it is not clear that this is demanded by the application. The token / constraint mapping employed is a relatively simple one, sharing a common language with common informal uses of magnetic tokens upon whiteboards. Nonetheless, this also speaks to a potential wide space of applications.

Bishop's influential Marble Answering Machine concept sketch illustrated the use of physical marbles as containers and controls for manipulating voice messages [Polynor 1995]. The marbles are moved between active surfaces to replay marble contents, redial a marble message's caller, or store the message for future reference (Figure 3.46). In addition to the marble answering machine, Bishop developed a broader series of designs exploring the manipulation of physically-instantiated information [Abrams 1999]. Bishop's designs provided one of the earliest illustrations for interlinking systems of physical products through a shared physical/digital "language," and were one of the most direct inspirations for the work of this thesis.

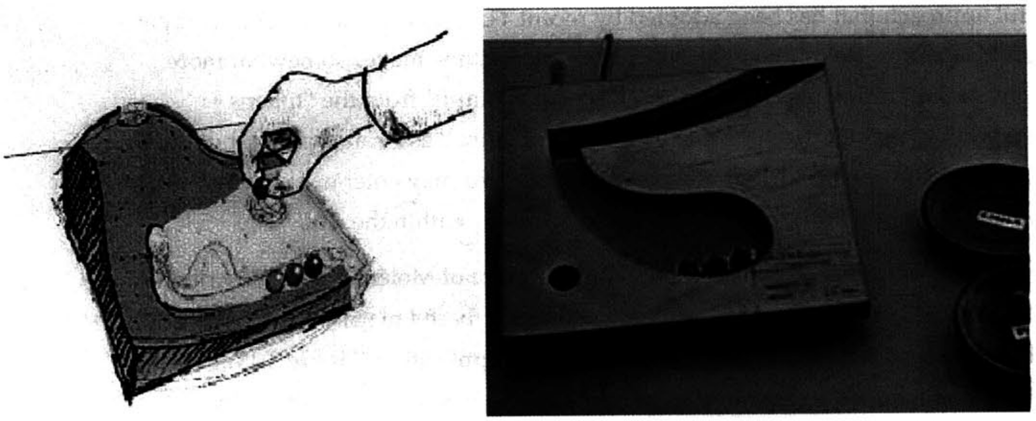


**Figure 3.46: Marble answering machine, animation and physical prototype [Bishop 1992]**

Bishop's designs illustrated a number of important functions that were utilized in the thesis' mediaBlocks system. These included the concept of physical objects as "containers" for digital media, and the use of these physical containers to "transport" digital media between multiple devices. Bishop also highlighted the importance of "out-of-band" manipulation of physical / digital tokens, such as the passive storage of marble-messages in labeled dishes for reference by other answering machine recipients (Figure 3.46b).

Moreover, Bishop designed a token "gateway" to a computer monitor (resembling the mediaBlocks' monitor slot) through which arbitrary digital contents from the GUI desktop could be associated with physical objects. These digital associations included folders of information, making Bishop's work the first to represent aggregates of digital information with physical objects. However, Bishop's designs did not support a way to physically operate upon these aggregate bindings outside of the computer, as realized by the mediaBlocks sequencer and query interfaces' query racks.

Like the mediaBlocks and tangible query interfaces, the LogJam video logging [Cohen et al. 1999] and ToonTown audio conferencing [Singer et al. 1999] systems also drew inspiration from Bishop's work. Both of these systems were based upon the configuration of physical tokens upon a multi-tier rack structure (described by the developers as a "game board"). In the LogJam system, domino-like physical blocks represented categories of video annotations. These category blocks were added and removed to the racks to annotate video footage by a group of video loggers (Figure 3.47). Like the above systems, LogJam did not employ the "manipulate" phase of token+constraint interaction; it interpreted only the presence or absence of tokens from its array of racks.

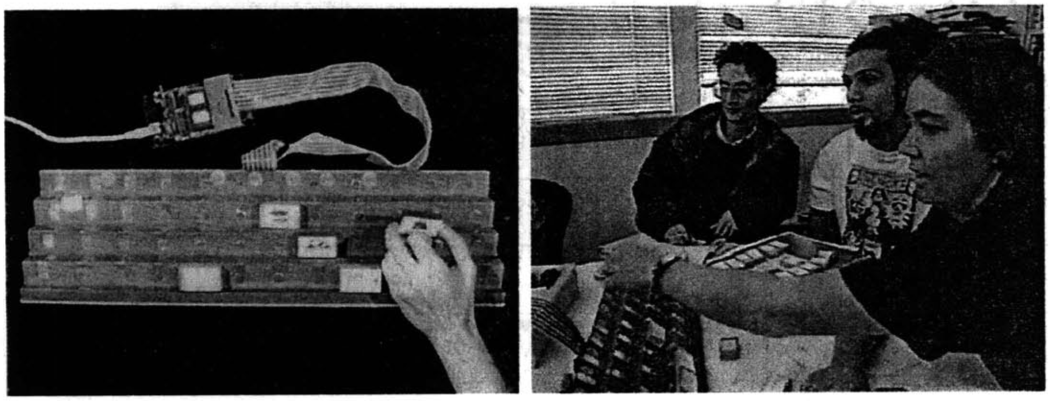


**Figure 3.47: LogJam: prototype close-up, system in use [Cohen et al. 1999]**

The LogJam system was actively used in group sessions by video loggers, and was positively received. The system was not observed to result in faster completion of the logging task; perhaps to the converse, it was found to encourage (productive) discussions that likely led to slower completion times. However, users did find LogJam more enjoyable to use over GUI alternatives, and the system fostered a variety of useful impromptu manipulations that had not been anticipated by the system's designers.

For example, all of LogJam's users made "out-of-band" configuration of their category blocks, organizing these blocks in front of them with individualized layouts and groupings. Second, users spontaneously employed behaviors like "sweeping" groups of blocks off the rack with one or both hands; and "snatching" blocks from colleague's spaces, when others were slow to activate them. These kinds of behavior, together with other open-ended modes of interacting with the LogJam system, seemed to strongly distinguish its use from that of GUI predecessors.

The ToonTown system, developed in parallel with LogJam at Interval Research, developed a tangible interface for controlling multi-user presence within an audio space [Singer et al. 1999]. ToonTown uses physical tokens topped with cartoon characters to represent users within the audio space (Figure 3.48). Manipulation of these tokens upon an array of racks allows the addition and removal of users from the space; the audio localization of individual users; assignment of users to tokens; and the display of information relating to individual participants. The associated mappings of the rack array are illustrated in Figure 3.49.

**Figure 3.48: ToonTown: close-up of prototype with tokens [Singer et al. 1999]**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Left pan | | S | O | F | T | E | S | T | | Right pan | |
| Left pan | | | | | Center Pan | | | | | Right pan | |
| Left pan | | | | | Center Pan | | | | | Right pan | Info Zone |
| Left pan | | L | O | U | D | E | S | T | | Right pan | Assign Zone |

**Figure 3.49: ToonTown: mapping of sensing cells within racks [Singer et al. 1999]**

The ToonTown system includes a number of interesting and provocative components. One of these is physical representation of people. Outside of Bishop's design studies, I believe this is the first such usage within a tangible interface. I believe this design choice has powerful potential in future communication systems. Second, I believe ToonTown's mapping of linear position to left/right fade is the only published use of the "manipulate" phase of token/constraint interaction outside of the thesis' mediaBlocks and query interface systems.

Third, the "ToonTown" tangible interface was the fourth interface designed for control of the audio space, and was designed after two prior graphical interface designs were found inadequate for the task. Speaking of this experience, the authors suggest as a design guideline:

> **GUI interfaces are a poor choice for audio spaces**
>
> In retrospect, it is not surprising that a graphical user interface is not optimal for interacting with an auditory experience. Audio communication does not demand visual attention. Furthermore, an audio space works like a utility and thus calls for a simple interface. With Thunderwire [a physical control prototype that allowed only "on/off" engagement with the audio space], we took simplicity too far by eliminating all forms of control and display. Some kind of tangible representation, building on a simpler version of the ToonTown model, might be the more appropriately balanced interaction mechanism. [Singer et al. 1999]

While other statements of this kind may well exist, this is the first claim I have encountered that asserts graphical interfaces are categorically inappropriate for a major area of computer-mediated interaction, and for which tangible interfaces seem well suited. ToonTown supported a kind of persistent, "background" mode of operation, offering a clean solution for a task where more "foreground" graphical approaches were found lacking.

Another TUI for manipulating audio content is the "Music Blocks" system, which (along with Zowie's playset products) was one of the first tangible interfaces to be commercially marketed [Neurosmith 1999]. This system binds different musical fragments to the faces of physical cubes, mapped as a function of the color of the blocks and the shapes on their sides (Figure 3.50). Blocks can be sequenced within several consecutive receptacles, and new music mappings can be exchanged with Wintel computesr via a "Cyber Cartridge" memory module.

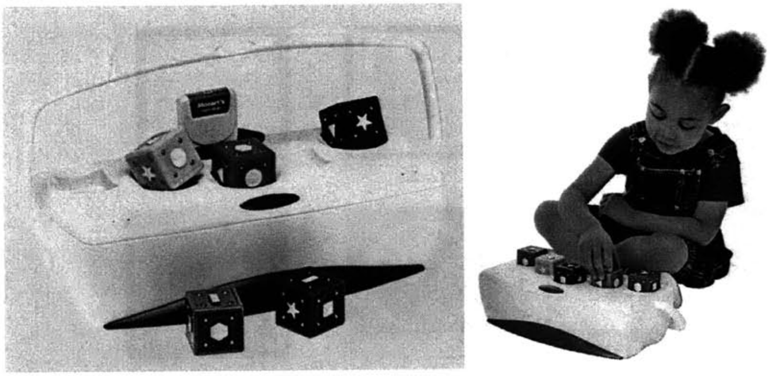


**Figure 3.50: Music Blocks [Neurosmith 1999]**

Perhaps the first token+constraint system to utilize force feedback is the "tagged handles" research of MacLean et al. [2000]. As with this thesis, this works highlights the combination of discrete and continuous modes of interaction. For example, the concept sketch of Figure 3.51a cited mediaBlocks as a starting point, and suggested an approach for giving mechanical detents to mediaBlocks manipulation. Ishii, Glas, and I considered adding such force feedback during the original mediaBlocks design, but this was technically difficult to resolve without "block docks" (illustrated in Figure 3.51a), especially for the sequence rack's multi-token syntax.

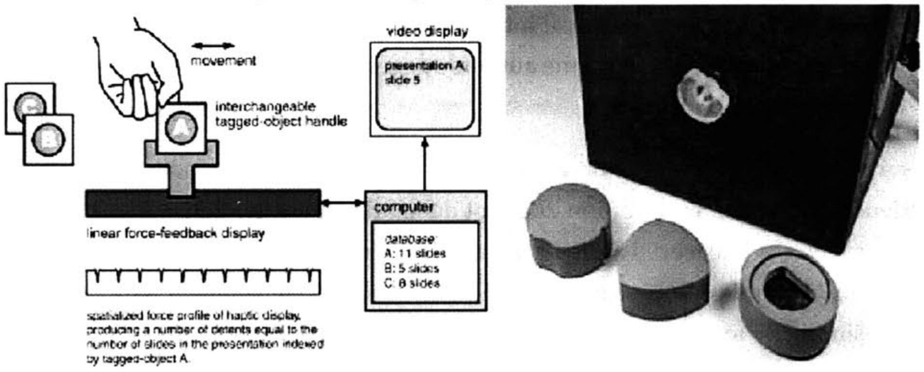


**Figure 3.51: Tagged concept and prototype** [Maclean et al. 2000]

Alternately, Figure 3.51b shows a working prototype with tokens resembling the parameter wheels. Collaborator James Patten and I partially implemented a force-feedback parameter wheels implementation in 1999, but left this aside to focus on the core parameter wheel concept. Nonetheless, as discussed in §6.6.1, adding force feedback to the parameter wheels could be a powerful aid to resolving some difficulties encountered during the experimental evaluation.

A final example related to the token and constraints approach is the DataTiles system [Rekimoto et al. 2001], which I participated in developing with Rekimoto and Oba. DataTiles used transparent plastic tiles to represent modular software elements which could be composed on a graphically augmented 2D grid. These tiles were faced with partially transparent printed matter and pen-constraining grooves that allowed tiles to be persistently associated with classes of information and functionality. Augmenting information and interactive manipulations were then mediated by dynamic computer graphics (Figure 3.52).

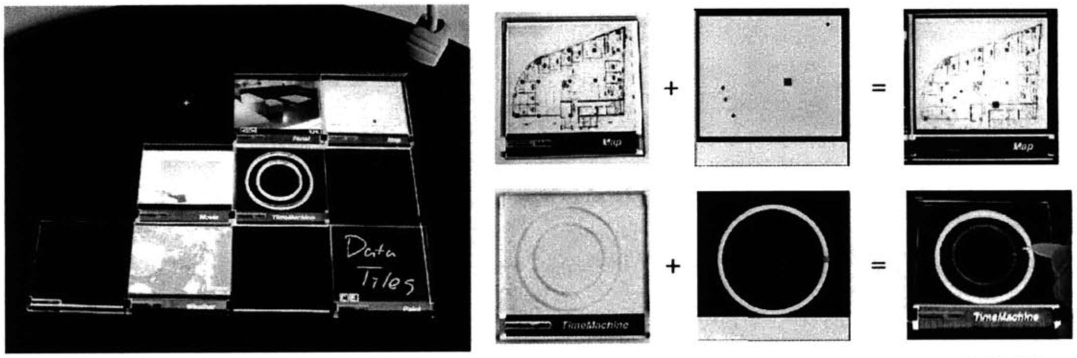


**Figure 3.52: DataTiles system, combination of physical + digital elements [Rekimoto et al. 2001]**

DataTiles were designed to be combined together using an open-ended "tile grammar," which is discussed further in §7.1.1. As with the thesis' query interfaces, DataTiles mapped computational semantics to the physical adjacencies of its tiles, and included a parameter tile for simple database queries. However, DataTiles relied upon pen-based interaction with GUI applets displayed within the tiles, which contrasts with the thesis' emphasis on physical representation and manipulation. Nonetheless, this hybrid approach supports many of the strengths of both physical and graphical interaction techniques, and seems a promising direction for continuing research.

## 3.6 Terminology

Throughout the dissertation I have somewhat loosely used a number of terms like "object," "artifact," and "container." As is common in rapidly evolving research areas, many of these terms have not yet reached widespread consensus (including the "tangible user interface" phrase itself). With such a consensus likely to be slow in coming, it is valuable to consider the terminology currently in use, as the choice of names often reflects important underlying assumptions.

"Objects," in the physical sense of the word, are clearly a central concern of tangible interfaces. At the same time, this term has been broadly interpreted in the computer science and HCI communities to mean many things, most having nothing to do with the physical world. Moreover, most physical objects have no connection with tangible interfaces. Therefore, while I have often discussed "objects" in the TUI context, it is a somewhat ambiguous term. The term "physical/digital objects" is sometimes used to clarify this ambiguity, highlighting the dual physical/digital aspect of TUI elements.

"Artifacts," carrying the implication of man-made physical objects, offers an alternate term with less prior use in the computer science and HCI communities. However, naturally occurring objects like stones and seashells have been used in several tangible interfaces, leaving this term again useful but imprecise.

The "props" term has been used in several related research systems, including Hinckley et al.'s influential "doll's head" neurosurgical interface [Hinckley 94]. However, the props term carries the implication of an element that is somehow peripheral to the core (presumably graphical) user interface. This is somewhat counter to TUI's emphasis upon physical objects as central elements of the user interface.

"Physical icons" or "phicons," a name introduced by Ishii and myself in [1997] with reference to the GUI "icon" concept, offers another possible descriptor. However, as I have discussed in [Ullmer and Ishii 2000], this term also has shortcomings. For one, it faces a dilemma that has been widely discussed in the GUI literature: strictly speaking, many so-called "icons" (and "phicons") are not "iconic," but rather "symbolic" in form. For instance, from the perspective of semiotics (the study of signs and symbols), the physical forms of mediaBlocks are symbolic, and not iconic.

The "tangibles" term refers specifically to the physical elements of tangible interfaces, and to their role in physically representing digital information. Partially inspired by the Marble Answering Machine and other work of Bishop [Polynor 1995], it was used in this context with the development of the LogJam video logging and ToonTown audio conferencing systems at Interval Research [Cohen et al. 1999, Singer et al. 1999]. The term was also used in earlier work including the DigitalDesk of Wellner [1991] and the AlgoBlocks of Suzuki and Kato [1993], appearing as a keyword in the title of both works.

The "tangibles" term has the advantage of brevity and specificity to the TUI context. It also suggests "intangibles" as a natural counterpart, helping to draw the distinction between interface components that are physically embodied, and those that are manifested in purely intangible form (graphics, audio, etc.).

## 3.7 Classification of TUI elements

Many of the tangible interfaces presented in this chapter can be considered in terms of "tokens" and "reference frames." Here, *tokens* are considered to be the physically manipulable elements of tangible interfaces, and *reference frames* are the physical interaction spaces in which these objects are used. For systems following the interactive surfaces approach, the primary interactive surface (often a workbench or wall) usually serves as the major reference frame. For token+constraint systems, the constraints serve as reference frames. Constructive assembly approaches often build elements out from a "seed" or "mother" object (which is frequently tethered to a controlling computer), but also can be referenced from a supporting table or wall surface, or implicitly referenced with respect to the users' hands, etc.

From an applied perspective, symbolic tokens are often used as "containers" for other media (as in mediaBlocks). Similarly, tokens that represent digital operations or functions often serve as "tools" (as in Urp). Where the token and reference frame terms are relatively new to the discussion of tangible interfaces, the "container" and "tool" terms have seen fairly widespread use.

Holmquist, Redström, and Ljungstrand suggest the terms "tokens," "containers," and "tools" as classifications for physical/digital objects [Holmquist 1999]. Where in this thesis "containers" are considered to be a kind of physical token, Holmquist et al. consider tokens as specific to iconic representations, similar to the building "phicons" (physical icons) of the metaDESK [Ullmer and Ishii 1997]. Alternately, Underkoffler presents a "continuum of object meanings," with objects interpreted as reconfigurable tools, verbs, nouns, attributes, and "pure objects" [Underkoffler 1999a].

## 3.8 Discussion

This chapter has briefly presented and contextualized a number of examples of tangible interfaces and related approaches. The set of systems I have reviewed is far from exhaustive, and has left undiscussed many compelling systems that may productively be considered as tangible interfaces. Moreover, the three high-level approaches I have discussed – interactive surfaces, constructive assemblies, and tokens+constraints – are intended neither as a taxonomy, nor as an exhaustive description of the TUI design space.

Rather, my primary objective has been to provide a starting point for considering these many systems not as isolated instances, but as related elements of a larger, fairly well populated design space. These systems have shared attributes and similarities in approach that may be usefully compared amongst each other, and taken together lend support to the idea of tangible interfaces as a broader overarching concept.

In this chapter, I have chosen to organize TUIs on the basis of their physical architectures, as high-level approaches that lend themselves toward different application areas and styles of use. In other documents, I have explored different kinds of categorizations. For example, the papers of [Ullmer and Ishii 2000, 2001] discuss tangible interfaces in terms of four major mappings by which physical/digital elements can be combined: "spatial," "relational," "constructive," and "associative."

At a first approximation, these alternate categories correspond roughly to the categories used within the thesis. For example, constructive mappings tend to be built upon constructive physical architectures; spatial mappings are often realized on interactive surfaces; and relational mappings are frequently combined with token+constraint approaches. However, these correspondences are not "one-for-one." For example, a number of systems with constructive physical architectures (e.g., [Suzuki and Kato 1993; Gorbet et al. 1998; McNerney 2000]) utilize both constructive and relational mappings.

Similarly, while the Urp urban planning simulator [Underkoffler and Ishii 1999] makes heavy use of a spatial mapping on an interactive surface, its designers have begun to employ relational token + constraint approaches for controlling certain system parameters (e.g., wind orientation and time control). In continuing work with Urp, constructive approaches have also been developed, allowing building geometries to be physically expressed through the stacking of discrete elements and the shaping of amorphous materials [Piper et al. 2002].

These choices of organizing concepts are significant, as different styles of interaction are exposed and obscured by these decisions. However, it is neither reasonable nor productive to seek categories for tangible interfaces with the same rigor as, say, the periodic table's ordering of the chemical elements. The nature and semantics of user interfaces are governed by no such immutable physical laws. As suggested by examples like Urp, mature systems may often combine many strategies and mappings. These combinations may be analogous to the use of menus and toolbars and polygonal+voxel modeling within graphical CAD applications, and illustrate some of the many possible modes of expression within tangible interfaces.

# chapter 4

# mediaBlocks

# 4 mediaBlocks

The mediaBlocks system is the first of two major thesis projects to be constructed around a system of tokens and interpretive constraints. MediaBlocks developed from an interest in the user interface issues surrounding networked devices and online media. A popular and reasonably plausible belief holds that numerous devices that are not principally "computers" may soon be linked to the Internet, and used to access and manipulate online information. In part, this is driven by some of the special properties of networked content. Online information is freed from the confines of individual computers and open for access on a widely distributed basis. Moreover, online content immediately reflects changes and additions to the information source, which is important for many kinds of news, collections of photographs, and so forth.

However, the addition of embedded computation and network connectivity into a device exposes a number of potentially challenging user interface issues. If a user wishes to access online content through a device, how should this information be referenced? In the case of console-based computers, this is generally done by typing in a web address (e.g., a URL such as "http://www.mit.edu/"); typing key words into an online search engine; or navigating a textual or graphical page of hyperlinks. These approaches may be adequate for an individual user seated at a desktop computer, but are problematic for many usage contexts situated within other physical contexts.

As an alternative approach, the mediaBlocks system uses physical tokens as representations and controls for online information. The system is based upon a series of small wooden blocks called "mediaBlocks," which each can be bound to an element or collection of online media (images, video, audio, etc.). As an important aspect of this approach, mediaBlocks do not actually store media internally. Instead, they are embedded with digital ID tags that allow them to function as "containers" for online content, or from a technical standpoint, as a kind of physically embodied URL.

MediaBlocks interface with media input and output devices such as video cameras and projectors, allowing digital media to be easily "copied" from a media source and "pasted" into a media display. MediaBlocks also interoperate with traditional GUIs, providing seamless gateways between tangible and graphical interfaces. Finally, mediaBlocks are used as physical "controls" in tangible interfaces for tasks such as sequencing collections of media elements.

The chapter begins with an overview of the functionality provided by the mediaBlocks system. This functionality is then considered more carefully in the context of the interpretive constraint approaches investigated within the thesis. The latter half of the chapter discusses the implementation of the mediaBlocks system, as well as the design of an exploratory system extension that lead to the second major thesis project.

The mediaBlocks system was first published in SIGGRAPH'98 [Ullmer et al. 1998], with a video published as [Ullmer and Ishii 1999].

## 4.1 Functionality overview

### 4.1.1 Physical containers

MediaBlocks are physical blocks that act as "containers" for online media. MediaBlocks do not actually store digital media internally. Instead, they are embedded with ID tags that are dynamically associated with sequences of online media elements. As such, mediaBlocks have a variety of interesting properties. Because contents remains online, mediaBlocks have unlimited "capacity" and rapid "transfer speed" (copying is instantaneous, while playback is a function of network bandwidth). For the same reason, a lost block is easily replaced. MediaBlocks may also contain live streaming media.

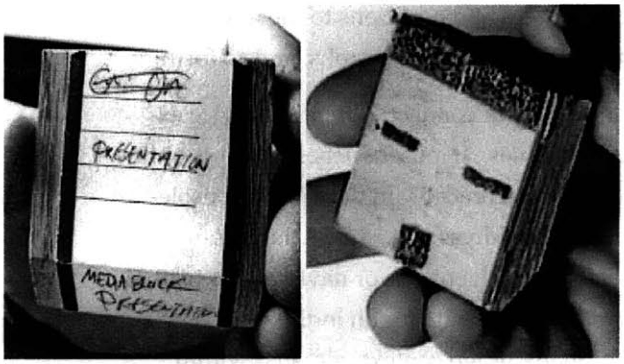


**Figure 4.1: mediaBlock tokens; front, back (showing contact pads)**

### 4.1.2 Physical transports

One role of mediaBlocks is support for simple physical capture, transport, and interchange of media between different media devices. While inter-application "copy and paste" is core to the modern GUI, comparably lightweight equivalents have not existed for physical media devices. The mediaBlocks system realizes a physical analog of "copy and paste" by combining blocks with physical slots that are mounted upon associated media devices.

The mediaBlock system implements support for four media devices: a wall-based video display, network printer, video camera, and digital whiteboard. Inserting a block into the slot of a media source begins recording to an online server. Recording stops when the block is removed. This can be understood as "copying" from the media source into the block. Similarly, contents may be "pasted" into a media display by inserting a block into the associated slot. This will display block contents, with removal halting playback.

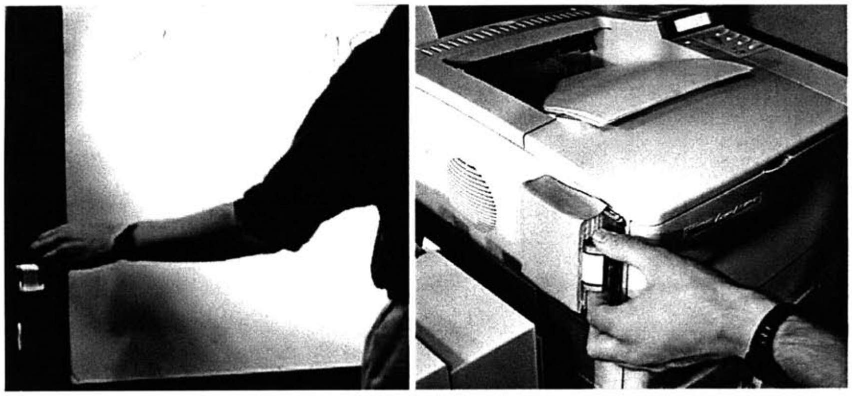


Figure 4.2: Whiteboard, printer mediaBlock slots

### 4.1.3 Physical gateways

MediaBlock slots have also been designed for use with general-purpose computers. Slots are mounted on the right margins of computer monitors. When a mediaBlock is inserted into the slot, a GUI window scrolls "out of the block" from the slot's left edge (contiguous with the screen). This window provides GUI access to block contents (Figure 4.3).

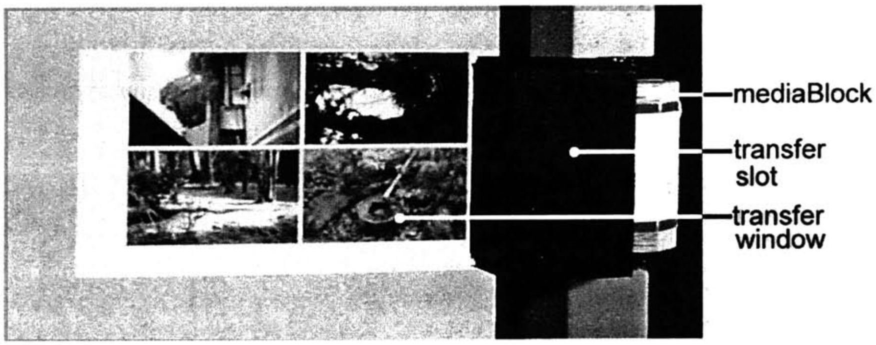


Figure 4.3: mediaBlock monitor slot

MediaBlock contents may then be transferred to the desktop or to GUI applications with conventional mouse-based "drag and drop" support. Media may also be copied into blocks in the same fashion. In this way, mediaBlocks can be used to seamlessly exchange digital contents between computers and media sources, displays, or other computers.

### 4.1.4 Physical browsers

While the transport function of mediaBlocks allows media to be exchanged between various output devices, it does not address interactive control of media playback, especially for mediaBlocks containing multiple media elements. The *media browser* is a simple tangible interface for navigating sequences of media elements stored in mediaBlocks (Figure 4.4).

The browser is composed of a detented browse wheel, a video monitor, and a mediaBlock slot. Useful both in casual viewing and formal presentation contexts, the browser supports the interactive navigation of mediaBlocks sequences for projector-based display, as well as displaying media on its local screen. A video display wall was also implemented with similar functionality.

Figure 4.4: Media browser device

### 4.1.5 Physical sequencers

The media browser provides interactive physical control of mediaBlock display, but does not support modification of mediaBlock contents. The *media sequencer* is a tangible interface that uses mediaBlocks both as containers and *controls* for physically sequencing media elements (Figure 4.5).

Where earlier sections have introduced mediaBlock slots, the sequencer uses physical *racks, chutes,* and *pads* as interpretive constraints that physically and digitally operate upon mediaBlocks. In particular, racks provide a constraint approach for digitally indexing and sequencing mediaBlock contents as a function of the blocks' physical configuration on the racks.

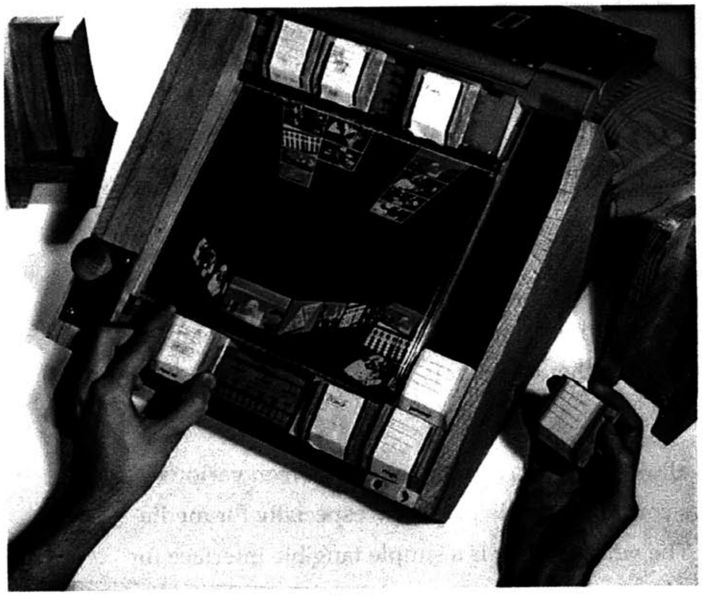


Figure 4.5: Media sequencer device

### 4.1.6 Example use

As a demonstration of the mediaBlock system's integrated functionality, the following scenario recalls the interaction sequence from the published video of this work [Ullmer and Ishii 1999a]. The first example presents an illustration of the mediaBlocks physical containment and transport functionality. The second example focuses upon the "control" aspect of mediaBlocks, using them to create an example multimedia presentation.

*First example: illustration of physical containment and transport*

The user inserts a mediaBlock into the slot of a digital whiteboard. The whiteboard responds with a musical chime, indicating that recording "into" the mediaBlock has begun. After the mediaBlock's insertion, all drawings made to the whiteboard are recorded into online space, and are associated with the mediaBlock object. When the drawing session is complete, the user removes the mediaBlock from the whiteboard slot. The whiteboard responds with a second chime, indicating that the recording is completed.

To illustrate the retrieval of this recorded content, this mediaBlock is next inserted into the slot of a network printer. While the printer is working, a new recording is made into a second mediaBlock. This time, the block is inserted into a slot associated with an overhead video camera. In some cases, this slot was placed upon a conference table, aiding users of the space to record meetings. In the published video, a slot is mounted near the printer for convenience of demonstration. Again, removing the mediaBlock from the slot stops the video recording, which is confirmed by an audio chime.

At this point, the printer has finished printing. The printout is retrieved, and the user moves on to a second display. First, the mediaBlock of the whiteboard recording is inserted into the slot of a back-projected wall display. The whiteboard recording is again downloaded from the network and is replayed on the display, this time with an accelerated playback of the original drawing's graphical "strokes." Secondly, the mediaBlock is removed and replaced with the block containing the overhead video recording. Again, the video is retrieved over the network, and rendered onto the video display.

These examples illustrate the use of mediaBlocks to bind and recall digital media from physically situated devices without the need for using a keyboard, mouse, or other GUI mechanism. At the same time, it is also useful to store and retrieve mediaBlock contents from traditional computer interfaces. As an example, the whiteboard recording mediaBlock is inserted into a slot mounted upon a conventional computer monitor. A window scrolls out of the slot, displaying the block's contents. GUI drag-and-drop may then be used to drag the whiteboard recording out of the block and into a supporting program.

*Second example: illustration of physical containment and transport*

The first example illustrates mediaBlock's basic container and transport functionality. In the second example, mediaBlocks are used to create a sample multimedia presentation. Toward this, mediaBlocks can be manipulated within the "media sequencer" device, which uses mediaBlocks themselves as physical controls to manipulate their digital media contents. For reference, the media sequencer's components are illustrated in Figure 4.6.

This second example begins with the whiteboard and video recordings from the earlier session. First, the whiteboard block is placed onto the sequencer's "sequence rack." A "digital shadow" appears beneath the block, indicating its digital contents. Next, the video recording mediaBlock is added to the sequence rack, to the right of the first block. Again, a digital shadow appears beneath the block, this time with the first frame of the video recording. In this configuration, a digital sequence of the whiteboard recording, followed by the video recording, is expressed. Alternately, swapping the physical order of the two blocks changes the sequence of their corresponding digital contents.

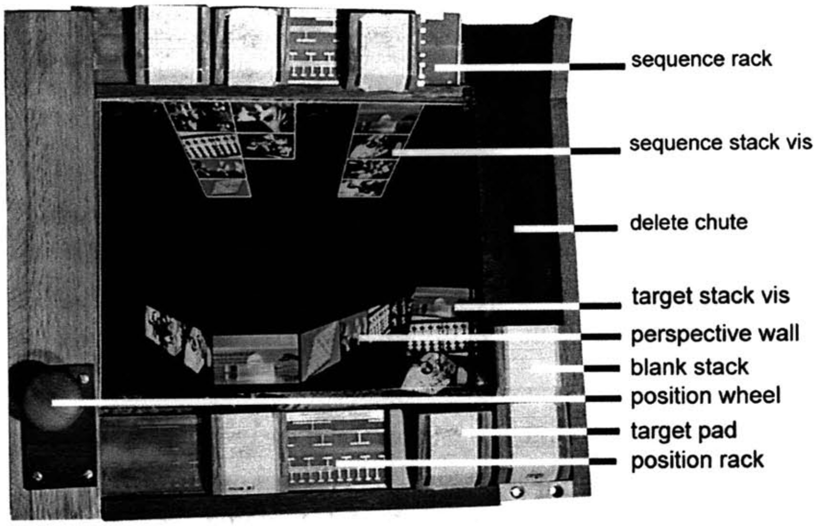


**Figure 4.6: Media sequencer components**

The sequence rack now contains four mediaBlocks: the title slide, whiteboard recording, video recording, and the container of two sample images. This sequence of five digital media elements is now copied to a blank mediaBlock, again using the target pad. (The source used by the target pad operation will be discussed in §4.4.) Again, a graphical animation illustrates this process. Alternately, the same sequence can be remotely dispatched to the printer using another kind of mediaBlock which is associated with the network printer's queue. When this block is placed into the target pad, any contents transferred into it are immediately submitted to the network printer.

When the sequencing activity is complete, the four blocks on the sequence rack may optionally be pushed into the "delete chute." When mediaBlocks pass through the bottom of the chute, their contents are erased (or rather, their links to any associated media are cleared).

As a final example, the resulting multimedia presentation is brought back to the wall display for presentation and review. When the block is inserted into the display's mediaBlock slot, the first element of the mediaBlocks presentation is displayed (in this case, the title slide). The remaining contents can be navigated using the display's browse wheel.

### *Discussion of example*

A video of the above interaction sequence appears in [Ullmer and Ishii 1999a] as five minutes of largely uncut footage. The example shows the creation, manipulation, and use of diverse multimedia content with a simplicity and speed that is highly competitive with other interaction approaches. Also, it is worth noting that the only keyboard, mouse, or other GUI interaction present in the entire sequence is access to mediaBlock contents through the monitor slot.

## 4.2 Use of tokens and constraints

The last two sections have described and illustrated the use of a number of different interpretive constraints. These structures are illustrated in Figure 4.7, and include:

- *Slots:* used to bind and recall digital contents to/from mediaBlocks
- *Racks:* used to aggregate and disaggregate mediaBlock bindings
- *Chutes:* used to delete mediaBlock contents
- *Pads:* used for establishing mediaBlock bindings within the sequencer device

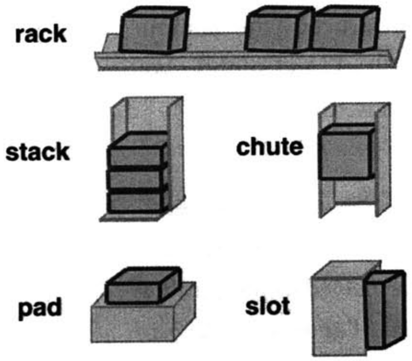


**Figure 4.7: Interpretive constraints within the mediaBlocks system**

These structures and functions may also be viewed in terms of the physical relationships and digital interpretations discussed in Figure 1.5. This is reproduced and annotated in Figure 4.8.

| Physical relationships | Interaction Event | Digital interpretations |
|---|---|---|
| Presence | Add/Remove | Logical assertion; activation; binding |
| Position | Move | Geometric; Indexing; Scalar |
| Sequence | Order change | Sequencing; Query ordering |
| Proximity | Prox. change | Relationship strength (e.g., fuzzy set) |
| Connection | Connect/Discon. | Logical flow; scope of influence |
| Adjacency | Adjacent/NAdj. | Booleans; Axes; other paired relations |

**Figure 4.8: Physical relationships and digital interpretations in mediaBlocks system**

The first physical relationship, token *presence,* relates to the "association phase" introduced in §1.1.2, and plays a strong role in all of the mediaBlock devices. In the case of slots mounted upon media recording and playback devices, the *binding* and *activation* interpretations are used, respectively. It is worth noting that each of the media devices discussed provides either recording or playback functionality, but not both. This point will be returned to later. In the case of the monitor slot, the *logical assertion* interpretation more closely describes the actual behavior;

i.e., insertion causes the mediaBlock to be digitally represented upon the screen, at which point contents may be added or removed from the block via a GUI.

The sequencer's chute and pads function also makes use of the presence relation and binding interpretation. The delete chute clears any existing mediaBlock bindings, while the target pad binds blocks to digital content from the position and sequence racks. The target pad is discussed further in §4.4.

The racks make use of both the presence, position, and sequence relations. In both racks, the addition of mediaBlocks (the "association phase") acts as a kind of logical assertion, expressing that their digital contents should be interpreted by the sequencing or indexing operators. The physical sequence and position relations (the "manipulation phase") in turn map to the digital concatenation and indexing operations, respectively.

As introduced in §1.4 and §2.6.3, the mediaBlock sequencer's racks take advantage of many of the advantages of interpretive constraints. The constraints simplify and structure the manipulation and interpretation of mediaBlock tokens, provide kinesthetic feedback, and support both the "epistemic" and "pragmatic" roles of object-manipulation discussed within §2.2.4.
The racks will be considered in further detail within §4.4.

As discussed in Chapter 1, syntax within the mediaBlocks system can be viewed from several different levels of abstraction. The first is in terms of the individual interpretive constraints illustrated in Figure 4.7 and Figure 4.8. At a higher level, syntax can be viewed as the cumulative assembly of interpretive constraints present within a given interface. For example, Figure 4.9 highlights the four different interpretive constraints present within the mediaBlocks sequencer: the position and sequence racks, the delete chute, and the target pad. In the course of interaction with the sequencer, a given mediaBlock may move through each of these different structures, in each case causing different digital operations to be applied to the block's digital contents. In this fashion, the syntax of the sequencer's full palette of operations is expressed in physically embodied form.

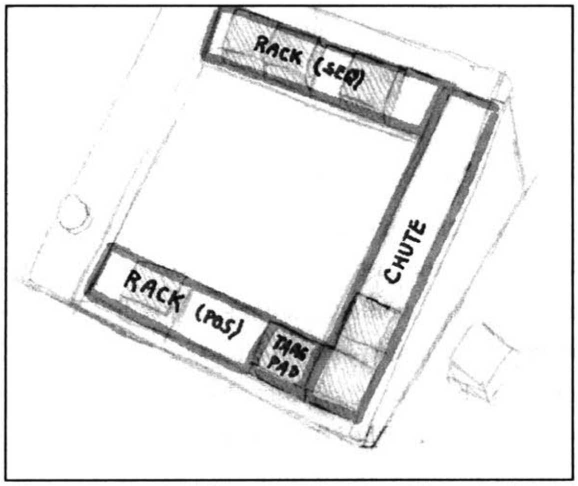


**Figure 4.9: Cumulative syntax of mediaBlocks sequencer**

## 4.3 Physical containment and transport

The most basic function of mediaBlocks is as tokens facilitating the capture, containment, and transport of digital media across diverse physical world devices. As mentioned before, this serves as a kind of physical "copy and paste" behavior for digital media, exported from the GUI world to physical devices and contexts. Figure 4.10 presents a loose illustration of this behavior. MediaBlocks serve as a medium of interchange between media source and display devices; between media devices and GUI-based computers; and between these pre-existing devices and new tangible interfaces for media manipulation.

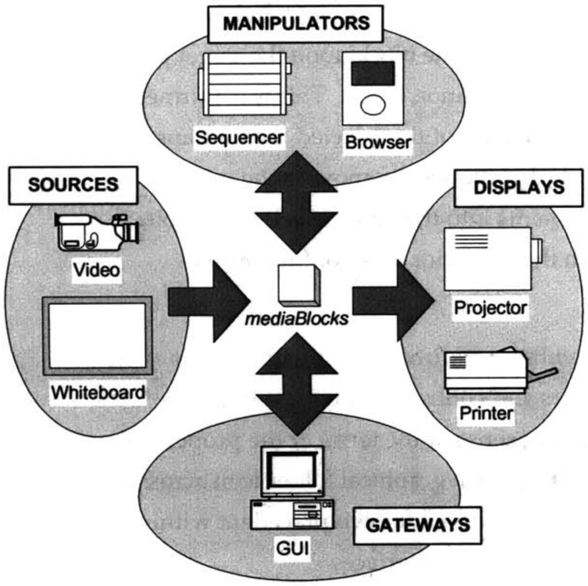


**Figure 4.10: Physical interchange of mediaBlocks between supporting devices**

## 4.4 Physical manipulation of digital contents

The slot-based binding of digital media to physical blocks is an important component of the mediaBlocks system, and provides the physical/digital bindings that are a prerequisite for subsequent interactions. While mediaBlock slots employ the "associate phase" of token+constraint interaction, they do not employ the "manipulate phase." The manipulate phase is developed by the media sequencer, illustrating how the combination of multiple interpretive constraints can be used to actively manipulate and recombine digital media.

The *rack* is the primary element of the media sequencer. The mediaBlocks rack was inspired by the Scrabble™ board game's combination of letter tiles and tile rack. In Scrabble™, the rack facilitates the rapid insertion, sequencing, and grouping of letter tiles into meaningful patterns. In the sequencer context, these physical attributes of position, sequence, and adjacency may be digitally recast as indexing, sequencing, and Boolean AND/OR operations, respectively.

Interaction with the media sequencer is dominated by two interpretive constraints: the *position rack* and *sequence rack*. When a mediaBlock is placed in the position rack, its contents are shown on the sequencer display as a perspective wall [Mackinlay et al. 1991]. The focus position of the perspective wall is interactively controlled by the relative position of the mediaBlock on the position rack (Figure 4.11). Moving the block to the rack's left edge moves the perspective wall to focus on the block's first element. Similarly, the right edge of the position rack corresponds to the block's last element.

The combined position rack and perspective wall serve several purposes. First, they support the interactive viewing of mediaBlock contents with imaging of both detail and context [Mackinlay et al. 1991]. Secondly, they allow the position rack to select an individual media element for copying between mediaBlocks.

Toward this end, a destination mediaBlock can be placed in the sequencer's *target pad*, which is physically adjacent to the position rack (see Figure 4.6). Pressing the block upon the target pad will append the currently selected media element into the destination block. This is confirmed with audio feedback and a haptic "click," as well as an animation of the selected media transferring "into" the target block. Pressing and holding the block, followed by moving the source mediaBlock to a new location, might copy a range of elements into the target block (analogous to dragging out a selection range with a mouse), though this behavior was not fully implemented.

In these interactions, mediaBlocks are used as *physical controls* for directly acting upon their internal state. Thus, mediaBlocks serve as both containers and controls, realizing a kind of "multiple inheritance" (in the object-oriented programming sense of the term) of the properties of both roles. The nearest equivalent in GUIs might be dragging a graphical folder icon across a scrollbar to index through folder contents. While a behavior without known precedent within GUIs, this usage seems to hold substantial promise within tangible interfaces.

The *sequence rack* extends this control functionality of mediaBlocks. This rack allows users to combine the contents of multiple mediaBlocks into a single sequence, which can then be associated with a new mediaBlock container through the target pad. When a mediaBlock is placed onto the sequence rack, its contents scroll out of the block into the sequencer display space (Figure 4.6). Multiple mediaBlocks may be arranged to construct a new digital sequence.

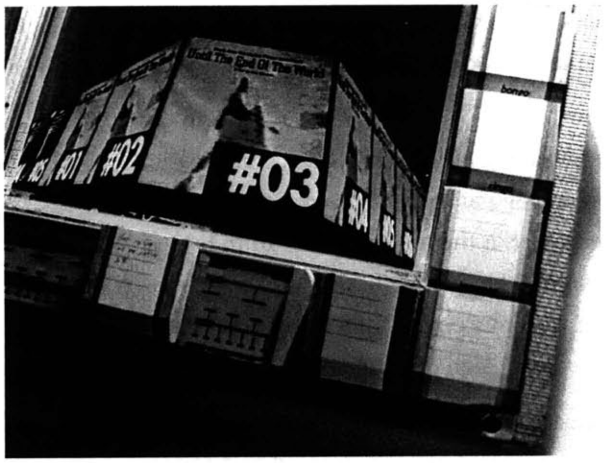


**Figure 4.11: Media sequencer perspective wall (alternate view)**
*associated with movement of mediaBlock on position rack;*
*music from several compact disks is "contained" within this block*

Both the sequencer screen and target pad are shared between the position and sequence racks. When a mediaBlock is located on the position rack, the target pad is bound to selections from the perspective wall display. When the position rack is clear, the target pad is associated with recording of the sequence rack's aggregate elements.

### *4.4.1 Challenges with consistency: the position wheel*

Especially when a source mediaBlock contains many elements, navigating the perspective wall by incremental steps may be more convenient than using the position rack. The *position wheel,* located to the left of the position rack, supports such incremental navigation (Figure 4.6). Haptic detents provide the user with physical feedback, where each detent corresponds with movement of one media element. Turning the wheel one detent to the left moves the perspective wall one element to the left (and similarly with rightward movement).

In the initial concept, the position wheel served a purpose very similar to the "scroll arrows" commonly integrated within GUI scrollbars (e.g., see [Apple 1992, p.163]). However, in practice, the position wheel exposed an important difference between graphical and tangible interfaces. When users employ the scroll arrows of GUI scrollbars, the scrollbar itself can graphically shift to reflect the new document configuration. Since the media sequencer did not integrate active force feedback, the position of the mediaBlock within the position rack could not be updated by the system, as is done with GUI scrollbars when scroll arrows are used. This produces a potential inconsistency.

Several alternatives were explored to resolve this inconsistency. Perhaps the simplest is to remove the position wheel altogether. This eliminates the inconsistency, but also removes potentially important functionality. For example, the media sequencer's racks were intended to sense block positions with "five-bit" accuracy (e.g., supporting a total of 32 positions). However, if a mediaBlock contained 50 elements, it might be impossible to access certain media elements with the position rack (especially since the sequencer is not configured with a keyboard, which in GUIs allow secondary navigation via the cursor keys). In practice, only four discrete sensing positions were implemented with the mediaBlocks racks, making the position wheel especially critical.

In practice, the position wheel was retained, with interpretation of the wheel/rack combination determined by a kind of hysteresis. When the position wheel was turned, the position rack's perspective wall display responded accordingly. Alternately, if the mediaBlock within the position rack was moved, the perspective wall would smoothly animate back to the newly specified position. This seemed to be a reasonable kind of compromise, although there was insufficient user experience with the system to evaluate this design decision.

As another design alternative, the use of graphical feedback to indicate the disparity between the logical selection position and the physical position of the block was considered and experimentally implemented. However, this seemed potentially more confusing than omitting this feedback altogether, and this addition was removed. Again, were the system to have been developed further, user feedback with these different alternatives would have been in order.

The use of force feedback was also actively considered, such as integrating a "conveyer belt" mechanism into the rack. However, this both presented technical challenges, and seemed like an excessive response to a relatively "minor" inconsistency. Nonetheless, such an alternative was proposed (but not implemented) by Maclean et al. [2000]. Maclean et al. also commented on the potential utility of active force feedback to indicate distinctive properties of media within such a rack.

## 4.5 Physical design

In the following sections, the physical, electronic, software, and overall system design and implementation of the mediaBlocks interface will be discussed.

### 4.5.1 Blocks

Initial experiments on the physical materials and forms for mediaBlocks were conducted with a series of acrylic, aluminum, and wooden tokens fabricated using a band saw, lathe, and CNC mill. Tokens of varying shape, size, thickness, mass, translucency, and mixed material composition (e.g., wood-framed acrylic) were all explored as a part of this process (Figure 4.12).

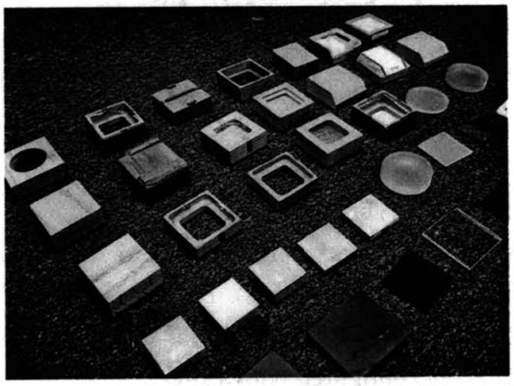


**Figure 4.12: Experiments with physical form of mediaBlocks**

The final shape was not resolved until near the end of the sequencer prototype's development, although a 2x2x¾" generic block format was adopted as a working format. The final mediaBlock shape was evolved in consultation with a resident industrial designer. The designer executed a series of sketches, several of which were prototyped in wood and clay. The final design was constructed with wooden blocks band-sawed to size, with a belt-sanded bevel, a paper label, and a dremmeled base indentation for housing an ID tag.

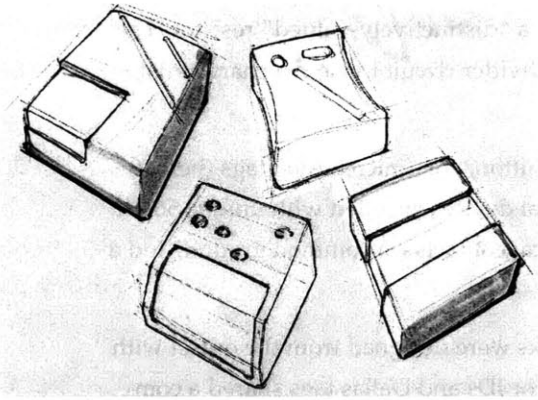


**Figure 4.13: Designer sketches of possible mediaBlock forms**

### 4.5.2 Sequencer

The mediaBlocks sequencer's physical design evolved in part through rough pen-and-paper sketches, but largely through a series of wooden and acrylic prototypes, constructed using a band saw, wood screws, and adhesives. These prototypes included stand-alone racks; intermediate sequencer prototypes, with mock-up screenshots computer- and hand-drawn paper inserts; and eventually the fully-constructed sequencer, including mounting fixtures for the flat panel display and other supporting electronics. This physical fabrication was done by Dylan Glas, a collaborator with strong machining and woodworking skills.

### 4.5.3 Slots

The mediaBlock system's slots were developed after Glas' departure. Without access to his fabrication expertise and with limited time for physical prototype construction, slots were initially implemented in layered foamcore (a material familiar to some as a backing surface for posters). Foamcore layers were cut to form with a utility knife, glued and taped together, and instrumented with copper tape for electrical contacts.

While careful execution might have produced better results, these slots were found to be bulky, crude, and subject to disintegration with extended use. A second revision was fashioned out of clear plastic utility drawers, which conveniently were available with matching dimension to the mediaBlocks, a "coincidence" considered within §2.2.2. These were cut to size and mounted with hot glue to a foamcore base, with formed copper strips used for electrical contacts (see §4.6.2). The assembled slots were taped to printers, monitors, whiteboards, etc. for situated use.

## 4.6 Electronics design

### *4.6.1 Blocks*

From an electronics standpoint, mediaBlocks themselves are relatively straightforward. Beyond their wooden carrier, mediaBlocks integrate electronic tags that were wired to electrical contacts, which in turn mediated the coupling of blocks with sensing cells (Figure 4.1).

Two technologies were used for mediaBlock tags: resistor IDs and "one-wire" digital ID tags from Dallas Semiconductor. Resistor IDs take the form of a "distinctively-valued" resistor. This can be measured against a reference resistor in a voltage-divider circuit to yield a characteristic voltage level.

Dallas Semiconductor's "one-wire," "touch memory," "iButton," or "microLAN" tags (hereafter referred to as "Dallas tags") are a family of small digital devices encoded with unique 56-bit IDs. These tags may be digitally queried with two electrical contacts – a common ground, and a data line that also serves as "phantom power."

While initially implemented with resistor IDs, mediaBlocks were designed from the outset with Dallas tags in mind. From a mechanical standpoint, resistor IDs and Dallas tags shared a common method of electronic coupling – a pair of electrical contacts. The resistor ID approach afforded a quick first implementation by way of commercial A/D (analog-to-digital) hardware. The Dallas tags offered a longer term unique-ID solution, at the cost of somewhat more complex implementation.

The electrical contacts of these tags were attached to two pads on the block's bottom, in a configuration illustrated in Figure 4.14. The pads were made of a conductive velcro material originally identified within the Triangles project [Gorbet et al. 1998]. These "fuzzy" pads facilitate electrical contact over mechanically unreliable connections, at the cost of adding variable resistance to the coupling.

This variable resistance complicated the identification of resistor IDs, as it added a confounding variable to an already imprecise and variable resistive characterization. However, these pads offered substantially improved reliability for Dallas tags. The configuration of mediaBlock ground and data pads (the "tripod" arrangement illustrated in the bottom-left of Figure 4.14) was evolved experimentally to facilitate mechanical stability, electrical reliability, and sensing resolution on the rack.

### *4.6.2 Block sensing*

MediaBlocks are detected and identified through the electrical coupling of blocks to a block sensing cell. The sensing cell involves a complementary pair of electrical contacts, together with the tag technology's associated "decode" electronics. MediaBlock slots and pads each require a single pair of contacts, while racks consist of a linear array of data contacts (Figure 4.14). Slot and pad contacts were based on copper tape and formed copper strips. The rack was fabricated as a printed circuit board, designed with a commercial CAD program (Protel) and fabricated at a printed circuit board production house.

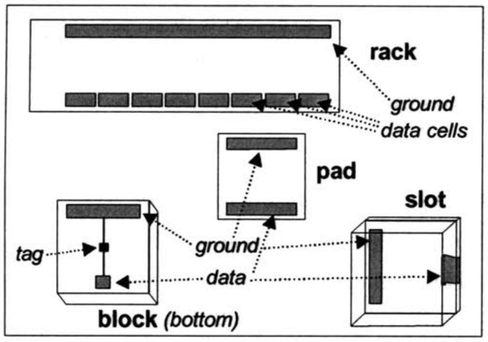


**Figure 4.14: mediaBlock contact configurations**

The reliability of the mediaBlock sensing system is partially a function of block mass and sensing-cell orientation, as some pressure of the block against the sensing pad is required for reliable electrical coupling. Magnets were used to improve a similar coupling within the LogJam and ToonTown prototypes, albeit in the absence of "fuzzy" contact pads [Cohen et al. 1999, Singer et al. 1999]. Our mediaBlock slot also uses a spring-like data contact to achieve an electrical coupling invariant to gravity's orientation. These sensing issues are discussed at greater length within Chapter 6.

For mediaBlocks using resistor IDs, block identification was performed with Infusion Systems' iCube device – a modular 24-input, 8-output, MIDI based sensor/actuator device. For mediaBlocks using Dallas tags, blocks were identified using circuits based upon PIC 16F84 microcontrollers – eighteen-pin integrated circuits that integrate an eight-bit microcontroller, modest RAM, EEPROM, and software RS232 serial support.

In the latter case, a miniature PIC development board, the iRX [Poor 1997], was used to minimize implementation time. Each 18-pin PIC was programmed to support nine independent channels of Dallas tag sensors. For both tag technology approaches, tag pull-up resistors and cable interconnects were prototyped on a solderless breadboard.

### *4.6.3 Other electronic components*

In addition to a collection of tag sensing cells, the mediaBlocks system employed several shaft encoders, switches, and a flat panel display. When resistor IDs were in use, the shaft encoders and switches were sensed via the iCube device. Later, these functions were coded in PIC firmware, and supported via additional iRX boards. The flat panel display was driven directly by an SGI Octane computer.

## 4.7 Software design

The mediaBlocks system was implemented with a custom-designed Tcl- and [incr Tcl]-based tangible interface software/hardware toolkit called 3wish [Ullmer 1997b]. 3wish includes modules supporting MIDI-based digitizers and synthesizers, Inventor-based 3D graphics, computer vision and magnetic field trackers, etc. 3wish also supports a distributed architecture called *proxy-distributed* or *proxdist computation* [Ullmer 1997a], which provides abstractions for mixed physical/digital systems such as the mediaBlocks interface.

The computational aspects of the mediaBlocks system can be considered at several levels of abstraction. First, the firmware and software underlying the system's operation will be considered. Secondly, the topology and distribution of functionality across the system as a whole will be discussed.

Sensor inputs were digitized with the Infusion Systems Icube device, and acquired through the 3wish extensions to the Tcl language [Ullmer 1997b] using C++ wrappers over the Rogus MIDI interface suite [Denckla 1997]. The position rack perspective wall and other graphical visualizations were written with 3wish's Open Inventor-based routines, and executed on an SGI Octane workstation.

### 4.7.1 Firmware, software, and APIs

The mediaBlocks system consists of 10 processors spanning three platforms – 5 PIC microcontrollers, 3 Wintel PCs, and 2 SGI workstations. Except for one Wintel PC, all of these machines are effectively dedicated computers for the mediaBlocks user interface.

These platforms interoperate to support the mediaBlock interface's two basic functions. First, the system manages the capture and playback of digital media on the system's five slot-based media devices. Secondly, the system supports the manipulation of mediaBlock contents on the media sequencer device.

The microcontrollers used within the mediaBlocks system were are all of the PIC 16F84 type, integrated on iRX 2.0 circuit boards [Poor 1997]. These microcontrollers work to sense the presence and identity of Dallas-tagged mediaBlocks, and to monitor the changing state of shaft encoders. Three PIC boards were configured to sense nine Dallas tag-cells apiece – one for an integrated tag uniquely labeling each tag reader board, and eight additional cells purposable by the developer. The remaining two PICs boards sensed four Dallas tag-cells and two shaft encoders apiece.

The PIC microcontrollers were programmed in C using the CCS development environment. Each PIC broadcasts the unique ID of its associated processor upon power-up using its RS232 serial line. Subsequent Dallas tag entrance/exit events and shaft encoder movement events are also broadcast via serial as ASCII strings.

The SGI and PC mediaBlock controller computers are each programmed in C++ and [incr Tcl] (iTcl), an object-oriented extension to the Tcl/Tk scripting language. Two C++ libraries make platform-specific features available from the Tcl language. On the PC, these libraries provide control of the digital whiteboard, the iCube, and a MIDI synthesizer, while a high-level Tcl interface to the Open Inventor 3D graphics toolkit is implemented on the SGI.

MediaBlocks user interface libraries are implemented as a series of six iTcl class hierarchies: mbPhysicals, mbDigitals, mbVisuals, mbAudibles, mbProxies, and mbDevices.

MbPhysical classes provide event-based abstractions of the slot, pad, sequence rack, and position rack TUI primitives. MbDigital classes implement database and client/server resources for storing, caching, retrieving, and manipulating the state of individual mediaBlocks, keyed upon their 56-bit unique ID.

MbVisual classes implement the perspective wall and stack visualizations that are rendered on the sequencer's flat panel display. MbAudible classes provide similar encapsulation for MIDI-based audio cues, which accompany slot entrance and exit events.

MbProxy classes integrate the highest-level physical, visual, and audible behaviors associated with the rack, slot, and pad primitives. Finally, mbDevices link the events of slot proxies to the media recording and playback facilities of individual devices.

### 4.7.2 Resolution of mediaBlock objects to contents

MediaBlock phicons are coupled to their media "contents" through several levels of indirection. These mappings include:

a) physical block → network address
b) network address → mediaBlock data structure
c) data structure element → individual media contents

The individual steps of this mapping process are illustrated in Figure 7. First, insertion of an ID-tagged block **(1)** is detected electronically by the slot **(2)**. The ID value is registered by a computer hosting the slot's tag reader **(3)**, and transmitted as a block entrance event to the display device's media manager **(4)**. The slot and media managers could be hosted on the same machine. In our implementation, the libraries supporting tag readers and media displays were specific to PC and SGI platforms, respectively, requiring separate computers for **(3)** and **(4)**.

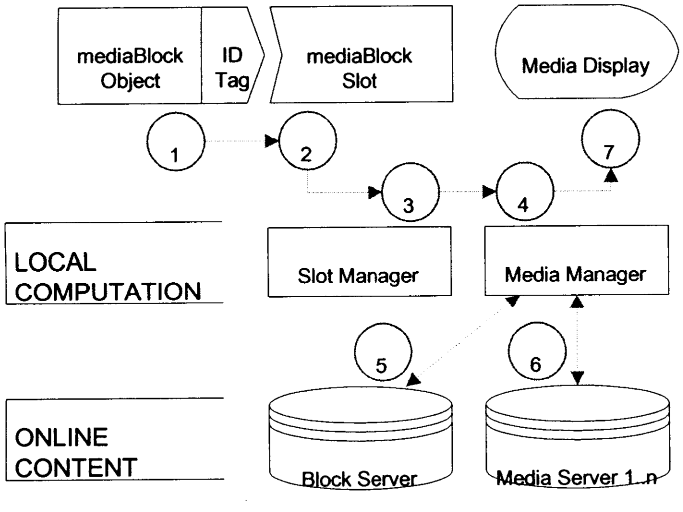


**Figure 4.15: mediaBlock display flow diagram**

Once an ID value has been obtained for a mediaBlock, the block server (5) is queried to retrieve a data structure describing the block's digital contents. The fields of this structure are described within Table 4.1. With the resistor ID scheme, a central block server is responsible for mapping block IDs to block data structures for all compatible block devices. However, the resistor-based approach does not scale for distributed operation.

The iButton-based mediaBlocks solve this problem by storing the URL of their hosting block server within internal nonvolatile RAM (4096 bits for the version that was used), allowing truly distributed operation. iButtons also support storage of encrypted data, potentially useful for authenticating mediaBlocks and granting read or write permissions by a given media device.

After retrieving the block data structure, the device media manager retrieves the specified media contents from one or more media servers **(6)**. This content is finally sent to the media display under control of display-specific MIME registries **(7)**, in a fashion resembling the plug-in registries of Web browsers.

**mediaList:** *List of contained media element addresses*
**physidType:** *Type of physical ID tag on block*
**physidInst:** *ID of block tag (usually a number)*
**mediaHost:** *Media stored on media- or block-server?*
**recordBehavior:** *New media appends or overwrites old?*
**lastObservedLocale:** *Last location block observed*
**lastObservedTime:** *Timestamp of last block sighting*
**blockLabel**: *Text describing block contents*
**blockLabelColor:** *Color of paper block label*

**Table 4.1: mediaBlock data structure**

Earlier sections have discussed the use of mediaBlocks as a medium of exchange between graphical and tangible interfaces. This works particularly well in conjunction with Microsoft's Internet Explorer "Internet shortcuts" feature (and equivalencies provided by other operating systems). "Internet shortcuts" allow distributed online media (e.g., URL-referenced Web documents) to be manipulated by the desktop and applications with most of the same provisions as files stored upon local disk drives.

While this feature was never fully integrated, media elements dragged out of monitor-slot mediaBlocks with GUI drag and drop were experimentally synthesized as "Internet shortcuts." This combination represents a step toward more seamless integration between the online media spaces of graphical and tangible interfaces.

## 4.8 System design

Figure 4.16 provides a high-level overview of the mediaBlocks system's components as actually implemented. The center column, media devices, lists the physical devices which were integrated with mediaBlock support. The left column shows devices underlying the operation of mediaBlock slots. The right column identifies the computers that manage media recording and playback.

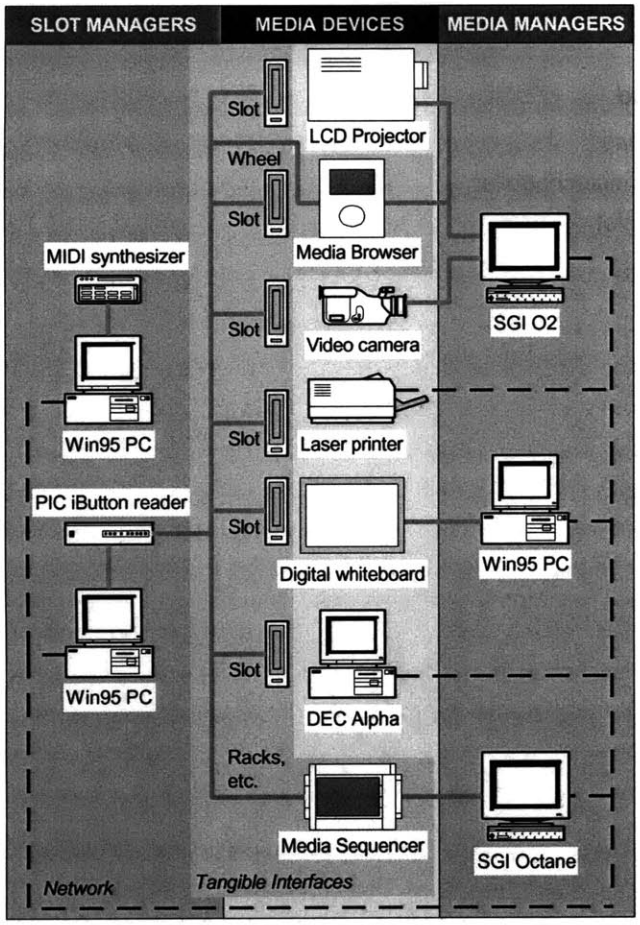


**Figure 4.16: First-generation mediaBlocks system diagram**

While this system configuration was functional, it was also rather ad hoc and emergent. It expresses the actual growth of the system under varying resource constraints, but not an architectural ideal for tangible interfaces of this sort. At a deeper level, Figure 4.16 indicates certain assumptions and pragmatics about what is "cheap" and what is "expensive." The figure reflects a system in which both computers and microcontrollers are "expensive" (in the microcontroller case, more as a function of time and effort than monetary cost), and optimization of these limited resources is an implicit goal.

Figure 4.17 illustrates a possible architectural rethinking of the mediaBlocks system. Rather than sharing microcontrollers and microprocessors between TUI primitives and slot-based devices whenever possible, the architecture of Figure 4.17 gives each TUI primitive its own microcontroller and ID tag, and every TUI-augmented media device its own dedicated, network-linked computer.

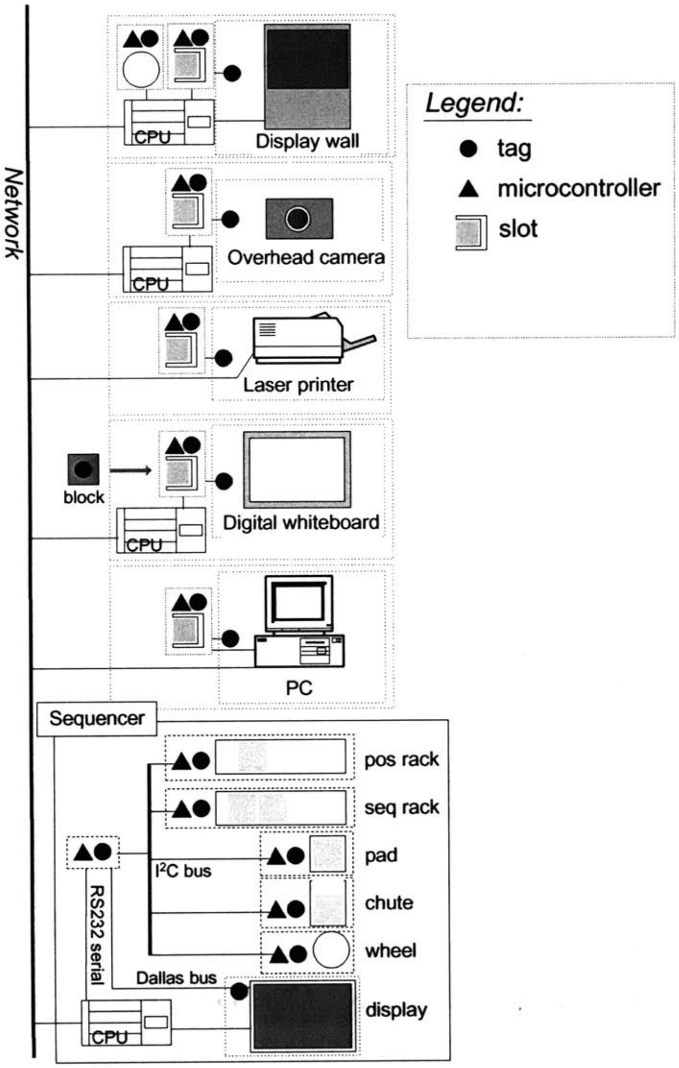


**Figure 4.17: mediaBlocks system, conceptual restructuring**

A potentially relevant analogy for this approach can be found under the X-windows window management system. Here, every window managed by the window manager is issued its own ID, which is guaranteed to remain valid for the duration of the window's existence. Making analogous provisions for the kinds of ID and computation which can be assumed of every TUI element could play a significant role in the evolution of new TUI tools and toolkits. For example, this kind of tagged microcontroller approach might allow resource information and APIs corresponding to individual TUI primitives to be accessed over the network. These and related issues will be considered at further length in Chapter 6.

## 4.9 Continued sequencer experiments

The mediaBlocks work opened the door to a number of new paths for development. Some of these were motivated by the behavior of the "printer" mediaBlock. This block could be placed within the mediaBlocks sequencer's target pad to transfer media to a remote network printer.

This suggested both the physical representation of remote devices, as well as the binding of computational *operations* to mediaBlock-like objects.

Figure 4.18 illustrates a sketch for one possible evolution of the mediaBlocks sequencer. The design was based around the surface of a plasma display, affording a large workspace and significant visual real estate. Here, we explored the use of multiple sliding racks and the physical embodiment of both data and operations.

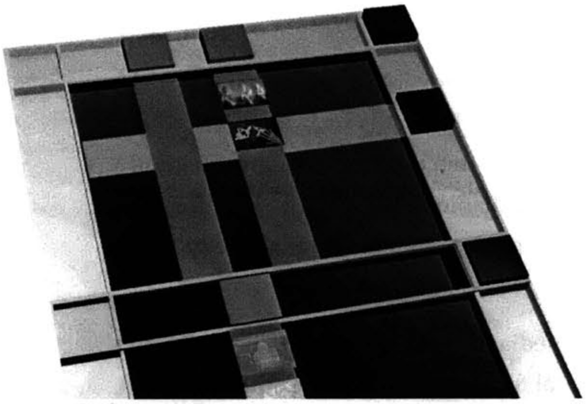


**Figure 4.18: Sketch of prototype evolved mediaBlocks sequencer**

The blocks on the center region of the two racks (a fixed rack on the device's upper margin, and a lower sliding rack) were envisioned as "traditional" mediaBlocks, each representing a list of media elements. The darker-colored blocks were intended to represent digital operations which could be applied to the horizontal racks (or to the whole sequencer), or configured in the vertical racks, with operations expressed as a horizontal "digital shadow."

This design shares some elements with the rack-based tangible query interface approach, and in some respects goes beyond the query interfaces by suggesting a means for binding new operations to racks. However, this revised design was complicated by a combination of conceptual and technical issues.

While the new sequencer seemed promising for (e.g.) filtering operations, it was difficult at the time to arrive at specific compelling operations for development. Also, the physical movement of the racks were associated with relatively weak semantics. At the time, these were conceived primarily as a way for reallocating visual real estate, analogous to the resizing of graphical windows on a GUI. Nonetheless, implementation of this design was begun, and an early semi-functional prototype was completed.

However, I encountered major technical difficulties in implementing the prototype's physical structure. After the departure of a collaborator with woodworking experience, I had turned to fabricating physical structures out of hand-cut foamcore. While this fabrication approach was marginally plausible for structures such as mediaBlocks slots, it was completely unsuitable for more demanding mechanical requirements like the racks. This experience drove me to rethink my approach both to fabrication and to physical design.

## 4.10 Discussion

### 4.10.1 mediaBlocks as containers

This chapter has discussed the role of mediaBlocks as "containers" for digital media, as well as the use of mediaBlock slots for media interchange between various devices. However, similar functions have long been supported by a range of removable media devices. For instance, videotapes and floppy disks are both "physical containers" for electronic media that support media interchange through systems of "physical slots." How do mediaBlocks relate to these well-known technologies?

The comparison will be explored for floppy disks (and other removable digital media), which share the ability to "contain" digital media of various formats. First, it is clear that mediaBlocks and floppy disks are technically quite different. Floppy disks function by taking information offline, storing data within the disk's internal medium. MediaBlocks instead work by transferring information into online networked space, referenced by the internal ID of the mediaBlock object.

It is also interesting to note that mediaBlocks support media of widely varying bandwidth and capacity. For instance, mediaBlocks are equally practical for recordings from digital whiteboards and digital video, even though the characteristic bit rates differ by roughly five orders of magnitude (~100KB vs. ~10GB per hour). As another example, mediaBlocks might be used to "contain" scientific datasets containing many terabytes of content.

From a user interface standpoint, mediaBlocks and floppy disks are also different in character. The contents of floppy disks are accessed indirectly through graphical or textual interaction on a host computer. In contrast, mediaBlock contents may be accessed through physical manipulation of the mediaBlock object itself. For example, inserting a target mediaBlock into a digital whiteboard's slot initiates recording "into" the block. Similarly, moving a host mediaBlock on the media sequencer's position rack allows sequences of images to be navigated.

MediaBlocks' support for physical media exchange does not force a "sneaker-net" ethic upon users. Instead, mediaBlocks offer the simplicity and directness of physically referencing *which data* and *which device* when physical proximity provides a convenient delimiter. Common "reference in absence" tasks such as dispatching jobs to remote printers for later pick-up or delivery may be supported by shortcut controls (e.g., a "print" button on the whiteboard), or by inserting mediaBlocks into TUI or GUI devices providing remote printer access.

MediaBlocks are not intended as a *medium of storage,* but rather as a *mechanism of physical reference and exchange.* In this sense, the use of mediaBlocks again more closely resembles the interactive process of "copy and paste" propagated out into the physical world than the storage medium of floppy disks. Conceptually consistent with the premise of tangible user interface, this is a major distinction that colors the spectrum of mediaBlocks applications.

### *4.10.2 mediaBlocks as conduits*

As another variation, mediaBlock's online aspect suggests their ability to "contain" live, streaming content: in other words, the ability to operate as media "conduits." This streaming media might find its "source" or "sink" in a local or remote media device. For example, the mediaBlocks video demonstrated a "printer" block that routed contents to a networked printer [Ullmer and Ishii 1999]. This block was demonstrated in the context of the media sequencer. It is also easy to imagine its use in combination with media sources. For example, in the case of the whiteboard source, the whiteboard strokes would be streamed to the remotely referenced printer, display, or other device. I also implemented mediaBlocks "containing" both streaming media sources such as RealAudio™ and RealVideo™ media, as well as pairs of mediaBlock conduits which together broadcast and receive streaming video.

At the same time, the conduit functionality of mediaBlocks represents a significant conceptual expansion with its own user interface questions. For instance, if a user wishes to both record and broadcast a whiteboard session, or both display and store a live video stream, how are these aggregate behaviors best accommodated? These and other open questions remain. As a result, expansion of the conduit functionality was left for future work.

### *4.10.3 Distinguishing and selecting between alternate mappings*

As mentioned in §1.3.1 and §2.6.3, a significant issue for token+constraint systems is the manner by which different interpretations are mapped to interpretive constraints, and the approaches by which users can distinguish and select between these alternate mappings. The mediaBlocks sequencer offers an interesting case example.

The sequencer integrates four different interpretive constraints: the sequence and position racks; the target pad; and the erase chute. (The erase chute was intended to erase mediaBlocks reaching its lowermost position, but was not fully implemented.) The racks and chute each were associated with fixed interpretations. In contrast, the mapping of the target pad was moded in nature. When mediaBlocks were present in the sequence rack, the target pad was associated with assignment of the sequence rack's aggregate contents. Alternately, when the sequence rack was empty, the target pad was associated with assignment of the position rack's selected results. The function of these constraint mappings was visually depicted through associated graphical animations. However, the mappings themselves were not visually indicated, and were not rebindable.

These choices suggest the possibility of other design alternatives for distinguishing and selecting between alternate mappings. Several of these are summarized in Figure 4.19. First, the sequencer racks illustrated a single mapping, and the target pad a moded mapping. Second, the mediaBlocks sequencer did not explicitly indicate the mappings of its interpretive constraints. As alternate designs, either graphical or physical indications might have been utilized. The use of graphical indications is used in the tangible query interfaces, and is discussed in §5.2.1.2.

| | |
|---|---|
| **nature of mapping?** | single mapping<br>*e.g., mediaBlocks sequencer racks*<br>moded mapping<br>*e.g., mediaBlocks sequencer target pad* |
| **how is mapping indicated?** | not indicated<br>*e.g., mediaBlocks*<br>graphically indicated<br>*e.g., query interface x/y pads (§5.2.1.2)*<br>physically indicated |
| **is mapping fixed?** | fixed mapping<br>rebindable mapping |

**Figure 4.19: Methods for distinguishing and selecting between alternate mappings of constraints**

Another prospect might be the use of physical tokens that bind specific mappings to interpretive constraints. These tokens could be inserted into special interpretive constraints of their own and remain as persistent physical indicators; or alternately could be touched momentarily, with the resulting binding indicated by a graphical display. I am unaware of token+constraint interfaces that offer such rebindable mappings of interpretations. But to the extent that this could be implemented while maintaining a simple, clear design, this seems a promising direction for continuing development.

### *4.10.4 Relationship between sequencer racks and display*

One of the challenges encountered by the projects of this thesis is the relation between the "object space" where TUI tangibles are manipulated, and the "display space" where the consequences of these manipulations are presented. One of the strong potentials for tangible interfaces, as highlighted by systems such as Underkoffler's Urp urban planning simulator [Underkoffler and Ishii 1999], is the fusion of these two spaces.

As is apparent in Figure 4.6, the mediaBlocks sequencer integrates a series of interpretive constraints around the margins of an embedded flat-panel display. While a looser integration of object and display spaces than realized by Urp, this design partially reflects the chosen interface domain, as well as other design constraints. The motivating task was the sequencing of presentation media, especially images and video. The interface was most directly inspired by the Scrabble™ tile/rack constraint system, along with the tray device from Fitzmaurice et al.'s Bricks system [1995].

As the mediaBlocks task domain centered upon the manipulation of visual media, a dynamic graphical representation of some form was essential. Initially, I hoped to integrate these visuals display into the base footprint of racks, displaying the contents of mediaBlocks through transparent blocks in a fashion following the metaDESK's passive lens [Ullmer and Ishii 1997]. However, given that mediaBlocks usually contain multiple media elements, I had difficulty determining an effective method of display within a block's 5x5cm footprint.

For this reason, I decided to incorporate a display space separate from the racks, while maintaining visual contiguity with both racks and active mediaBlock containers. Back-projected, front-projected, and integrated displays were all explored as potential prospects. While each of these approaches had advantages, a 32cm-diagonal 1280x1024-pixel integrated flat panel display was selected on the basis of resolution, dot pitch, and compactness of integration. These design decisions are discussed further in Chapter 6.

### 4.10.5 Contributions

The mediaBlocks system makes a number of specific research contributions. First, mediaBlock tokens demonstrate the physical embodiment of online information, illustrating new approaches by which network-based information (including live data) can be represented and manipulated within physical devices. The system also demonstrates the use of physical tokens for containing aggregates of multiple digital elements.

The mediaBlocks system realizes a novel "copy and paste" approach by which online digital information can be transferred between multiple physical media devices. Further, mediaBlocks serve as physical embodiments of both digital "container" and "control" behaviors, demonstrating a novel "multiple inheritance" interaction approach. Finally, the mediaBlocks system unites these contributions into an integrated system for the capture, manipulation, and presentation of digital media.

The "copy&paste" and sequencer aspects of the mediaBlocks system build upon each other, yielding a combined functionality that is more than the sum of these individual parts. The slot-based "copy&paste" function provides a mechanism for binding mediaBlocks to data and moving this data between physical devices, but provides no mechanism for actively manipulating this content. The sequencer provides a mechanism for actively operating upon mediaBlock contents, but depends upon external support for providing meaningful sources and destinations for mediaBlock bindings. Taken together, the resulting system supports physically situated tasks that are fundamentally difficult to achieve with desktop computers, while complementing these with digital behaviors that were previously the sole province of desktop machines.

### 4.10.6 Limitations

The mediaBlocks system has several kinds of limitations. First, the sequencer's "indexing" and "sequence" operators are relatively "weak" digital operations, and do not inherently suggest how other, more powerful operations might be derived. Secondly, the "contents" of mediaBlocks are visible only when blocks are located within an interpretive constraint (or through manual annotations on mediaBlock labels). While this is also true with floppy disks or flash memory, mediaBlocks are intended for use in combinations of 5 to 10 blocks, each with rapidly evolving bindings, making the dynamic indication of contents more important.

Another class of limitations relates to assumptions implicit in the mediaBlocks approach.
For one, the mediaBlocks approach assumes that many media devices throughout the physical

environment are interconnected with high-speed network links. While at present this assumption is only partially the case, there is a widespread consensus it will become increasingly true. Secondly, the interoperability of mediaBlocks slots depends upon a level of physical, electronic, and software compatibility and common infrastructure that might in practice be hard to achieve. Still, a partial workaround is visible in the mediaBlock system's integration with a standard network printer. This demonstrates that slots and other interpretive constraints can be added to existing network devices, with slots acting as kinds of gateways or proxies that relay commands over the network to the "host devices."

A final kind of limitation encountered in the design of mediaBlocks related to the physical fabrication of the sequencer and other mediaBlocks devices. In the absence of the collaborator with longstanding machining and woodworking skills, I found it very difficult to extend or replicate the mediaBlocks design. While at some level this challenge could be addressed by finding a new collaborator or cultivating the requisite skills, it seemed likely to be a common challenge and perhaps barrier for a large subset of the community of user interface designers. In addition to seeking skills in physical fabrication, I also sought to identify approaches that might allow the description and fabrication of tangible interfaces through software, potentially bringing some of the powers for creating electronically instrumented physical objects that modern desktop printers have brought to the printed page. This effort is reflected both in the design and construction of the second major thesis project, and in the approaches described in Appendix A.

## chapter 5

# tangible query interfaces

# 5 Tangible query interfaces

Growing out of the mediaBlocks research, the tangible query interfaces project is the second major system of the thesis. These interfaces further develop the token+constraint approach, extending the range and power of what can be expressed with tangible interfaces. In particular, the project realizes several TUIs for physically expressing queries to database systems.

Tangible query interfaces use several kinds of physical tokens to represent database parameters. Placing these tokens onto "query racks" expresses queries composed of the corresponding parameters, and also invokes visualizations of the associated parameter distributions. Physical manipulation of these tokens is then used to modify parameter value thresholds; express Boolean relationships; and configure the system's visualizations (Figure 5.1).

**Figure 5.1: "Parameter wheels;" "parameter bars" with adjacency-based Boolean relations**
*note: contrast adjusted on pictures throughout chapter to compensate for wide dynamic range*

The mediaBlocks system was the first tangible interface to support the physical representation and manipulation of aggregates of digital information. However, the mediaBlocks sequencer supports only the simplest of computational operations: indexing, concatenation, and assignment. It is desirable to extend tangible interfaces to support more complex operations that leverage computers' capabilities for processing large aggregates of information.

Where the mediaBlocks sequencer expressed individual operations, tangible query interfaces support the expression and manipulation of higher order declarative expressions. Moreover, where mediaBlocks are used as "containers" for specific information elements, the parameter tokens of tangible query interfaces physically describe relationships that hold across aggregates of information. In a way, mediaBlocks can be thought of as "nouns" that reference specific instances of information, while parameter tokens serve as kinds of descriptive "adjectives." This design choice significantly increases the scalability of the query interfaces approach.

Two major variations of tangible query interfaces have been implemented. The first approach is based upon tokens called "parameter wheels." These wheels are small cylindrical objects embedded with RFID tags and faced with cardstock labels. The second approach, "parameter bars," uses tokens embedded with adjustable sliders and active displays.

The chapter begins with an example of the query interfaces functionality. This functionality is then considered in the context of the thesis' token+constraint approach. The chapter continues with a comparison of the parameter wheel and parameter bar approaches, followed by a review of several earlier design iterations. Finally, the chapter discusses the implementation of tangible query interfaces, and considers several usage contexts in which they might be usefully deployed. Chapter 6 continues with a user study evaluating the parameter wheels approach.

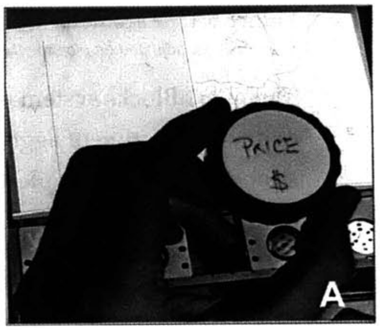
## 5.1 Example uses

To introduce the functionality of the query interfaces, I will present example interactions with the parameter wheels and parameter bars. Both of these examples are fully implemented; still images from a video illustration are included within the text.

Both examples develop a real estate query application. This domain was chosen for several reasons. First, this domain was used by the "Dynamic Homefinder" application of the dynamic queries technique [Williamson and Shneiderman 1992], making it convenient for comparing the graphical and tangible interfaces. Second, this domain seemed easily accessible for subjects of the user study. Third, it was easy to imagine how such a system might see use in a group interaction context, with (e.g.) a real estate agent working together with a couple to interactively identify candidate homes.

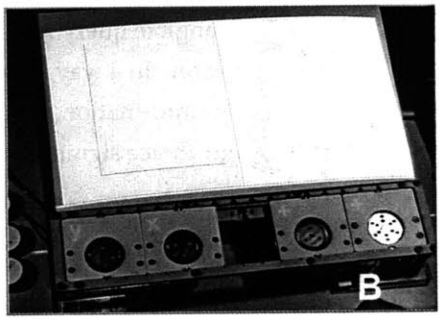
### *5.1.1 Parameter wheels*

In the first example, parameter wheels are used to explore candidate homes in a real estate database. Nine parameter wheels are present, each representing fields of the database (A). Six wheels represent continuous parameters like price and acreage (hectares). Three wheels represent discrete parameters like building types and features.



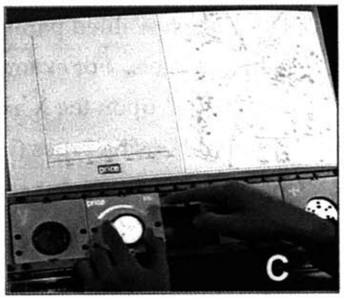
These parameter wheels are used within a "query rack" made up of a series of "query pads," each with a round receptacle for a parameter wheel (B). Placing a wheel upon one of these pads expresses the associated parameter as part of the active query. A display surface is located adjacent to the query rack. Two visualizations – geographical and scatterplot (starfield) views – appear on this surface.

An embedded projector is used to illuminate both the display surface and the query rack, including its four embedded query pads. The left two query pads are proximally, visually, and functionally associated with the 'Y' and 'X' axes of the scatterplot (B). The right pads offer additional query space without direct scatterplot mappings.

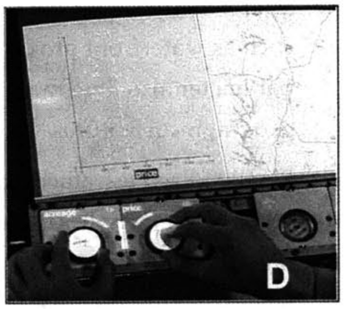


An example interaction might begin by picking up the "price" parameter wheel and placing it upon the "X axis" query pad (C). Correspondingly, the "price" label and value range are illuminated on the query pad surrounding the wheel, and a 1D plot of price appears on the scatterplot.

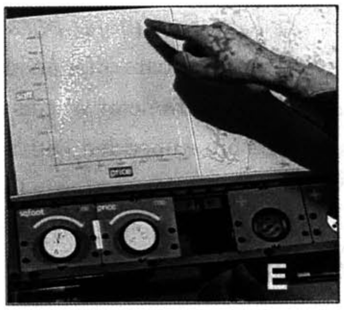
Moreover, the locations of all homes meeting this parameter criteria are displayed on the geographical view. The "price" wheel initially specifies the parameter range spanning from the least expensive homes to the middle-cost homes. The upper bound can be adjusted by rotating the wheel within the query pad (C). The query pad, scatterplot, and geographical visualizations all update correspondingly.



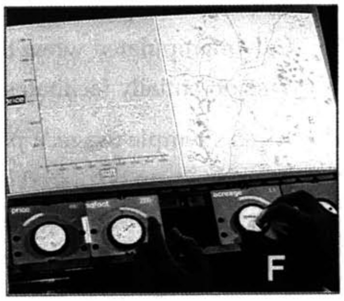
It is also useful to add a second parameter criteria to the query. Placing the "acreage" wheel upon the "Y axis" query pad, the Y axis of the scatterplot is updated accordingly, yielding a 2D plot of acreage against price. The geographical view is also instantly updated. To identify (e.g.) low-priced houses sited upon relatively large properties, the two wheels can be manipulated simultaneously using two hands (D). The scatterplot indicates available prospects, while the geographical view indicates their corresponding locations (in this case, on the periphery of the city).



In some cases, it may be desirable to spot trends in the data. Such patterns are not immediately visible with the price and acreage pairing. However, replacing the acreage token with the "square footage" wheel, a distinct correlation becomes visible in the scatterplot view (E). This is even more apparent by swapping the plot's axes (moving price to the Y axis). This can be realized by switching the placement of the two parameter wheels. Parameter wheels remain persistently bound to their associated value ranges when moved to or from the query rack, thus easing this change of view.

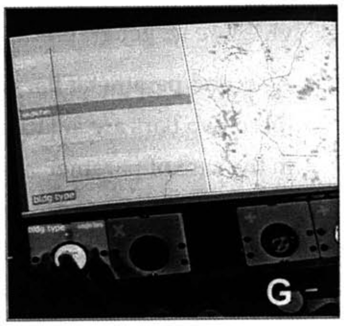


To view a third or fourth parameter, the associated wheels may be placed on the query rack's rightmost pads. For example, the acreage wheel can be returned to the query rack alongside the price and square footage wheels (F). In this case, the acreage token is not explicitly represented on the scatterplot view. However, its impact is visible both through the corresponding changes in the geographical view, and also through changes in the highlighted properties within the scatterplot view's selected region.



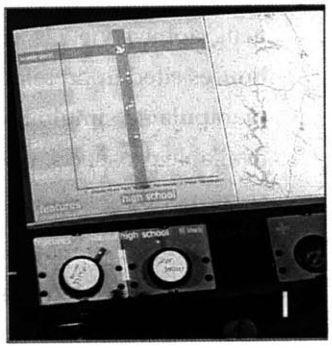
Several other continuous parameters are provided to express distance from locations in the geographical view. In the prototype system, these locations are fixed; they could eventually be manipulable through a touch-sensing display surface.

Several wheels are also associated with discrete parameters. One such wheel is associated with different types of buildings (G). For instance, turning the wheel to select "patio homes" (sometimes preferred by retired people) shows clustering in a certain area of the city. Similarly, selecting "mobile homes" exposes locations on the city's periphery.



The discrete-valued parameter wheels can easily be combined with continuous-valued wheels. For example, placing the "building type" wheel on the Y axis and "price" upon the X axis shows distinct clusterings of prices associated with different housing types (H). When price is replaced with acreage or square footage, quite different clusterings are visible .

A second discrete-valued parameter wheel selects for area high schools. As expected, this parameter shows strong clustering corresponding to different school districts. A final discrete-valued wheel selects for building "features" such as waterfront proximity, the presence of a pool or porch, etc. This parameter also illustrates visual clustering around lakes and other geographical features.



Finally, discrete-valued parameter wheels may also be combined with each other. For example, (I) illustrates the intersection of building features with high school districts. This indicates (e.g.) the high school districts in which waterfront homes are available.

### 5.1.2 Parameter bars

Parameter bars offer an alternative approach for physically expressing queries. Where parameter wheels are faced with a passive mat board label, parameter bars are embedded with active LCD and LED displays (Figure 5.1b, Figure 5.5). These displays indicate the identity and a value distribution histogram for the active parameter, and facilitate dynamic binding of bars to new parameters.

The embedded double sliders allow both the upper and lower bounds of a target parameter range to be modified. Additionally, the combination of embedded displays and manipulators allow parameter wheels to be meaningfully manipulated when they are not physically present on the query rack. This potentially facilitates use of the system in group interaction contexts.

As an example usage, a parameter bar representing the price of real estate properties can be placed onto a query rack. As with the parameter wheels, corresponding scatterplot and geographical results are displayed. Unlike the parameter wheels, both the lower and upper bounds of the price distribution can now be controlled. This supports the identification of patterns that were not previously visible; e.g., geographical clusterings of high-priced homes (J).

A second parameter bar can again be added to the query rack. When these two bars are adjacent, a Boolean "AND" operation is applied, as in the case of parameter wheels (K). However, when the parameter bars are spatially separated on the rack (which is detented to supported stable positioning and haptic feedback), an "OR" operation is instead applied. This "OR" operation has special value for comparing the distributions of different parameters. For example, when an "OR" relation between high-priced and high-acreage homes is displayed, the original "price" clustering is distinctly visible alongside the distribution associated with the acreage parameter. Pushing the two tokens together again yields the "AND" intersection. More complex logical relationships can also be physically expressed, complemented by the corresponding visualization of parameter distribution (L).

Where this visualization of value distributions is meaningful in the real estate domain, it takes on special value for other kinds of datasets. For example, the parameter bars were also applied to a database of mutual funds. In this domain, it is valuable to (e.g.) compare the one-year and ten-year returns for a group of funds. At first, it is desirable to view both of these distributions simultaneously, which is facilitated by the "OR" relation and corresponding visualizations. As particular values of interest are identified, it is then desirable to express an "AND" conjunction, and to concentrate on the relationship between this intersection and additional variables (e.g., risk assessment).

## 5.2 Roles of tokens and constraints

As discussed earlier, tangible query interfaces are built around systems of tokens and constraints. The tokens – parameter wheels and bars – each represent the parameters that may be manipulated by users. The constraints – query racks and (in the case of parameter wheels) query pads – embody the interactive workspace in which the tokens are manipulated to express, manipulate, and visualize queries and query results.

Like other token+constraint systems, interaction with the tangible query interfaces has two phases: *associate* and *manipulate*. In the "associate" phase, parameter tokens are placed upon the query racks, thus expressing query operations and invoking views of specific database parameters. In the "manipulate" phase, the parameter tokens are rotated or translated within the query rack, expressing assignment and Boolean operations. This is illustrated in Figure 5.2.

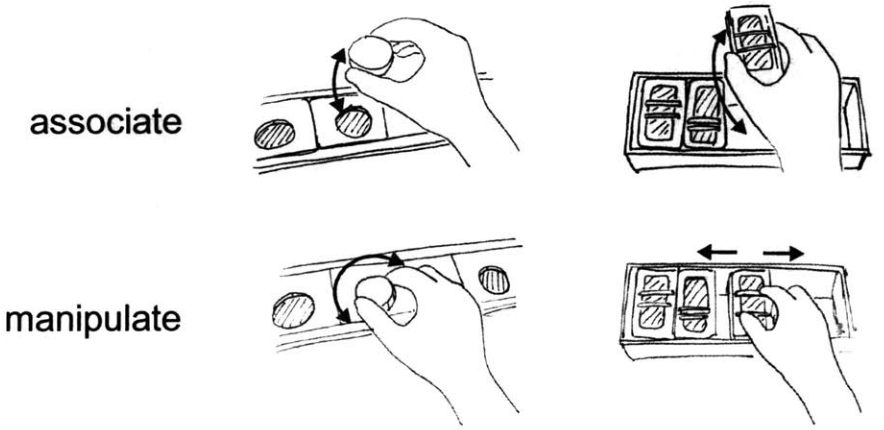


**Figure 5.2: Associate and manipulate phases of interaction with parameter wheels and bars**

### 5.2.1 *Wheel, bar, rack, and pad mappings*

Four kinds of operations are associated with the query rack constraints: query, view, assignment, and Boolean operators. These corresponded to the following physical/digital mappings:

1) physical presence → query parameter assertion
2) physical placement → view selection
3) physical rotation → parameter value selection + assignment
4) physical adjacency → Boolean operation

#### 5.2.1.1 *Query operation*

The "query" and "view" operators are both invoked during the "associate" phase of interaction. The act of placing a token upon a query rack expresses a "select… where…" query operation in SQL (the most common database query language). The "where" clause takes the associated parameter as an operand. For example, if a "price" token is placed on a query rack that is associated with a real estate database, a query like:

```
select bldg_id where (price > [price.min] AND price < [price.max])
```

…might be evaluated. If multiple tokens are present upon the rack, all of the associated parameters are used as "where" operands.

#### 5.2.1.2 *View operation*

For the "view" operator, somewhat different mappings are used by the parameter wheel and bar interfaces. In the parameter wheels interface, the 'X' and 'Y' scatterplot axes are associated with two specific pads of the query rack. For example, in Figure 5.3, a parameter wheel is sitting on the 'X' pad, while the 'Y' pad is empty. Here, placing a parameter wheel on the 'X' or 'Y' pad invokes the "view" operator for the corresponding scatterplot axis. In addition, this action also invokes the "query" operator as well as the "view" operator for the geographical plot. Thus, three distinct operations are triggered by the same physical action.

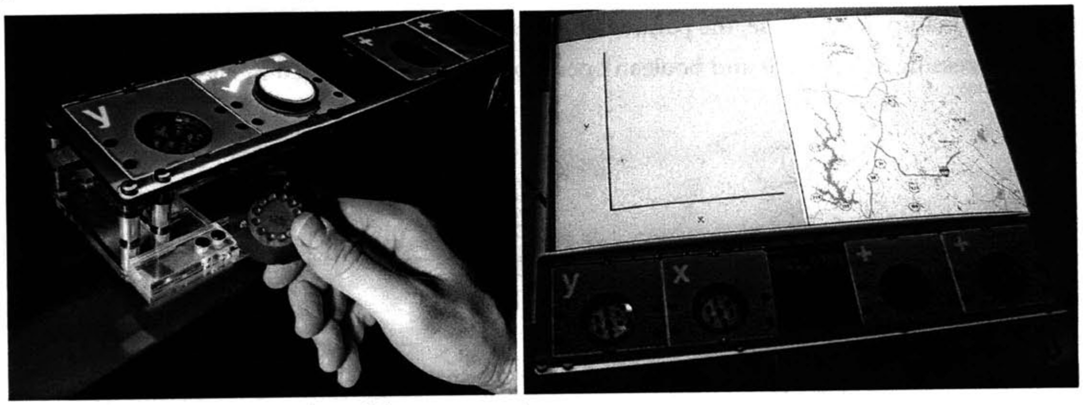


**Figure 5.3: Parameter wheels and the query rack**

In the parameter bars interface, the first parameter bar (ordered from left to right) specifies the y-axis parameter of a visualization scatterplot (given its closer physical proximity to the y axis). The next parameter bar specifies the x-axis parameter.

#### *5.2.1.3 Assignment operation*

The assignment and Boolean operators are both invoked during the "manipulate" phase of interaction. The assignment operator is specific to the parameter wheel interaction. When a parameter wheel is turned (while seated within a query pad constraint), the threshold value associated with the wheel is correspondingly altered and assigned to the associated parameter.

While the range sliders on the parameter bars also perform a similar role, they are permanently fixtured to the parameter bars, and their functionality is not specific to tangible interfaces. (In thesis terms, the sliders do not support an "associate" phase of interaction.) Thus, while the movement of these sliders is also "constrained," parameter bar sliders do not make use of a token/constraint style of interaction.

#### *5.2.1.4 Boolean operation*

As discussed in §5.1.2, the parameter bars interface maps token adjacency to Boolean operations. Token adjacency is mapped to the Boolean 'AND' operator, while non-adjacency maps to the 'OR' operator. This constraint/operator mapping has the distinguishing feature that it enables the "sweeping" or "bulldozing" action identified within [Fitzmaurice et al. 1995], where multiple physical artifacts can be pushed en masse with the hand. Since the Boolean mapping transforms each change in adjacency into a digital operation potentially affecting hundreds or thousands of elements, this mapping allows fairly powerful digital operations involving multiple tokens to be substantially altered through one physical gesture.

It is also worth noting that several other applications of the Boolean adjacency-operator were developed by the query interfaces. First, in the parameter wheel interface, each query pad can be moved within the query rack (and correspondingly sensed) to express adjacency relationships. In early interface iterations, this was mapped to Boolean operations. Second, several query interface prototypes have allowed multiple query racks to be combined in Boolean relationships using the adjacency operator. Both of these design alternatives will be considered further in the chapter's discussion.

### *5.2.2 Other token+constraint mappings*

The query interfaces illustrate the mapping of physical token+constraint relationships to four different digital operators. As referenced earlier, these are specific instances of a broader family of possible token+constraint mappings, which are again summarized in Figure 5.4.

| **Physical relationships** | **Interaction Event** | **Digital interpretations** |
|---|---|---|
| Presence | Add/Remove | Logical assertion; activation; binding |
| Position | Translate/Rotate | Geometric; indexing; scalar |
| Sequence | Order change | Sequencing; query ordering |
| Proximity | Prox. change | Relationship strength (e.g., fuzzy set) |
| ├Connection | Connect/Discon. | Logical flow; scope of influence |
| └Adjacency | Adjacent/NAdj. | Booleans; axes; other paired relations |

**Figure 5.4: Grammars for mapping token/constraint compositions to digital interpretations**

In this figure, the "presence" relationship is usually expressed in the "associate" phase of interaction, while other relationships are often expressed in the "manipulate" phase. The darkly shaded elements are actively employed in the query interfaces. The lightly shaded mappings were also explored, and are considered further in §5.3.3 and the discussion.

As referenced in §5.2.1.4, adjacency-based relationships have several special properties. A key feature of tangible interfaces is their multiple physical loci of control (in contrast to the single pointer of traditional GUIs). The adjacency-based Boolean operation makes strong use of this aspect, allowing one or more users to quickly recluster parameters for exploratory queries using two hands and ten fingers. This contrasts with mappings like mediaBlocks' "indexing" and "sequencing" operations, which often may be satisfied with a single physical point of control.

## 5.3 Token details

### *5.3.1 Parameter wheels*

Each parameter wheel physically represents a database parameter (i.e., a field from a database table), together with the range of values this parameter can take on. This can be a continuous range, as with parameters like "price" and "acreage;" or a series of discrete parameters, as with parameters like "building type" and "building features."

Each parameter wheel identifies both a particular parameter instance, as well as a value selection associated with the wheel. Currently, parameter wheels for discrete parameters reference a single value. Continuous parameters currently reference a single-ended range of values, either with respect to a fixed upper or lower bound.

### *5.3.2 Parameter bars*

Parameter bars are discrete, physically embodied extensions of the double-ended "range sliders" introduced in [Ahlberg and Shneiderman 1994]. Each parameter bar has two physical slider levers, which travel over an embedded backlit LCD display. This embedded display shows the parameter name; a gridded axis of the parameter's value range; a histogram of the parameter's value distribution; and a visual confirmation of the current selection.

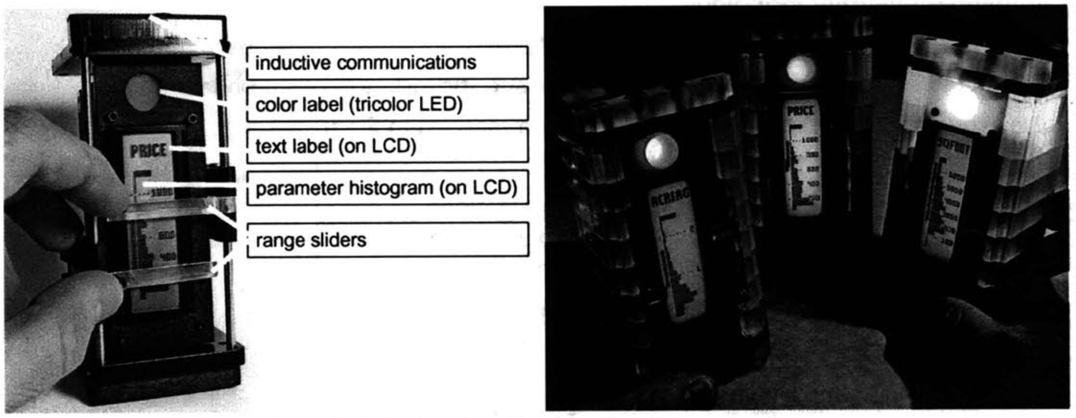


**Figure 5.5a,b: Parameter bars (original and revised designs)**

The range sliders of parameter bars are used to select continuous ranges of associated values. These ranges are usually of numerical values (as in the case of the two "return" parameters). They can also represent other kinds of properties that span a continuous range (e.g., risk ratings from "high" to "low").

Parameter bars are also embedded with a tricolor (full-spectrum) LED. This is used to associate one of eight color labels with each parameter (Figures 5.1, 5.5, 5.6), and serves several purposes. First, these same colors are used in query visualizations, aiding the visual association between parameter bars and their associated data. Secondly, similar to the color labels for files in Macintosh™ Finder, they aid discrimination between multiple parameter bars. This takes on added significance in multi-user contexts, as parameter bars' LCD visuals may not be visible from a distance.

In order to associate parameter bars with new bindings, a monitor-based binding gateway has been implemented. When a query dataset is active, a series of associated parameters is displayed next to a series of binding points on the bottom margin of a computer display (Figure 5.6). These binding points are annotated with both textual and color labels. When a parameter bar is placed next to one of these binding points, its LCD and LED displays are illuminated with the corresponding labels, and its internal state is changed accordingly.

**Figure 5.6: Parameter bars and binding gateway**

However, it should be noted that in some respects the parameter bars are problematic as examples of TUI tokens. Some of these issues are explored in §5.7.2.

### *5.3.3 Dataset containers*

The combination of parameter tokens and query racks allows users to physically express the parameters, value ranges, and Boolean relations forming a multi-parameter SQL query. It is also important to provide a way to specify the dataset upon which a query is to be evaluated – i.e., to identify the desired database and tables out of multiple possible alternatives. Additionally, users may wish to capture and compare the results of multiple queries, or use prior results as the basis for new queries. This takes on a special significance for tangible interfaces, since their physical configurations are less easily recalled than those of textual or graphical interfaces.

The "dataset containers" support these functions. These objects are embedded with LCD displays indicating their contents (sometimes proxied with paper labels), and operate upon two "binding pads" located beneath the interface's associated display monitor (Figure 5.7).

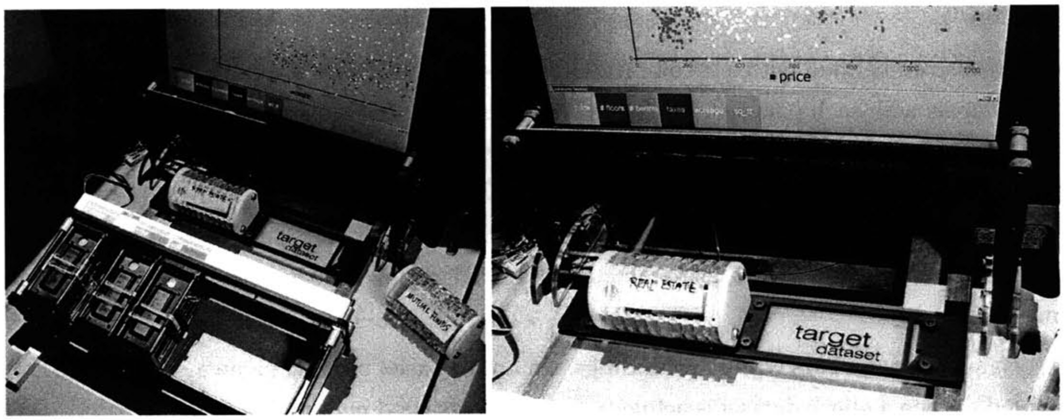


**Figure 5.7a,b: Dataset containers: in context, closer view (both with older generation of system)**

On placing a dataset container onto the "source pad," several windows appear on the interface's display surface (by design, an LCD display, proxied with paper labels in Figure 5.7). First, an array of parameter bindings is displayed. Second, one or more visualizations of query results are displayed. Removing the dataset container removes these windows from the display. If the display is shared with a graphical interface, this provides a physical mechanism for managing screen real estate.

Alternately, placing a dataset container onto the "target pad" causes the current database, tables, parameter ranges, and currently selected data values to be bound to this container. If the container held prior contents, these bindings are cleared. A visual indication of these contents is shown on the container's display.

## 5.4 Query result visualization

The primary interest in query interfaces from the thesis perspective lies in the physical representation and manipulation of database queries. For this activity to be meaningful, a way to represent the results of these queries must also be realized. A hybrid approach is currently used, with queries expressed through the above tangible interface, and query results visualized upon adjacent graphical surfaces. Two different visualizations have been implemented. While these are similar to conventional visualization approaches, their application to the example real estate dataset will be briefly considered.

First, a scatterplot visualization has been applied to both mutual fund and real estate data (Figure 5.8). The selected parameter bar ranges are visually highlighted on the scatterplot. Figure 5.8 illustrates the X and Y parameters reflecting an "OR" relation. If an "AND" relation were present, the highlighted datapoints would be limited to the center box of the X and Y range selections. To view other parameters, parameter bars can be exchanged on the query racks, or a multivariate visualization could be implemented.

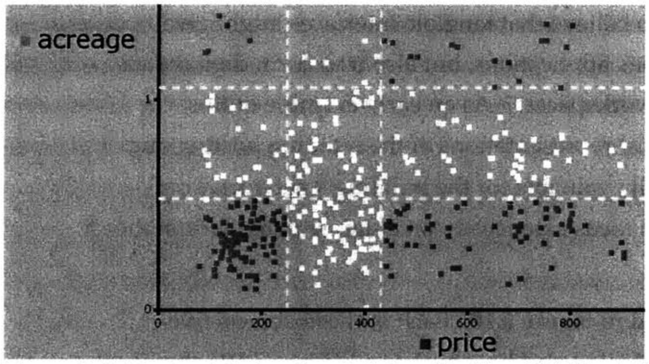


**Figure 5.8: Real estate scatterplot visualization, "OR" relation**

In addition to the scatterplot display, a geographical display has been implemented for the real estate dataset (following the "Homefinder" of [Ahlberg and Shneiderman 1994]) (Figure 5.9). For datapoints that satisfy the query, the system should help users visualize the contributions of different parameters (especially those linked by "OR" relations). Toward this, each selected datapoint is surrounded with a small segmented box. Each segment corresponds to a parameter bar present on the query racks. If a parameter holds true for a given datapoint, the corresponding segment is filled with the parameter's characteristic color.

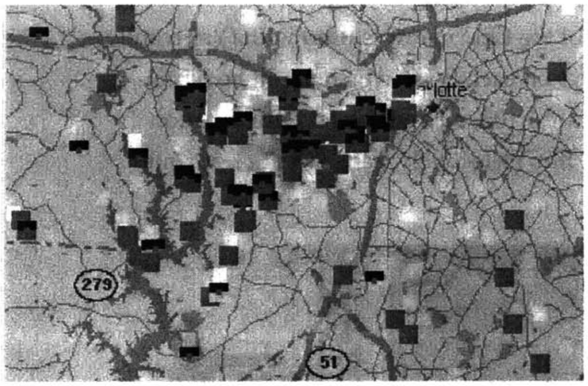


**Figure 5.9: Geographical visualization (two parameter)**

Another interesting possibility would be to display query results into existing physical world contexts. For example, the TouchCounters system embedded physical storage containers with sensors and local LED displays [Yarin and Ishii 1999]. Shelves with large numbers of these containers were used as a "distributed visualization" for the containers' usage history. Tangible query interfaces could provide a strong approach for querying such a system, with results displayed directly onto the containers. As another idea, our interfaces could query census information in combination with the Urp urban planning simulator [Underkoffler and Ishii 1999], with query results displayed directly onto Urp's interactive graphical workbench.

## 5.5 Early iterations

### *5.5.1 Early design concepts*

Tangible query interfaces originated as a component of a system called "Strata" that was developed in the middle stages of the thesis. The Strata project was inspired by my early experiences with the laser cutter, which led me to believe that tangible interfaces might give physical form not only to the "software" of various applications, but also transform digital data itself into medium- and large-scale "interactive workspaces." As an early example of this approach, I worked to develop interactive physical representations of the building hosting our laboratory. I believed these might prove especially valuable for the building's network, computing, and facilities management groups, which focus extensively on issues situated within the physical context of the building.

Several early Strata design sketches are illustrated in Figure 5.10. Each of these depicts five-layer models representing the five floors of the Media Lab's host building. Figure 5.10a shows the earliest laser-cut representation of the Media Lab building. These physical models were constructed at a variety of scales and distortions of the inter-floor distance (for providing visibility and physical access to the different surfaces).

The latter three sub-figures present graphical sketches of different concepts for this interaction. Figure 5.10b illustrates an approach with different "pegs" (perhaps containing actively addressable LEDs) that could be placed in different locations in the building to represent spots of interest. The slider elements on the model's margins were imagined to control different parameterized representations of the building (absolute and relative time being among the most obvious examples). Figure 5.10c represented layers composed of the LED matrices used with [Yarin and Ishii 1999]. Here, the concept was to use paper overlays on each floor, which might be illuminated from beneath with the multi-color LEDs. In Figure 5.10d, a larger-scale representation was considered, using overhead projection and PDLC film (selectively transparent or opaque) to allow projection on each layer of the model.

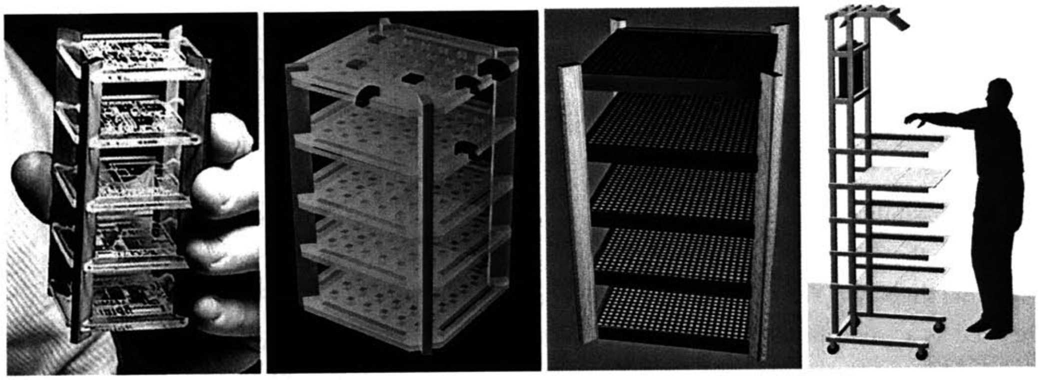


**Figure 5.10a,b,c,d: Early Strata design sketches**

Each of these approaches faced the challenge of how to represent information and operations that did not map into the inherent spatial "real estate" of the floors of building. The slider tokens of Figure 5.10b represented one early attempt to integrate such non-geometric information directly into the building model. However, at least with the visual/physical language used within Figure 5.10b, the close juxtaposition of different interpretations for the model's spatial axes seemed potentially problematic.

In response, a new generation of designs sought to develop separate physical representations for the building itself and building-related queries. For example, Figure 5.11 illustrates the combination of a five-layer building model with a rack-based device for expressing queries to building-related databases. In Figure 5.11a, this query space draws closely from the media-Blocks sequencer design, with rectangular blocks representing query fragments manipulated within three fixed racks.

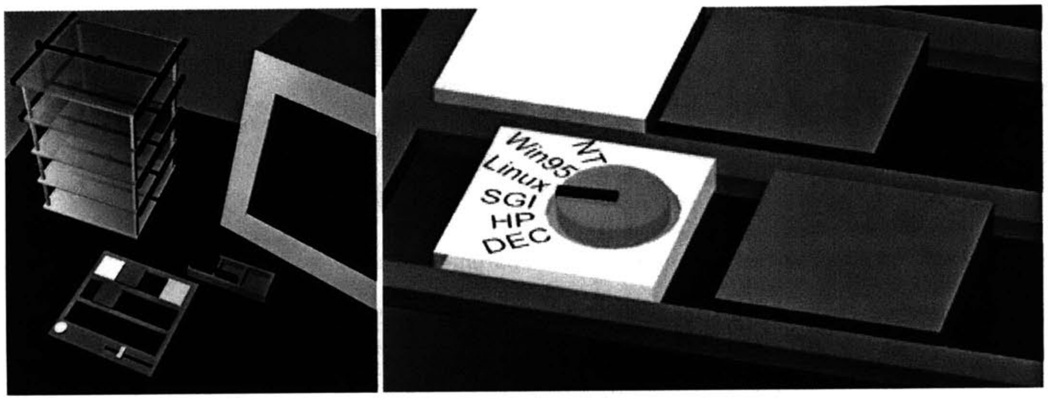


**Figure 5.11a,b: Early Strata sketches, w/ query rack, first-generation parameter wheel**

However, if complex queries were to be expressed with such a device, binding each query variation to its own physical block seemed likely to result in an unreasonably large number of physical objects. As an alternative, it seemed useful to consider approaches for representing multiple possible queries within a single physical artifact. The result was a first version of the "parameter wheel:" a rotary selector embedded within a physical block, used to select and represent one of multiple values within a parameterized query (Figure 5.11a).

While this seemed a promising start, the variation of Figure 5.11a sparked several concerns. First, when approached from the perspective of discrete parameters, the visuals of Figure 5.11a suggested some difficulty in representing even moderate numbers of different parameters. Additionally, the wheel itself was initially envisioned as attached to its rectangular base, which from an aesthetic standpoint seemed somewhat suspect.

Several second-generation variations upon parameter wheels are illustrated in Figure 5.12. Here, the wheels rotated about a horizontal (rather than vertical) access, and were bound either to a single parameter, or to a cluster of complementary parameters. For instance, the "activity" wheel-cluster of Figure 5.12a is intended to represent a query for computers which have not been "active" in greater than ~5 hours (where "active" might constitute mouse or keyboard activity at the console).

In the "activity" example of Figure 5.12a, the physical token integrates three different parameter wheels. The first expresses a temporal selection ranging between minute, hour, day, week, month, and year. The second represents a continuously variable multiplier, perhaps ranging from 1 to 10. The third represents an inequality operator, allowing selection between "less than," "equal to," "not equal to," and "greater than." These subcomponents could either be permanently integrated, or realized as snap-together pluggable elements.

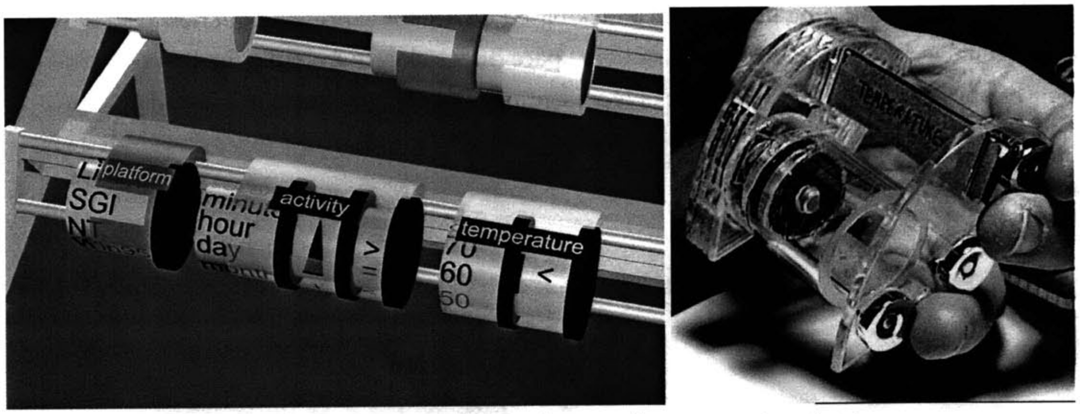


**Figure 5.12: Sketch, physical prototype of second-generation parameter wheel**

This second-generation variation of parameter wheels seemed to suggest their potential for supporting exploratory queries. For instance, the "platform" and "activity" wheels were sketched (and later physically prototyped) in consultation with our lab's network managers, while a "temperature" wheel was created in consultation with the lab's facilities managers. While conceived for use within different contexts, these three wheels serendipitously came together in the 3D modeling program used for sketching. As pictured in Figure 5.12a, they express the query "select all of the building's SGI computers that have not been active in more than a few hours, and are in rooms with temperature less than ~65°F." While queries involving the physical and computing environments had not been intended for use together, this query held a certain logic – especially with variants involving inactivity over weeks or months in rooms with high temperature.

Where the original intention of these parameter wheels centered on the expression of standing queries for triggering alarm conditions involving networking and facilities data, the prospective ease of physical manipulability and serendipitous discovery illustrated by this design seemed to support exploratory querying scenarios. Several physical prototypes of these parameter wheel variants were fabricated with the laser cutter (Figure 5.12b). Here, multiple wheels were envisioned as coupling together using the magnetic purse snaps from Yarin's early Touch-Counters prototypes [Yarin and Ishii 1999].

However, the horizontal-axis cylinders seemed subject to visually occluding most the non-selected values. (Again, this was biased by the specific design sketches of Figure 5.12; in retrospect, other alternatives come to mind.) With the Strata project oriented more toward building than query representations, the first-generation wheel design seemed more amenable at the time for integration into building models.

### 5.5.2 Integration within Strata/ML prototype

Two Strata interfaces incorporating parameter wheel query mechanisms were designed and implemented. Both approaches were built around representations of the Media Laboratory composed of three display layers. These layers were intended to display different physical and logical layers of the laboratory, including different floors, different infrastructure layers, different temporal views, and so forth. Both variants used cylindrical RFID-embedded parameter wheels. Both also used RFID-tagged clear acrylic layers as screen overlays, each visually etched and electronically coded to identify a floor of the building.

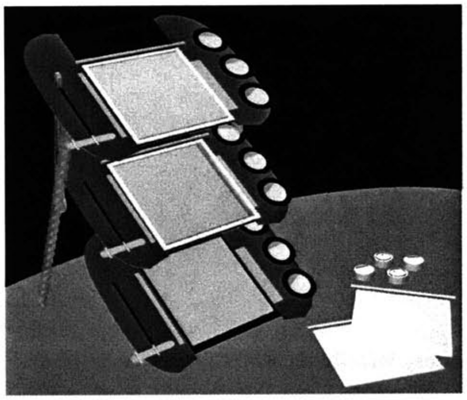


**Figure 5.13: Strata design with query pads on multiple display layers**

In the first interface variation, query pads were mounted on each of the interface's three physical layers, allowing parameter wheels to define and query these layers within the display structure (Figure 5.13). A physical prototype of this basic approach was implemented, including the integration of one of the three display layers.

The token receptacles on each layer of the structure provided the potential for layer-by-layer content control. A variety of approaches were explored for (e.g.) displaying different temporal representations on different physical layers (e.g., present-time on the top layer; one month ago on the second layer; two months ago on the third layer), then quickly shifting to a representation of different content on each layer (different infrastructure layers, floor layers, etc.).

However, this approach seemed liable to trading off control and expressiveness at the expense of complexity. For a variety of reasons (including consultation with potential users), I believed that the most common system usage would be to express a single query over multiple floors of the building. Here, the presence of many alternative locations for expressing a query, without a clear affordance for how this query would be interpreted (or without the requirement for manually replicating a query layer-by-layer) seemed to indicate a problematic design. Given this and other issues, I decided to return to a single unified space for expressing queries.

In this second prototype, three display layers were mounted within "drawer slides" (Figure 5.14). This allowed the display layers to be grouped vertically (nearest the natural building configuration), or tiered in different horizontal configurations to support ready viewing. The lowest layer supported pen-based input, used for selecting spatial query results and entering new information using traditional graphical interface techniques. The pen's location was intended to be reflected with a shadow-cursor on other display layers. Pen-based input was planned for integration on all three layers, but display-only flat panel screens were eventually used for the upper layers due to angle-of-view limitations on the stylus-sensitive display (a Wacom PL400).

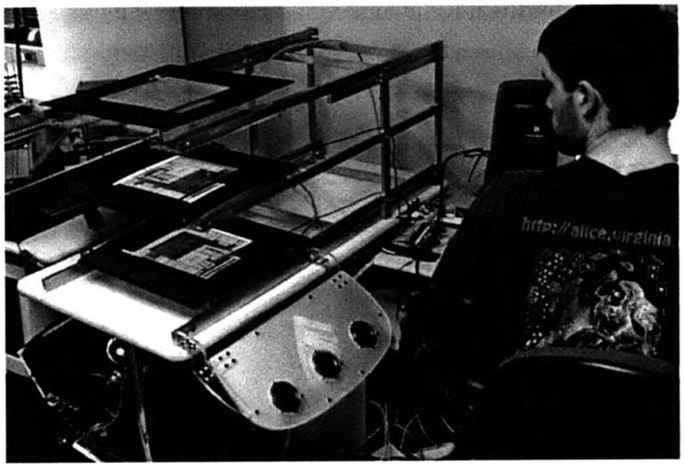
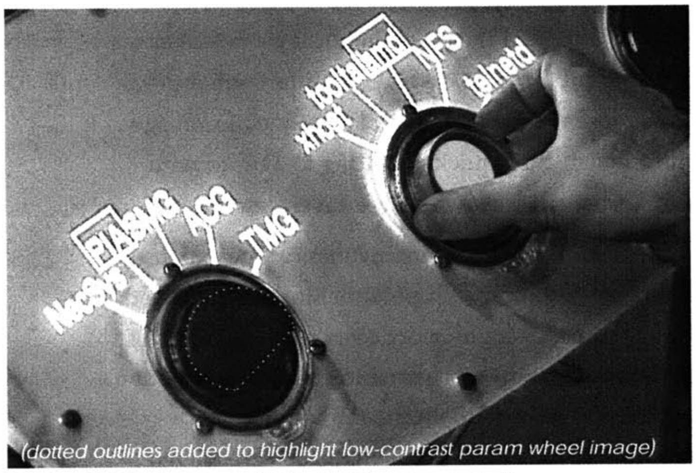


**Figure 5.14: Second implemented Strata/ML prototype, with single query interface**

The prototype's query interface contained three "query pads," each statically fixtured to a central back-projected control panel (see Figure 5.14 and Figure 5.15). Several associated parameter wheels were permanently bound to parameters relating to building network security. Parameter wheel results were joined via an implicit AND, with the results displayed spatially as machine and network hardware locations within a building floor plan.



*(dotted outlines added to highlight low-contrast param wheel image)*

**Figure 5.15: Close-up of Strata/ML query interface**

As examples, one parameter mapped to different network vulnerabilities (determined via a SAINT security scan); another mapped to machine type (determined via an NMAP host-identification scan); a third mapped to the machine assignments across several building research groups. The host database contained several dozen tables of salient security and configuration information, affording additional token bindings. A GUI-based query pad (similar to the mediaBlocks monitor slot) supported dynamic binding and editing of parameter wheel queries by way of a GUI database front-end.

### *5.5.3 Early query racks*

While the Strata/ML interface showed promise, it also seemed to face several kinds of limitations. For one, the system seemed divided into two fairly separate elements: a display-oriented building representation, and the more interactive query interface. Secondly, Strata/ML's spatially fixtured, three-pad query interface seemed restrictive in flexibility and expressiveness.

In response, I chose to focus on developing the query component of the interface in conjunction with a more traditional graphical visualization of query results. Instead of Strata/ML's fixed, tightly integrated query pads, I redeployed query pads as a series of sliding rectangular panels within "query rack" devices (Figure 5.16). This new approach had several benefits. First, it suggested the prospect that multiple query racks might be used to express larger queries, and themselves spatially manipulated as a kind of nested token/constraints. Second, query racks could hold and manipulate physical aggregates of parameter wheels, allowing fully formed multi-parameter queries to be set aside and later reused. Third, the sliding query pad elements supported development of the adjacency-based Boolean mapping first proposed in the mediaBlocks work [Ullmer et al. 1998], thus extending the range of queries that could be expressed and visualized with the interface.

**Figure 5.16: Second generation parameter wheel interface**

The first implementation of parameter racks and wheels is pictured in Figure 5.16. The first content domain was a database of some 500 research projects. As visualizations of results, the system displayed a Venn diagram showing the logical relationships expressed upon the query rack; a TileBars-like visual representation of the query results' distribution [Hearst 1995]; and an array of image thumbnails representing the actual query results. The Venn and TileBar visualizations were color-coded to match the colors associated with corresponding parameter wheels.

While I was pleased with the performance of parameter wheels, I decided to pursue the parameter bar variation for several reasons. First, I wanted to explore the use of dynamic displays within parameter tokens, enabling rapid logical and visual rebinding of tokens to alternate values. Second, I was interested in supporting the active reconfiguration of parameter tokens while outside of the query rack. Third, I wanted to support double-ended range selection. Fourth, perhaps the dominant reason was that I thought the query bar approach might ease comparison and evaluation with respect to the well-established dynamic queries technique (e.g., [Ahlberg and Shneiderman 1994]).

The resulting prototype is pictured in Figure 5.17. As with the parameter wheels prototype of Figure 5.16, this closely resembles the final-form design pictured in Figure 5.1, with several exceptions. First, the prototypes of Figure 5.16 and Figure 5.17 showed query results on traditional graphical displays, which contributed to a physical and conceptual gap between the query racks and result displays. Secondly, the parameter wheel and bar tokens themselves were redesigned for the final prototype. The parameter wheels were slimmed down, while the parameter bar casing was redesigned to realize a cleaner, more robust device that could withstand use by outside users and subjects (see Figure 5.5).

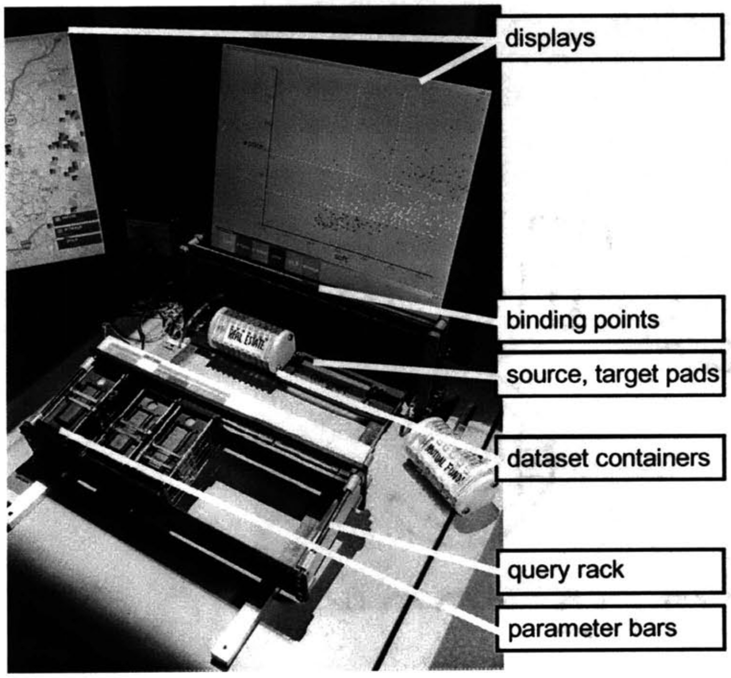


**Figure 5.17: Second generation parameter bar interface**

A final aspect supported by the initial parameter bars implementation was the ability to express adjacency-based Boolean combinations of multiple query racks (Figure 5.18). While I believe this approach was sound in principle, the realization of Figure 5.18 had several limitations. First, the parameter bars and rack itself were relatively large, which created both space issues and somewhat compromised ergonomics when manipulating multiple query racks. Secondly, tightening the proximity between the visualization of query results and the query racks themselves seemed a higher (and somewhat conflicting) priority.

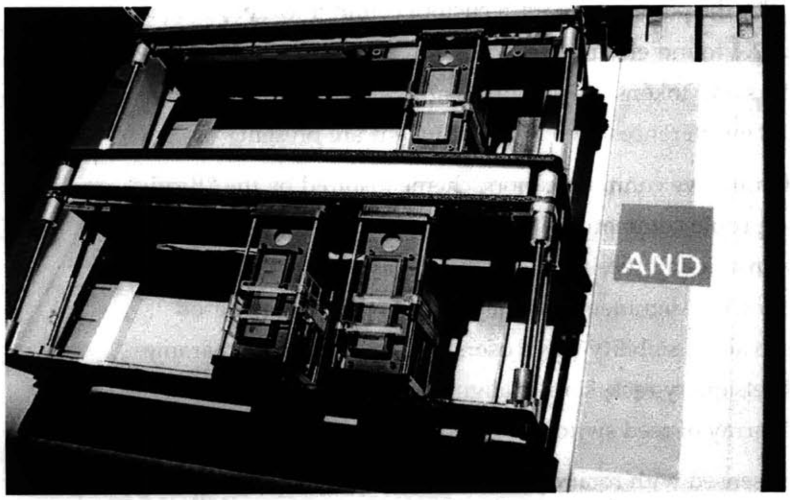


Figure 5.18: Boolean operator mapping for adjacent racks

## 5.6 Implementation

### 5.6.1 Physical design

Most of the query interfaces' mechanical components were fabricated in wood and acrylic on a Universal Laser Systems 100 watt $CO_2$ laser cutter. The laser cutter was controlled by a common 2D drawing program (CorelDRAW™). Components of the final-generation parameter wheels query rack were cut out of aluminum on an OMAX waterjet to provide greater mechanical rigidity. Aluminum rods were cut to length with a band saw, and some screw holes in the parameter bars were manually drilled and tapped. Most of the acrylic components were also sand-blasted. Usage of these fabrication tools is discussed further in §A.4. Projection was via a small InFocus LP150 1024x768 pixel, thousand-lumen video projector, and oriented with a table-based mirror jig (Figure 5.19).

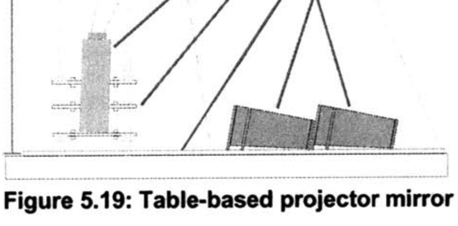


Figure 5.19: Table-based projector mirror jig

### 5.6.2 Electronics design

The parameter wheels were embedded with Philips HiTag2 RFID tags. They were interrogated with an IB Technology H2C tag reader, which was multiplexed across four antenna coils with circuitry by Paul Yarin [Yarin 1999]. For the parameter bars, a custom communication scheme was developed. As discussed in §A.1.2, I found electromechanical interconnections to be unreliable for communication between racks and tokens. Also, optical and other radiative communication mediums are prone to mutual interference when multiple tokens are present.

In response, I developed a near-field inductive communication scheme inspired by the "Beads" of [Resnick et al. 1999]. A pair of low-profile commercial inductors were embedded within each parameter bar. These were used to transmit and receive magnetic impulses with an array of 12 inductor pairs embedded within each rack. Alignment was maintained with gravity-based mechanical detents. These detents also aided stability while users were manipulating parameter bar sliders. For the parameter wheels query rack, similar alignment was achieved with magnetic detents, and sensed with an array of reed switches.

Rotation of the parameter wheels was sensed with rotary potentiometers. Alternately, sensing of the parameter bar levers was monitored with linear slide potentiometers. Parameter bar display functions were realized with 120x32-pixel Seetron backlit LCD displays and tricolor Ledtronics LEDs. These components were controlled with embedded PIC 16F876 microcontrollers. Circuit boards were designed with TechSoft's PCB design software, and fabricated in-house using a Roland CAM-3 vinyl cutter and Modela MDX-20 mini-mill. Parameter bar power was provided via rechargeable NiMH batteries. Query racks were linked by RS232 to a host PC.

As another variation, with support from Prof. Joe Jacobson, I explored the use of with bistable "thermochromic" paper labels for parameter wheels. These would have the advantage of providing parameter wheels with persistent, electronically rewritable labels without requiring the tokens to contain active electronics. While undergraduate assistant Zachary Malchano and I were successful in reversibly printing upon thermochromic paper with a small thermal printhead, the printhead geometry was designed for printing against a cylindrical platen, which was incompatible with the rigidly flat surface of parameter wheels. Correspondingly, this approach was set aside for future development.

### 5.6.3 Software design

The main user interface software was written in Java using the Java2D graphics API, and ran on a multi-processor IBM IntelliServer workstation. The database was hosted on a Linux-based PostgreSQL server and queried using JDBC. The embedded PIC microcontrollers were programmed with the CCS PIC C compiler, and burned to flash-memory PICs with a Microchip PICstart programmer.

Data communication with the parameter bars used a continuous train of four-byte packets – one apiece for framing, token identity, and the position of the parameter bar's two levers. A custom software UART on the host PIC 16F876 microcontroller transmits this information at a rate of roughly 250ms per packet. This is slower than desired (Shneiderman suggests a minimum delay of 100ms), but increasing throughput is a relatively straightforward engineering issue. For future near-field wireless communication, I would probably turn to use of dual-ported RFID transponders, as realized through devices such as Atmel's AT88RF001 [2002] and IBM's Asset ID technology [2000].

## 5.7 Discussion

### *5.7.1 Mapping and integration alternatives*

One of the largest challenges presented by tangible interfaces is the design of strong physical/digital mappings that are both intuitive to people and interpretable by computers. In approaching the manipulation of a quite abstract application domain, this challenge was very evident in the design of tangible query interfaces. This challenge was most apparent for three issues: view composition, query composition, and the overall integration of physical and graphical elements.

In the earliest Strata versions of the query interfaces, views of query results were provided by the building display structures (Figure 5.13, Figure 5.14). While this approach lacked integration between the query and visualization components, it simplified visualization by providing an intrinsic geometrical mapping. The geographical views in the later query interfaces provide this same advantage. Also, prospective integrations of the query interfaces with systems like Urp and TouchCounters (as discussed in §5.4 and elsewhere) are also attractive in providing strong display contexts for the presentation of query results.

However, for the query approach to generalize beyond these more obvious mappings, I felt it was necessary to provide views of information where the display mappings were less clear. This lead to a series of prototypes that exhibited a strong divide between the TUI query racks and separate screen-based graphics (Figure 5.20).

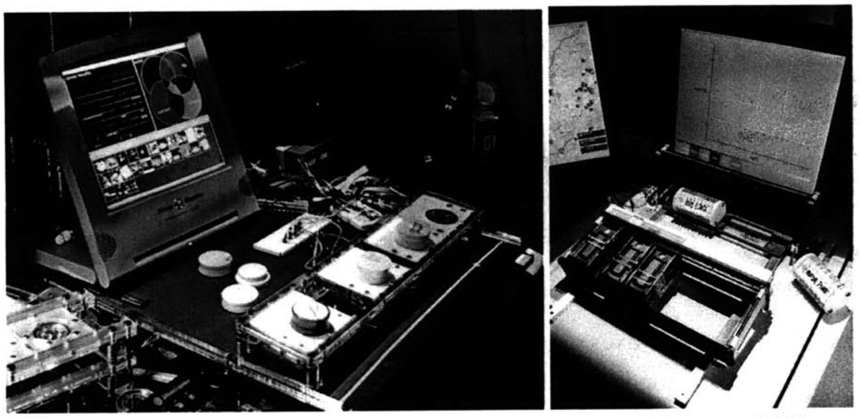


**Figure 5.20: Early parameter wheel+bar implementations with strong TUI/screen divide**

In an effort to provide better system integration and to physically describe the contents of graphical views, I implemented a prototype "view rack" (Figure 5.21). This rack integrated a display surface with two slots for accommodating parameter bars. One slot was associated with the X axis; the other, with the Y axis. The view rack could display both 1D and 2D views, depending upon which of its slots were filled. Also, multiple view racks were implemented, allowing multiple views to be spatially juxtaposed and manipulated.

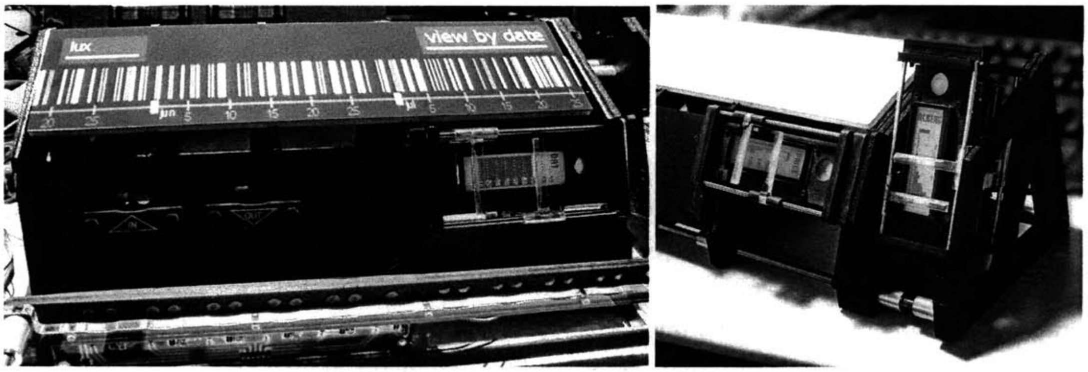


Figure 5.21: View rack prototype; at right, with parameter bars in both X and Y axis slots

While I believe this view rack approach was quite promising, it lead to a lengthened "setup phase" and greater overall complexity of the interaction. As a middle ground between view racks and separate graphical display, I used a projected display surface immediately adjacent to the query rack. Moreover, I used the presence and location of parameter tokens upon the query rack to physically express both the query and view configuration.

As discussed earlier, the parameter bar prototype retained the adjacency-based Boolean mapping, and mapped parameter bars to scatterplot axes on the basis of ordering. In contrast, the final parameter wheels implementation made a fixed association between the leftmost pads and the X and Y axes, and used a purely AND-based Boolean interpretation (Figure 5.22).

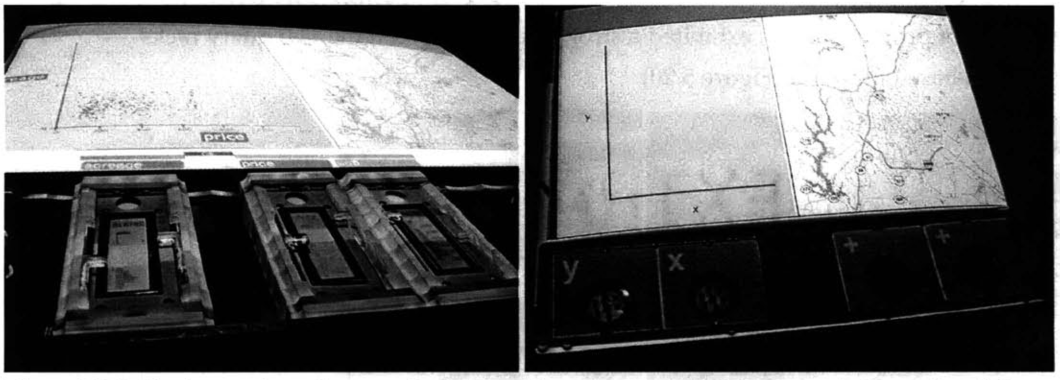


Figure 5.22: Final mapping of parameter bars and parameter wheel query pads to scatterplot axes

I felt that the fixed pad-mappings of the X and Y axes worked quite well. For the parameter bars, while the axis mapping technically functioned, I felt it was a weaker approach than that of the parameter wheels. This was partly because of the fixed, labeled location of the parameter wheel mapping; and partly because the parameter bar mapping was "overloaded" with the Boolean interpretation.

Nonetheless, I believe that the Boolean mapping for the parameter bars holds real value, especially for the kind of context discussed in the mutual fund example of §5.1.2. Simultaneously, the role served by Boolean operators in the parameter bars implementation seems less a function of logical relationships, and more as a means for visualizing the relationships of different parameter distributions. As such, the value of this technique is likely to strongly depend upon the strength of supporting visualizations. While the techniques discussed in §5.4 have provided a starting point, further work is needed to bring this approach to fruition.

### *5.7.2 Reflections on parameter bars*

In some respects, parameter bars are quite different kinds of objects than parameter wheels. These differences relate in part to the parameter bars' integration of LCD displays and fixtured sliders. Among other things, these elements helped further "de-integrate" the elements of the query interface, allowing the state of parameter tokens to be viewed and manipulated by one or more users even when the bars are not present within the query rack. In the process, more functionality was integrated within the bars, and in some respects they were pushed toward the threshold of generic handheld computer devices.

Many aspects of the parameter bar design were evolved to help better integrate them within the query interface. For example, Figure 5.23 depicts some of the evolutions in the physical design of parameter bars. Figure 5.23a shows an early parameter bar in which the internals of the parameter bars were physically exposed, making them visually distracting and awkward to pick up. Moreover, while I had tried to emphasize the physicality of the range sliders, the initial versions occluded the display, and several early users objected to placing their fingers upon the screens.

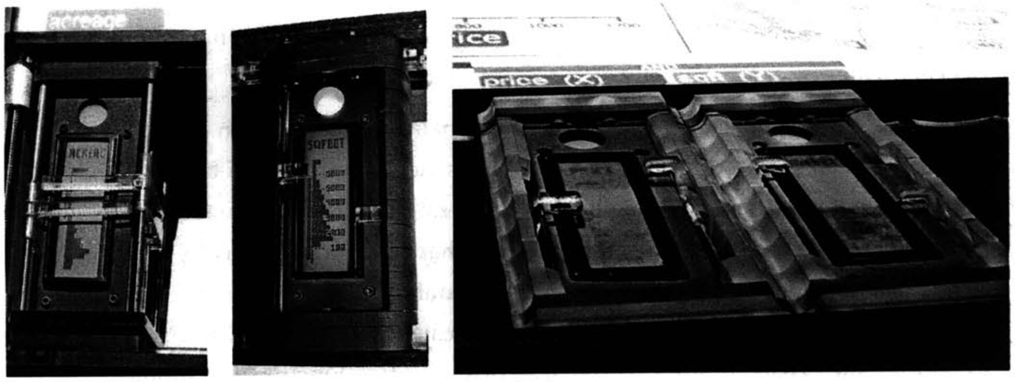
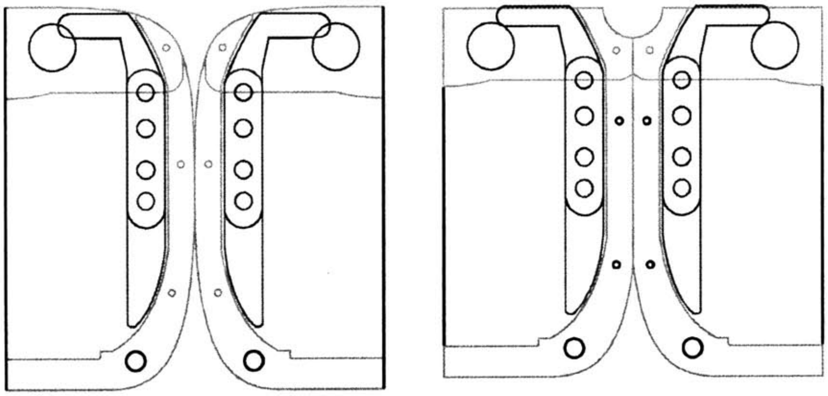


**Figure 5.23a,b,c: Evolutions of parameter bar design**

In the next generation, Figure 5.23b shows an implementation where these issues had been resolved. While I was pleased with the design as a stand-alone object, I discovered that the relationship of parameter bars with each other and with the query rack had been compromised. The physical juxtaposition of these tokens when configured in an "AND" relationship is shown in Figure 5.24a, illustrating their poor physical integration. In response, another design was developed with tokens that integrated more cleanly both with each other and the query rack (Figure 5.23c, Figure 5.24b).



Early design (black bar) Later design (frosted bar)

**Figure 5.24a,b: Side profiles of two parameter bar designs**

However, Figure 5.24 also illustrates one of several failings of the parameter bar design: their physical depth and bulk. While I had strived to minimize the object's size, the use of DIP integrated circuits, single-sided through-hole circuit boards, and other low-integration components contributed to a fairly "chunky" object. At ~6x6x15cm, the parameter bar design more closely approximates a small brick than (say) a domino or deck of cards.

While size and mass are common issues with portable electronic devices, I found that they play a central role in tangible interfaces. Where (e.g.) laptop computers are moved about primarily when *not* in use, TUI tokens are constantly lifted and moved about as their primary modality of interaction. The parameter bar's form factor requires a totally different hand posture than the smaller parameter wheels, fostering a substantially different interaction than intended.

In designing the parameter bars, I had decided on a target functionality and fabrication process, and worked to approximate the desired physical form as best possible. In retrospect, I believe this was a poor heuristic. A better heuristic for tangible interfaces might be to decide upon the physical form appropriate for the intended interaction, and then shape the functionality by what is possible within these constraints. In some respects, this parallels a "lesson from experience" reported by designers of the Xerox Star: "[We] should have established performance goals.... [and when] performance goals couldn't be met, the corresponding functionality should have been cut" [Johnson et al. 1989].

In addition to bulk, I believe the parameter bar fell short in an even more important way. The physical range sliders did offer a persistent, manipulable representation of the bar's digital state. However, they achieved this through traditional controls embedded within a manipulable object, rather than the TUI model of object manipulation as control. This naturally invited questions about the use of more flexible "soft" controls.

From the standpoint of TUI research, I believe this diverged and distracted from the intended design approach, and that the parameter wheels were a stronger TUI design. However, from a broader standpoint of human-computer interaction, I believe this points to the legitimate prospect of developing physical languages appropriate for interacting with general-purpose computing devices. From this vantage point, the parameter bars offer a useful contribution in a design space including work such as [Harrison et al. 1998; Hinckley et al. 2000].

### 5.7.3 *Scope of database functionality*

Tangible query interfaces support the common "select-from-where" form of SQL queries, including database selection, parameter selection, range selection, Boolean operations, and parentheticals (through multiple query racks). The system also supports "select into" through the dataset target pad, and result ordering + rudimentary view selection through alternate interpretations of parameter bar adjacency.

This is a relatively small subset of the full SQL language. Nonetheless, I believe this functionality supports meaningful interaction with a number of content domains. It also represents a superset of prior successful approaches such as the original dynamic queries research [Ahlberg and Shneiderman 1994].

One important database operation I have not discussed is the "join" operation (for relating multiple database tables). While physical expression of this operation is not currently supported, the system internally performs "joins" to relate parameters from different tables following the "universal relations" approach [Maier et al. 1984].

I have considered alternatives for expressing interactive "join" operations. For example, I have explored embodying "views" as physical objects, and expressing "visual joins" [Olston et al. 1998] through the physical stacking of "view objects." However, I believe TUIs are better combined with graphical than textual representations, and existing "visual join" approaches did not seem appropriate for our data.

Another possible extension relates to the expression of logical "not" operations. One possible mechanism for physically expressing the "not" relation is flipping the corresponding parameter token upside down. This seems especially appropriate for the parameter wheels. This approach could also be used in future parameter bars that are thinner and faced with visual display surfaces upon both sides. For the current parameter bars, an alternate prospect might be turning the tokens end-for-end. However, this would effectively shift the location of the embedded range sliders, which may be an unacceptable side effect.

Finally, it is interesting to note that parameter tokens can be embedded with cryptographic IDs, giving them interesting potential for interactions involving sensitive information (e.g., in meetings between people from different organizations). In such contexts, users can bring parameter tokens with them, knowing that the associated data or relations can be accessed only in the physical presence of the parameter object.

### 5.7.4 *Contributions*

Tangible query interfaces build upon the techniques introduced by mediaBlocks, while significantly extending the space of what is expressable with tangible interfaces. Where interaction with the mediaBlocks sequencer expressed individual operations like indexing and concatenation, tangible query interfaces support the expression and manipulation of higher order declarative expressions. Similarly, where mediaBlocks served as containers for specific data elements, the parameter wheels and bars describe relations and logical constraints that are computed over large sets of information. This approach scales to interaction with larger collections of information, and also suggest other kinds of complex computations that can be physically described and manipulated with tangible interfaces.

Tangible query interfaces have been applied to the task of querying databases, which encompasses a broad space of interaction with the digital world. The systems support the rapid expression and manipulation both of queries and also of visualizations of query results. While I do not claim these TUI approaches offer a substitute for the complex expressions crafted by database professionals, I believe they hold value for a range of exploratory interactions, presentation delivery and discussion, and other usage contexts, and believe this claim is supported by the user study of the following chapter.

Tangible query interfaces also illustrate a number of more specific interaction techniques. These include the use of adjacency-based grammars; the use of radial (as well as linear) constraints for token+constraint interactions; dynamic binding of information to tokens that incorporate integral displays; as well as techniques for manipulating both physical and digital aggregates of information, as illustrated by the manipulation of query racks containing configurations of multiple parameter tokens. I believe these techniques are valuable both by themselves, and also in combination with other tangible and graphical interface techniques and systems, such as the Urp and TouchCounters scenarios of §5.4, and the Senseboard and Strata discussions of §7.1.1 and §7.1.2.

# chapter 6

# experimental evaluation

# 6 Experimental evaluation

This thesis has presented both new and pre-existing research developing the concept of "tangible interfaces." I have argued that broad strengths of this approach include support for two-handed interaction and colocated collaboration; leveraging of existing physical skills and work practices; provision of strong physical and cognitive affordances; and support for interaction within niche physical contexts.

In this chapter, I will work to further advance this argument through the empirical support of user studies. In particular, I have conducted a user study that evaluates different aspects of the tangible query interfaces implementations.

## 6.1 Evaluation objectives

The broad goal of the experimental evaluation is to go beyond (reasoned) speculation for supporting some of the thesis claims. However, this leads immediately to the challenge of identifying a metric and process for evaluation. Likely the most common metric for evaluation within the HCI community is that of time-based performance, as compared between a target system and some more traditional control condition.

Partly implicit in this is the assumption that time-based task performance is a primary metric for the success of a system. It is worth noting that this is not necessarily the case – especially for systems involving multi-person collaboration, but also in single-user contexts. For example, in the early days of graphical user interfaces, a number of studies attempted to demonstrate the performance superiority of GUIs over their character-based predecessors. However, experimenters were frequently unable to find GUI performance improvements (e.g., see [Jones and Dumais 1986]), and often found character-based interfaces to be "faster" from a pure time-performance perspective. However, while character-based approaches remain in widespread use, history suggests that many users prefer graphical interfaces for many kinds of interaction.

As with the early experiences of graphical interfaces, I believe the strengths of tangible interfaces in general and tangible query interfaces in particular are difficult to accurately quantify, and often lie outside of task-based performance times. Nonetheless, in evaluating a user interface, some experience with real users seems important. Moreover, to the extent that a quantitative increase in time-based task performance can be observed, this reflects positively on higher-level claims about (e.g.) cognitive and physical affordances.

## 6.2 Evolution of evaluation approach

My efforts to evaluate tangible query interfaces began with early iterations of parameter wheels. I observed that the parameter wheels held common ground with the dynamic queries work of [Ahlberg et al. 1994] and [Williamson et al. 1992], among others. Correspondingly, I sought to compare tangible query interfaces with the GUI "range sliders" approach.

Initially, I was concerned whether the parameter wheels were sufficiently comparable to the GUI range sliders. Also, the early parameter wheels lacked support for several potentially important properties, including dynamic binding and double-ended range selection. While these features could potentially be realized with wheel-based variations, they led me to develop the parameter bar approach. As described earlier in the thesis, a working implementation of parameter bars was realized. However, several technical and conceptual issues made them difficult to employ in user studies (these are discussed further in §6.6.2). Therefore, I returned to evaluation of the parameter wheel approach.

I began a two-part study for evaluating parameter wheels, exploring both "pure manipulation" and expression of query tasks. The first part was loosely modeled upon an empirical study by Fitzmaurice and Buxton described in [1997] and [Fitzmaurice 1996b]. In the Fitzmaurice study, users attempted to match the spatial position and orientation of four physical objects with evolving target positions displayed by the computer. I constructed an analogous experiment that compared parameter wheels with a graphical interface. Here, the computer displayed a series of target configurations involving four different parameters. The user was to match the configuration of the TUI and/or GUI to these targets as quickly as possible.

I ran a "pre-pilot" trial of this first experiment with several users. From a purely quantitative standpoint, the early results were quite encouraging: the tangible interface appeared more than twice the speed of the traditional GUI, even with highly experienced GUI users. However, after further discussions with colleagues, questions arose over the meaningfulness of this experiment. First, the task seemed to differ radically from any meaningful query task. Secondly, the task seemed to reduce parameter wheels to a display-enhanced "dial box," perhaps de-emphasizing the TUI aspects of the interaction.

## 6.3 The experiment

For the main experiment, I chose to focus on the performance of actual query tasks. The experiment was modeled loosely after the Dynamic Queries "HomeFinder" experiment [Williamson et al. 1994]. Where the HomeFinder experiment compared its novel graphical interfaces with textual and paper-based interfaces (yielding favorable results), I compared the parameter wheel query interface with an approximation of the GUI "HomeFinder" prototype.

One claim for tangible query interfaces is that they might provide strong support for exploratory interaction with data. To help test this claim, as well as test the broader usability and effectiveness of tangible query interfaces, I sought to create a test condition that would encourage extended manipulations of the query parameters.

Toward this, with the assistance of undergraduate student Anna Lee, I set up a series of experimental tasks that required the manipulation of between two and four continuous parameters. These were manipulated both within a GUI and on the four-cell query rack. The task parameters drew evenly from a pool of six alternative parameters. This ensured the need for spatially reconfiguring parameter wheels on the query rack, and helped differentiate the configuration from a traditional dialbox setup.

The experimental tasks were designed to require users to balance between multiple competing criteria. For example, a simple task might be "Find the homes that minimize price while maximizing acreage." To help evaluate user success with this task, I implemented a simple scoring algorithm. The algorithm iterates through all database entries that meet the user's current selection criteria. The algorithm then sums up the arithmetic "distance" between each parameter value and the target value for the query; divides the result by the number of selected entries; and compares the result with a "target score" that must be satisfied to complete the task.

The target scores were chosen to be quite selective, generally corresponding to the identification of a small cluster of homes. The best "balance" between the query tasks' competing criteria sometimes required users to identify parameter values that were some distance from the stated target configuration. This required subjects to quickly explore a number of possible selection values to satisfy the experimental task.

The experiment was intended to address three hypotheses:

**Hypothesis 1:** *Tangible interfaces using physically constrained tokens can provide a feasible approach for expressing simple queries*

This first hypothesis corresponds loosely to a "usability" assessment of the parameter wheel-based query interface. Since tangible query interfaces present a new approach for expressing queries, this usability claim benefits from verification through the experience of human users.

It is also worth clarifying that "simple queries" are not the same as "trivial queries." The experimental tasks require the expression of queries involving as many as four independent parameters. While this falls well short of the kind of query complexity that might be required by (e.g.) a database administrator, it also represents a non-trivial querying task.

This is a rather simple hypothesis, with no need for support from statistical analysis. Successful execution of the study will be loosely taken as a kind of "existence proof" for this hypothesis.

**Hypothesis 2:** *TUIs facilitate and elicit parallel two-handed interactions within querying tasks.*

One general claim of tangible interfaces is support for parallel two-handed interactions. While this style of interaction has been demonstrated with numerous previous tangible interfaces, it remains to be shown that this interaction style carries over to interactions and tasks such as querying.

I believed that the nature of the selected experimental task, with its sensitive dependence between multiple competing criteria, might naturally draw forth and benefit from two-handed interaction by human subjects. On the other hand, users might also decide to use only single-handed interaction with the interface. This could either arise out of habit (from previous single-handed interactions with computers), or out of a desire or need to focus on the manipulation of only a single parameter at any given time.

To expand slightly on the hypothesis' wording: by "facilitating" two-handed interaction, I mean that tangible query interfaces provide a mechanism by which parallel two-handed interaction can be done. By "eliciting" two-handed interaction, I mean that tangible query interfaces encourage parallel two-handed interaction, to that extent that this is frequently done in practice.

**Hypothesis 3:** *TUI is faster than GUI for a range of simple querying tasks.*

The most ambitious of the three hypotheses is the proposition that tangible interfaces (e.g., the parameter wheels) are quantitatively faster in time-based task performance than comparable best-practice graphical interfaces (e.g., well-implemented versions of range sliders).

I anticipated a performance increase by the tangible interface for several reasons. First, I believe the tangible interface aids users in focusing upon the "objects of interest" – in this case, the specific parameters of the query task. Secondly, I believe the TUI is more "direct" in manipulation than the GUI. This allows better use of kinesthesia, with eyes focused on the query and scoring results, and muscle memory leveraged for mediating interaction with the parameters and parameter ranges. Moreover, to the extent that the second hypothesis is confirmed, I expected that parallel two-handed manipulation of the parameter wheels would also contribute toward a TUI performance increase for the experimental task.

At the same time, it is possible that the tangible interface's performance may be comparable to or slower than that of the graphical interface. I will discuss several possible reasons in the "biasing effects" section (§6.4.4). But more broadly, especially if some of the predictions in the last paragraph do not hold true, it is possible that tangible interfaces (at least in this particular configuration, for this particular task) do not hold a speed advantage over graphical interfaces.

## 6.4 Method

### 6.4.1 Equipment

The study is based on two experimental conditions: the tangible interface and graphical interface. The parameter wheel-based tangible interface was configured similarly to the descriptions in Chapter 5, and as pictured in Figure 6.1a. The central artifact is the "query rack," with its four rotary sensing cells shifted to the left of the device. In front of the query rack, six continuous-valued parameter wheels were placed with standardized order and orientation in a holding tray. They were returned to the tray in the same configuration between each experimental task, to minimize experimental variables.

A white mat board display surface, 30x50cm in size, was placed immediately adjacent to the query racks. Both this display surface, as well as the wheel pads of the query rack, were illuminated by the output of an InFocus LP130 projector. The projector itself has an output of 1024x768 pixels at 1100 lumens. Due to projector geometry and minimum "throw distance" constraints, effective resolution on these display surfaces is slightly less than 800x600 pixels.

The graphical interface configuration used an 18" LCD screen (30x35cm display area), driven at 800x600 pixel resolution. Mice configured in both left- and right-handed versions were provided to meet the preference of individual users. Keyboards were neither used nor present in the work area. The same underlying software is used for both the graphical and tangible interfaces, and I worked to equalize the responsiveness of the two interfaces as much as possible.

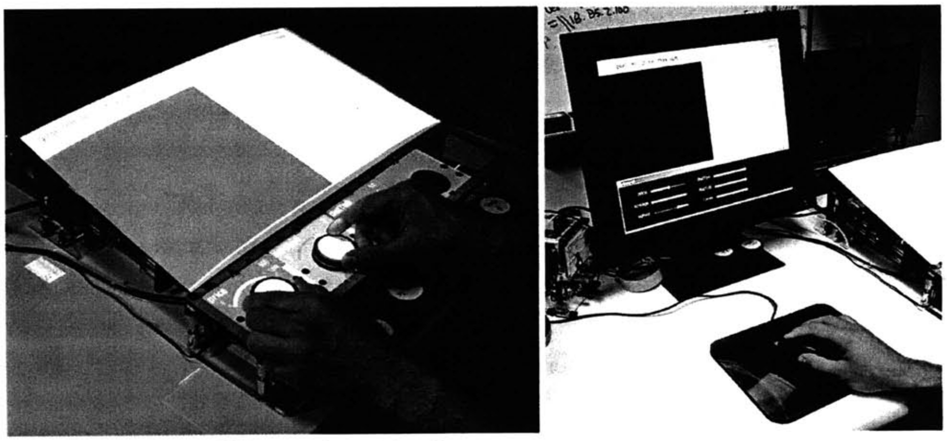


**Figure 6.1a,b: TUI, GUI experimental setups**

Roughly the same two-dimensional graphical layout was used for both GUI and TUI interfaces (see Figure 6.2). At the top of the display surface/screen, the text of task questions was listed, and graphical "score bars" were displayed. Beneath this, a geographical view of the query results was displayed. In the TUI, parameter wheels were illuminated as in Chapter 5. In the GUI, a variation on the original range sliders were developed, and evolved in response to user feedback. The GUI sliders occupied roughly the same "visual real estate" as the TUI query rack.

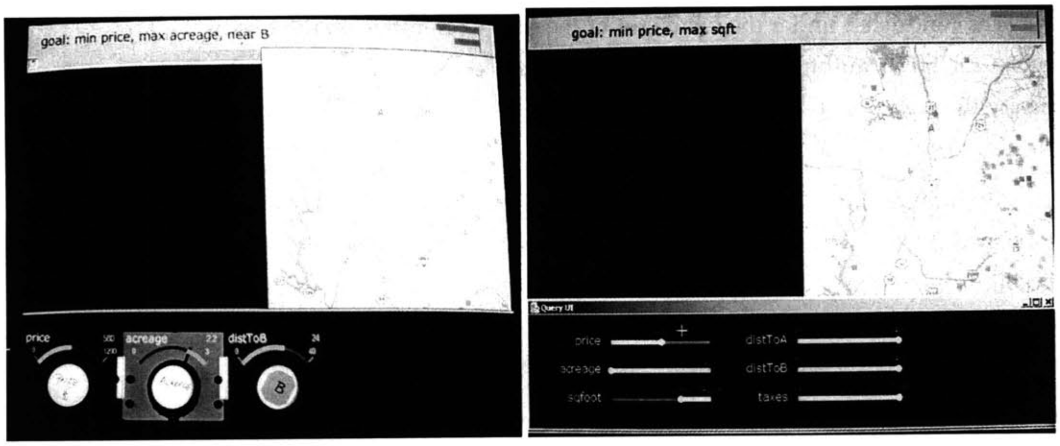


**Figure 6.2a,b: Visual layouts of TUI and GUI experiments**

### 6.4.2 Task

As introduced in §6.3, each experiment consisted of a series of single-user tasks both within a GUI and on the four-cell query rack. In both usage conditions, subjects were prompted with a brief textual description of a desired set of conditions. Examples include "min price, max acreage" and "min taxes, max sqft, near location B." The task was for subjects to express the specified condition such that the competing criteria are "mutually optimized," as evaluated by the system's scoring algorithm (described in §6.3).

Subjects express these conditions either through the manipulation of graphical range sliders, or the placement and manipulation of parameter wheels. The results were continuously evaluated and displayed graphically with a bar graph comparison of the current score vs. the target score for successful task completion (see [Figure 6.2]). When the target score was achieved, the system acknowledged the task completion, and the experimenter moved on to conducting the next task.

The user study task offers a subset of the full TUI functionality described in Chapter 5. For example, where the full TUI uses parameter wheels representing both continuous and discrete parameters, the experimental task only makes use of continuous parameters. Since the tasks were chosen to emphasize rapid fine tuning of parameter values, I decided that it was best to limit queries to continuous value ranges for the experimental tasks.

Similarly, Chapter 5 also illustrated the use of the scatterplot as well as geographical visualizations. The user study task instead provided only the geographical visualization and the scoring display, with the implicit expectation that users will focus most of their attention on the scoring display. While the scatterplot visualization offers much additional information that could potentially speed user responses, its method of axis selection builds upon a TUI interaction for which no common GUI equivalency exists. Comparing the novel tangible interface with an equally novel GUI seemed an unwise increase in the experimental variables. For this reason, I removed the scatterplot visualization from the study.

### *6.4.3 Procedure*

The experiment was conducted under the oversight of the MIT Committee on the Use of Human Subjects (COUHES), authorization number 2801. Subjects were drawn from the Cambridge community through the posting of call-for-subject fliers. Experiment sessions took on the order of 30 minutes. Subjects were paid $10 for their time.

All subjects used both of the study's two interaction conditions: the TUI and GUI interfaces. The study established two clusters of experimental tasks that were conducted in a fixed order. A within-subjects study methodology was adopted, with Latin-square counterbalancing used to minimize order effects. In practice, half of the subjects used the TUI first, followed by the GUI, and half of the subjects used the converse ordering, with the order switched for each subject.

In order to reduce experimental biasing, the experiments were conducted by undergraduate assistant Anna Lee. Spoken instructions followed a fix script (see Appendix B). After completing consent forms and verifying that subjects were comfortable, subjects began the first of the two clusters of tasks. Each task-cluster began with a series of four training/"warm-up" tasks. In these warm-up tasks, the system prompted subjects with questions and scoring similar to those of the actual study. Subjects were encouraged to ask questions during this phase of the experiment, and given as much time as necessary to feel comfortable with the system.

The experiment then moved on to a series of timed query tasks. The tasks were remotely initiated by the experimenter from a second computer to avoid interference during GUI tasks. Task questions were presented by the computer, and timing data was automatically logged to file. On completion of each task, the experimenter confirmed success, and then reset the physical + graphical configuration of the apparatus for the next test in the trial.

On completion of the cluster of tasks for the first trial condition (whether GUI or TUI), the experiment then switches to the second trial condition. At the conclusion of the experiment, subjects were given a questionnaire and debriefed.

The computer required users to maintain the target parameter configuration for 0.5 seconds. This duration was intended to minimize the prospects for satisfying the automated scoring through "unorthodox" methodologies – e.g., through quick random adjustments of parameters.

The results of selected experiments were videotaped for further analysis (as authorized by subjects through a consent form).

### 6.4.4 Experimental biasing

In §6.3, I discussed several reasons why tangible interfaces might be faster than graphical interfaces for the experimental querying task. At the same time, there are several reasons why tangible interfaces might be slower that at least partially reflect experimental bias. For one, a great many people have extensive experience interacting with mice and graphical interfaces. Therefore, even if the parameter wheels "leverage existing physical skills," they are also less familiar devices for most first-time users than the mouse & GUI slider interaction.

Secondly, the TUI interaction within the experimental task appears to naturally elicit a "setup phase" – the selection and placement of parameter wheels onto the query rack, corresponding with the associate phase of token+constraint interaction – that is not present in the graphical interface. In the GUI, all available parameters are by default "active" and displayed on the display screen. In TUIs, there may generally be more parameters available than present within the interaction area; and toward this, I intentionally chose a number of parameter wheels greater than the number of available query pads. However, it was unclear what the "fairest" analog might be for the experiment's TUI tasks. Should certain parameter wheels already be present on the query rack at the beginning of each task? Or should users be required to reconstruct a configuration that is already implicitly done by the GUI?

For the experiment, I began TUI tasks with no parameter wheels on the query rack. I believe this represents a kind of biasing in favor of the GUI. In the first pilot experiment, I attempted to reduce this biasing by running clusters of tasks "back-to-back" (without pause). In some portion of tasks, successive questions referenced the same parameter wheel. In this way, some wheels would remain on the rack between successive tasks, as would be expected in actual usage situations.

However, based on the experiences of the first pilot, I decided to administer and time tasks individually, resetting the GUI & TUI configurations between each task. I will describe the reasons for this in the discussion of the first pilot experiment (§6.5.1). I will also analyze the experimental results both with and without this "setup phase" to better understand its impact on time-based performance.

Another potential bias concerns the starting values of range sliders and parameter wheels. At the beginning of each task, as with the dynamic query experiments of [Williamson and Shneiderman 1992], GUI range sliders are configured to begin in a fully selected state (see Figure 6.3a). To be consistent with the range sliders in the published literature (which are "always active"), this is the only configuration that allows starting a query task with a "blank slate."

**Figure 6.3a,b: Fully & partially selected range sliders; fully and "50% selected" parameter wheels**

The parameter wheel interaction is somewhat different. Parameter wheels are "activated" by placing them upon the query rack, and "deactivated" by their removal. Thus, beginning each task with an empty query rack achieves the "blank slate" configuration. Parameter wheels could be configured to begin in a fully selected state. However, I believed that starting at a 50% selected state (see Figure 6.3b) would be a more "legible" configuration, especially for first-time users. I believed this because wheel rotation in either direction would produce an immediately visible result; and the distance between the starting and target wheel configurations would (on average) be minimized. While these starting values may introduce a slight bias toward the TUI, it seemed to fairly reflect the different functional constraints of these two approaches, and to be more than counterbalanced by the decision to start with an empty query rack.

It is also worth noting that the tangible interface used by subjects is a more experimental and less evolved interface than the corresponding graphical interface – especially from a hardware perspective. Complications in the smoothness and speed of response yielded by interface electronics; various mechanical shortcomings; and other issues are all factors that detract from the TUI's performance. On the other hand, one can assume that tangible interfaces will generally have added costs over graphical interfaces, both in terms of money, physical space, and availability. To make the case for TUIs, they should ideally be able to prove themselves even among the limitations common to many experimental systems.

## 6.5 Results

### 6.5.1 First pilot experiment

An initial pilot experiment of the querying task was run with four users. Two were male, two female. One had extensive computer experience; the others had roughly average computer backgrounds. The experimental tasks consisted of two clusters of tasks, each with four consecutive (back-to-back) tasks. Each user performed the same set of tasks on both the GUI and TUI (albeit in different orders). While this duplication was not intended to continue in the actual experiments, it gave more data for comparison within the small pilot experiment.

The initial pilot results, both quantitative and qualitative, were quite encouraging. In these pilots, the TUI outperformed GUI in cumulative task completion time by an average of 29%, albeit with large variance. All of the subjects indicated a strong subjective preference for the TUI. All of the users made heavy use of two-handed, parallel interactions with the TUI. Three of the users completed a questionnaire. All listed the two-handed operation as a major reason for their preference. In the evolved questionnaire used for the final two users, both described their qualitative sense of the TUI's speed as significantly faster than GUI (both giving it a 6 on a 1 to 7 scale; 1=GUI much faster, 7=TUI much faster).

Based on the experiences of the first pilot, I decided to administer and time tasks individually, resetting the GUI & TUI configurations between each task. One reason stemmed from user complaints about the GUI configuration. To be consistent with the range sliders of the published literature, I did not provide a "global reset" button to clear slider state, or a mechanism for individually disabling particular sliders. However, in the case of the TUI, all four users expressed a strong appreciation for the ability to simply remove a parameter wheel from the query rack to remove its contribution to a query, and all four requested a similar ability in the GUI. I felt uncomfortable adding non-standard "features" to the GUI, in case they might unduly bias GUI performance (whether positively or negatively); but also felt it unwise to ignore this strong, consistent feedback.

Simultaneously, both electrical and mechanical TUI limitations appeared in the first pilot. As discussed in Chapter 5, the query pad's sensing of parameter wheel rotation was accomplished with potentiometers. In the pilot experiment, the sensing of these potentiometers was quite noisy, to the extent that all users complained this interfered with their ability to solve the task. Fortunately, I was able to solve this problem by adding a voltage regulator to the power supply of the microcontroller hosting the potentiometer's analog-to-digital converters.

The use of single-turn potentiometers also introduced a ~300° mechanical limit to the sensor's rotary range. To address this, I took care to "zero" the position of potentiometers before the beginning of each cluster of tasks. Full range selection with the parameter wheels mapped to a ~100° rotation of the potentiometers. This meant that selection with a single parameter wheel rarely led to any complications; but successive usage of even four parameter wheels in the same slot could lead to the potentiometer reaching a mechanical limit during the experiment. I had intended to replace the potentiometer with a shaft encoder or optical quadrature phase encoder. However, time limitations did not allow for this before conducting the experiments.

In order to address both the "GUI reset" concern, the potentiometer rotation limitation, and potential breakdowns if users failed to answer a particular query tasks, I chose to alter the experiment. In particular, I decided to pose a larger series of query tasks, doubling both the number of warm-up tasks (from 2 to 4) and experimental tasks (from 4 to 8) for both GUI and TUI conditions. I also decided to administer and time each task individually, resetting both GUI and TUI to their starting condition between each task.

### 6.5.2 Second pilot experiment

A second pilot experiment was run to test the revised experimental conditions and tasks prior to the full experiment. The procedure for the pilot was as described in §6.4.2 and §6.4.3. Four subjects were used, two male and two female. For the first three subjects, the experiment again consisted of two clusters of tasks. Each cluster began with four "warm-up" questions, and continued with eight timed tasks. The fourth subject followed the final task configuration.

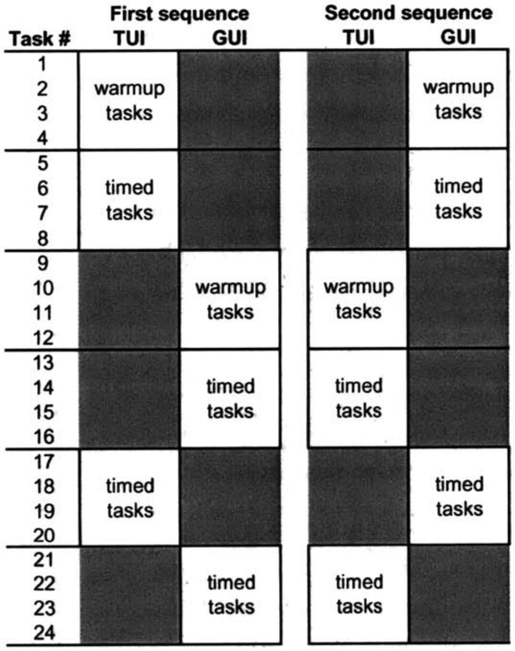
In the second pilot and the final experiment, I decided to conduct each task individually, resetting the parameter wheels and range sliders to their initial state after each task. As described in §6.6.1, this was largely in response to experiences from the first pilot, where all four users expressed frustration with the need to manually reset GUI sliders between tasks. While this technique echoed the approach of [Williamson and Shneiderman 1992], I felt uncomfortable in either maintaining a condition users unanimously found unsatisfactory, or in introducing novel features to the GUI control condition. Thus, the separate-task approach seemed the simplest and cleanest compromise.

I believe that conducting tasks individually introduced several kinds of biasing, mostly to the detriment of the tangible interface. First, it removed an aspect of interaction with the TUI that users strongly appreciated. All users in the first pilot commented with pleasure at the way physically removing tokens deactivated the associated query terms. The new experimental condition effectively removed this feature. Secondly, it removed an aspect of interaction with the GUI which, while frustrating to users, is consistent with their published style of usage. Thirdly, it meant that TUI users could not leave parameter wheels upon the query rack between successive tasks. This is an artificial sequence that causes users to repeatedly incur the cost of the TUI "setup phase" in a fashion that differs substantially from the expected "normal" use.

In another difference from the first pilot, I decided not to include repeats of certain tasks across both TUI and GUI conditions. While this repeated measure is very useful for comparing individual user's performances under the two conditions, I was afraid this non-standard experimental practice might weaken the study's results. As a consequence of this decision, it was impossible to gauge the comparative within-subjects TUI/GUI performance with the new task structure.

The second pilot experiment contained 24 tasks (including 8 warm-up tasks). This was twice the number of tasks present in the first pilot. These additional tasks were prepared by undergraduate assistant Anna Lee. Their composition was more complex than for the initial dozen tasks, as I had already identified and put into use many of the easily identifiable parameter pairings. As with the initial round of tasks, our goal was to identify the most challenging target conditions that we could reasonably expect subjects to complete. As a consequence, one of the second pilot's most important functions was to identify and modify the four tasks that initially proved too difficult for pilot subjects.

With the increased number of tasks, Lee and I also noticed an increased learning effect. Additionally, the first two subjects of the second trial commented explicitly that they achieved comfort with the tasks midway into the experiment. Correspondingly, we decided to interleave the GUI and TUI conditions into four distinct clusters of tasks. The final task configuration is illustrated in Figure 6.4. In this new configuration, we decided to concentrate analysis of results upon the final eight tasks (numbers 17-24), believing that this sequence would give subjects ample time to "come up to speed."



| | First sequence | | Second sequence | |
|---|---|---|---|---|
| **Task #** | **TUI** | **GUI** | **TUI** | **GUI** |
| 1 | warmup tasks | ■ | ■ | warmup tasks |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |
| 5 | timed tasks | ■ | ■ | timed tasks |
| 6 | | | | |
| 7 | | | | |
| 8 | | | | |
| 9 | ■ | warmup tasks | warmup tasks | ■ |
| 10 | | | | |
| 11 | | | | |
| 12 | | | | |
| 13 | ■ | timed tasks | timed tasks | ■ |
| 14 | | | | |
| 15 | | | | |
| 16 | | | | |
| 17 | timed tasks | ■ | ■ | timed tasks |
| 18 | | | | |
| 19 | | | | |
| 20 | | | | |
| 21 | ■ | timed tasks | timed tasks | ■ |
| 22 | | | | |
| 23 | | | | |
| 24 | | | | |

**Figure 6.4: Ordering of tasks within two experimental sequences**

### *6.5.3 Main experiment*

The main user experiment included 16 users, nine male and seven female. Fifteen of the sixteen users were right-handed. In posting the call for subjects, I wished to avoid a subject pool primarily drawn from the MIT undergraduate population, as I did not feel this population was representative of the base of users who might most benefit from or be drawn to tangible interfaces. Lee and I posted primarily within a several-block radius of our laboratory, which is several blocks from the MIT campus, and in a business area. Our subjects reflected this proximity; half (8) had MIT affiliation; five were active MIT students; and three of these were MIT undergraduates. None of the subjects was affiliated with the Media Laboratory. Subjects ranged from 19 to 45 years of age (our experimental approval required subjects to be at least 18 years of age). The average age was 27; ten subjects were under 25, three were between 30 and 35, and three were between 40 and 45. While this was neither our goal or preference, the subjects were relatively computer literate; all but one responded using e-mail.

As described in §6.5.2, each experiment consisted of 24 tasks, ordered in the fashion illustrated in Figure 6.4. Subjects were assigned consecutive ID numbers. Even-numbered subjects followed the first task sequence; odd-numbered subjects followed the second. For the analysis of results, I concentrated upon the final eight tasks; the first eight timed tasks were also considered.

The performance of subjects, both within and across tasks, varied substantially. The fastest subjects performed on the order of three times the speed of the slowest subjects. Also, the time-performance from task to task varied significantly. The slowest task averaged more than four times the length of the fastest task.

On a task-by-task basis, the average GUI vs. TUI completion times for a given task were determined by comparing the average completion times by the odd-numbered subjects with those of the even-numbered subjects. The ratio between the resulting times indicates the percentage of relative performance increase or decrease.

The aggregate time-based performance of subjects within the experiment can be calculated in several different ways. In one approach, the ratios of GUI vs. TUI performance for each task could be averaged. Alternately, the average of GUI performance times across all tasks can be compared with the average of all TUI performance times.

The average of ratios approach has the advantage of weighing each experiment more equally, given the widely varying completion times of individual experiments. Alternately, the average of times approach has the advantage of arguably being "more direct" in its measure.

The aggregate GUI vs. TUI time-performance results from the experiment are presented numerically and graphically in Table 6.1, Figure 6.6, and Figure 6.5. These tables and figures compare several different results. First, they present the average of ratios and average of times for the aggregate performance data. Second, they display TUI results both as directly measured, and as adjusted to remove the time for the TUI "setup phase" (the "period in which parameter wheels were being moved to the query rack, before active parameter manipulation began). Finally, they compare the performance results both for the final eight questions of each experiment, and for a six-task subset of these questions.

Several observations are apparent from these results. First, the raw TUI time-based performance of the tasks was slower than that of the GUI. The degree of slowness depends on the metric that is selected. Omitting tasks #20 and #24 and using the average of ratios approach, the TUI was roughly 3% slower than the GUI. Alternately, using the average of times metric over all eight tasks, the raw TUI performance was nearly 30% slower than the GUI.

However, looking at the results in a slightly different fashion yields very different results. If the setup phase is removed, depending upon the metric and comparison tasks that are chosen, the aggregate TUI time-based performance ranged from nearly identical to that of the GUI, to a 45% improvement over the GUI.

| Average across eight final tasks | | | Average less tasks #20 and #24 | | |
|---|---|---|---|---|---|
| **Task type** | **Ratios** | **Times** | **Task type** | **Ratios** | **Times** |
| TUI | -11.7% | -28.7% | TUI | -2.5% | -7.9% |
| TUI w/o setup | 26.3% | -0.5% | TUI w/o setup | 45.1% | 31.5% |

**Table 6.1: Average time performances of tasks, relative to the GUI.**

*Negative values represent percentage by which TUI performance is slower than comparable GUI performance. Conversely, positive values represent percentages of TUI performance increase over comparable GUI.*

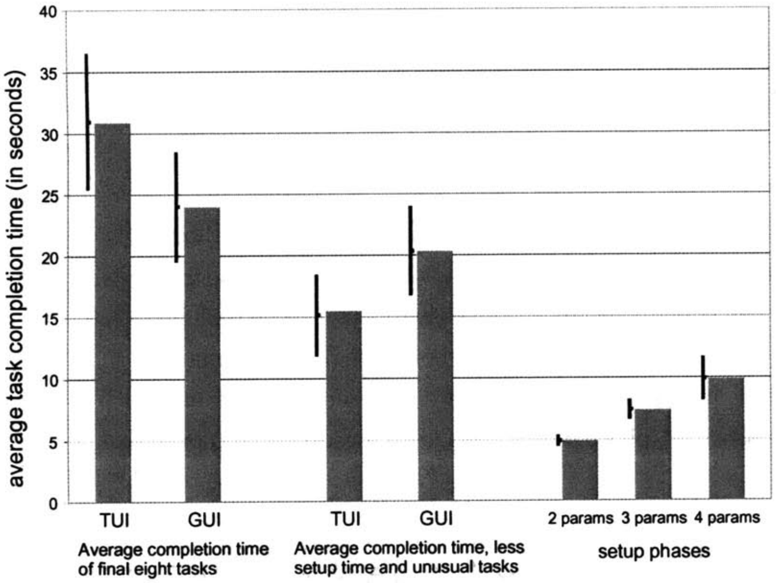


**Figure 6.5: Average task completion times and 95% confidence intervals for experiment**

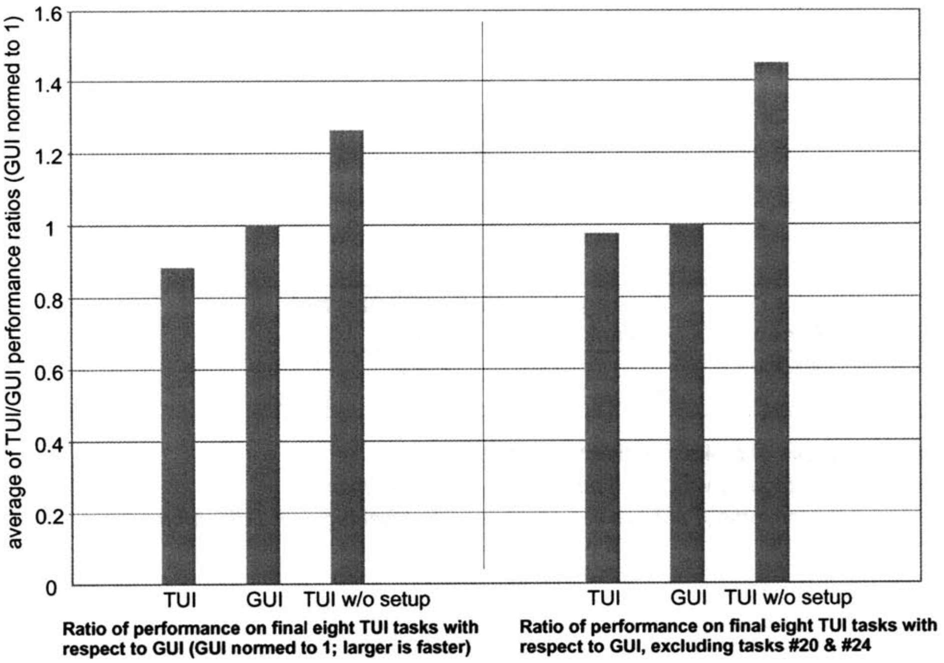


**Figure 6.6: Average of TUI/GUI performance ratios**
*(GUI performance normed to one; higher numbers indicate faster relative performance)*

### 6.5.3.1 *Impact of setup phase*

To understand these diverse interpretations of the results, several factors are important. First, subjects engaged with the TUI in a qualitatively different way than with the GUI. While this was never suggested, fifteen of the sixteen subjects began TUI tasks with a "setup phase," picking up all relevant parameters and placing them on the query rack before beginning to manipulate the associated parameter values. In a sense, the setup phase represents a separate performance of the token+constraint "associate" phase by users (as introduced in §1.1.2) prior to beginning the "manipulate" phase.

The TUI setup phase had the beneficial effect of focusing user attention upon the specific "objects of interest" within each query task, leaving only the relevant elements present upon the interactive workspace. Moreover, most users ordered the parameter wheels in the same order as expressed by the task question at least some of the time. Fourteen of the sixteen users (88%) maintained this order at least some of the time. These users made mirrored the task order with the wheel configuration an average of 71% of the time. While this was not necessary to solve the task, it perhaps offered an additional cognitive benefit by creating a more direct physical "model" of the task at hand.

However, the setup phase has a significant time cost. As illustrated in Figure 6.5, the setup phase for two-, three-, and four-parameter tasks had an average duration of 4.9, 7.4, and 9.9 seconds, each with a standard deviation of roughly half a second. Moreover, as illustrated in Figure 6.6, these durations represent on the order of 20%-30% of the average time spent solving the final eight TUI tasks.

The graphs of this chapter compare the TUI performance both in "raw" form, and with the setup phase subtracted out. The merits and implications of these alternative views will be considered in the chapter's discussion.

### *6.5.3.2 Impact of task selection*

The contents of Table 6.1, Figure 6.6, and Figure 6.5 illustrate that the experimental tasks #20 and #24 made a significant impact in performance times. To better illustrate the role of these tasks, Figure 6.7 presents a task-by-task view of average task completion times. Odd-numbered tasks included two parameters; even-numbered tasks included three parameters. The duration of the TUI setup phase within each task is also plotted for reference.

In considering this graph, it is useful to revisit the composition and scoring of tasks. As mentioned earlier, the experiment consisted of 24 tasks, each composed of combinations of six parameters: price, acreage, square footage, taxes, and distances to two points on the map (A and B). In composing the tasks, we chose combinations of two to four parameters, and requested users to "balance" between the competing criteria. The same scoring algorithm was used for each task. Assistant Anna Lee and I determined target scores for each task, picking the most challenging target conditions that we could reasonably expect subjects to complete.

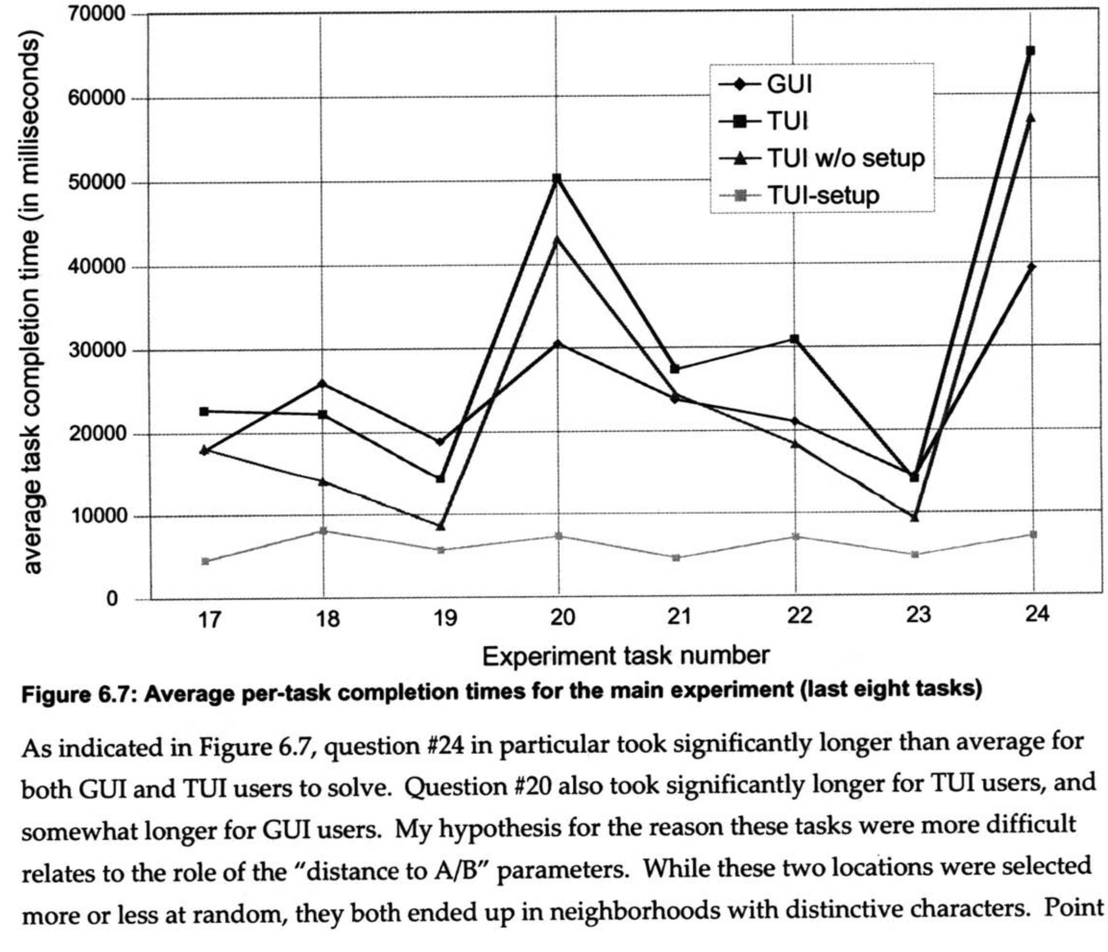


**Figure 6.7: Average per-task completion times for the main experiment (last eight tasks)**

As indicated in Figure 6.7, question #24 in particular took significantly longer than average for both GUI and TUI users to solve. Question #20 also took significantly longer for TUI users, and somewhat longer for GUI users. My hypothesis for the reason these tasks were more difficult relates to the role of the "distance to A/B" parameters. While these two locations were selected more or less at random, they both ended up in neighborhoods with distinctive characters. Point

A was located in an area with relatively few homes, of which few had low prices. Point B was located in a more densely populated area, where few homes had large acreage.

Questions 20 and 24 both asked for homes with small price and large acreage, near points A and B, respectively. In order to successfully solve these tasks, users had to select parameter values for distance to A/B, price, and acreage that were relatively far from the stated targets. TUI users tended to keep their eyes on the map and score bar, rather than the parameter wheels; while GUI users were forced to frequently look at the range sliders. In the case of questions 20 and 24, I believe these tendencies lead TUI users to continue making many small (and ultimately unsuccessful) "tweaks" in close vicinity to the requested target positions. In contrast, I believe it may have been easier for GUI users to [observe and back off faster on values.]

Tasks #20 and #24 were picked more or less at random, and exacerbated unusual configurations in the data. Correspondingly, I have also provided data that reflects TUI/GUI comparison if these tasks are dropped. Figure 6.8 provides another view of the per-task performance times in the absence of these two tasks.

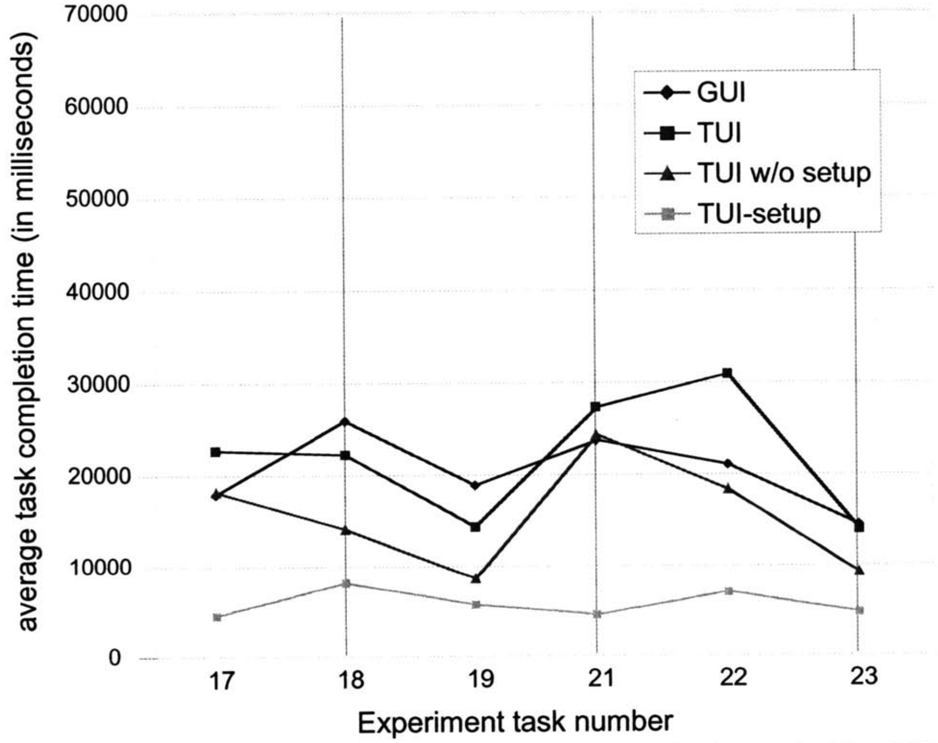


**Figure 6.8: Average per-task completion times: last eight tasks, less tasks 20 and 24**

At the same time, it can be argued that tasks 20 and 24 are equally legitimate, and should be included in the experimental analysis. Also, the poor TUI performance on these two tasks raises the question of whether the TUI was somehow poorly suited toward solving "hard" tasks. This is a somewhat open-ended question, as "hard" has many different possible metrics. However, for one concrete perspective, it is useful to look at the time-performance data for the first eight timed TUI/GUI tasks, as plotted in Figure 6.9.

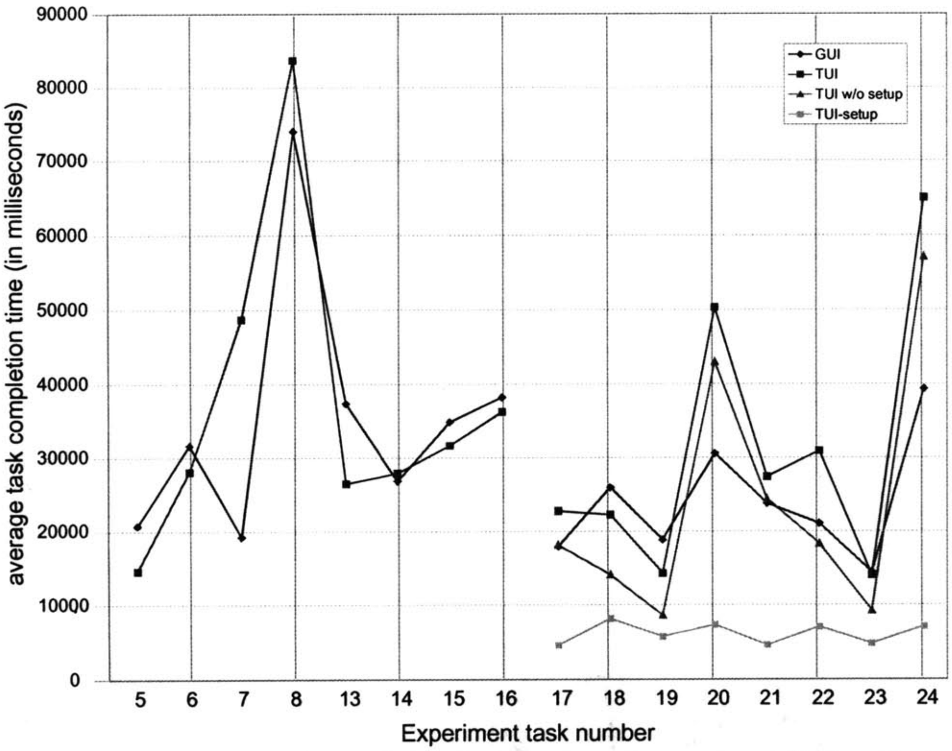


**Figure 6.9: Average per-task completion times for the main experiment (16 timed tasks)**

This figure indicates that even before the setup phase is accounted for, raw TUI performance was faster than GUI performance for five out of the eight tasks. This data also indicates interesting behavior on tasks seven and eight. While a two-parameter task, question #7 included the same criteria that I believe were problematic in question #20: optimizing for minimum price in the proximity of point A. The resulting discrepancy in GUI/TUI performance was even larger than in questions #20 or #24. In contrast, question #8 was a four-parameter question, and apparently the most difficult task of the experiment. While it contained the three criteria of question #24, it also contained square footage as a fourth criteria. With this additional constraint, TUI and GUI performance were relatively similar. This loosely suggests that the relationship between task difficulty and GUI vs. TUI performance is not so simple.

### 6.5.3.3 *Qualitative user assessments*

In addition to time-based task measures, subjects also completed questionnaires with a qualitative evaluation of the GUI and TUI interfaces. One question involved a comparison of the GUI and TUI interfaces on a seven-point Likert scale, where subjects were requested to assess their preference between the GUI and TUI interfaces. These results are summarized in Figure 6.10.

The user responses were roughly evenly split: 8 users preferred the TUI, 7 preferred the GUI, and one was undecided. The average user preference is 4.5 on a 7 point scale, weakly favoring tangible interfaces. However, it is significant to note that the preference histogram illustrates a distinctly bimodal distribution (Figure 6.10). Seven of the sixteen users (44%) had a moderate or strong preference for the tangible interface. Alternately, six subjects had a weak preference (and one a strong preference) for the graphical interface.

Adding an additional dimension to these preferences, seven users identified familiarity as the strongest strength of the graphical interface (with specific comments like "comfortable and common mouse control" and "it's familiar to lots and lots of people"). Of these seven users, three had a weak preference for the GUI; three had a moderate preference for the TUI; and one had no preference.

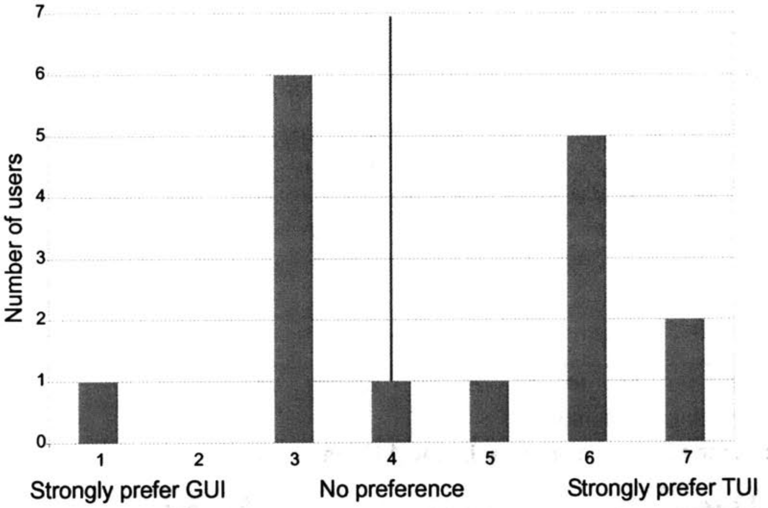


**Figure 6.10: User preferences between the graphical and tangible interfaces**

GUI/TUI preferences did not seem to cluster by sex. Four females preferred the TUI; three preferred the GUI. Four males preferred the TUI; four preferred the GUI.

The TUI seemed to be preferred by a somewhat younger audience. Of the users preferring the TUI, the average age was 23; the average age of those preferring the GUI was 30. Again, the average age of the subjects overall was 27 (Figure 6.1). However, there are several important caveats to this. The subject pool was self-selecting, and seemed to gravitate to those with moderate to significant computer experience. Also, while I sought to encourage the participation of users over 50, the oldest subject was 45 years of age. This is considered further in the chapter's discussion.

Several subjective user ratings are illustrated in Figure 6.11. As discussed above, slightly more than half of the subjects preferred the TUI, and on average these users had a moderately strong preference. In comparison, users preferring GUIs had a weaker preference, often citing the GUI's familiarity as the major reason. When asked for their subjective assessment of interface speed, subjects who prefered the tangible interface felt that it was faster, and vice versa.

In a somewhat surprising result, I had imagined that users with less computer experience might drawn to tangible interfaces. While to a certain extent this was true, the spread in self-assessed expertise between those preferring the TUI and GUI was relatively small. Moreover, of the subjects who rated themselves in the top two tiers of computer expertise, more than half of them preferred the tangible interfaces.

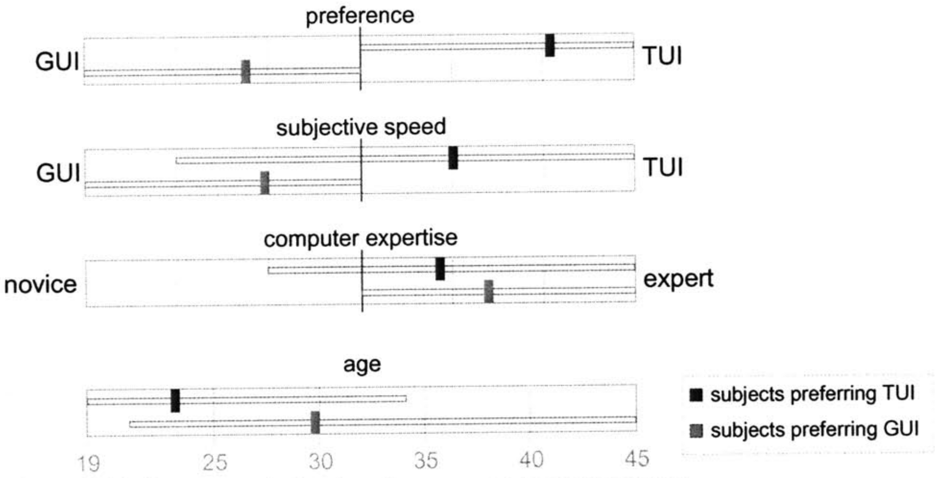


**Figure 6.11: Summary of structured responses to user surveys**

*User responses to each question are divided on the basis of users' interface preference. The red (upper) hatch marks indicate the average ratings of the 8 users who reported an overall preference for the TUI. The blue (lower) hatch marks indicate average ratings for the 7 users who preferred the GUI. Horizontal bars indicate the range of responses.*

Users were also asked to rate the interfaces' ease of learning and use, and the interface's likelihood to support effective interaction with real databases. These results are summarized in Figure 6.12. While these results were not statistically significant, on average users rated the tangible interface as slightly preferable on each of these criteria. I was somewhat surprised and excited by the "effectiveness" result in particular, since the GUI technique used within the study is actively used in commercial practice.

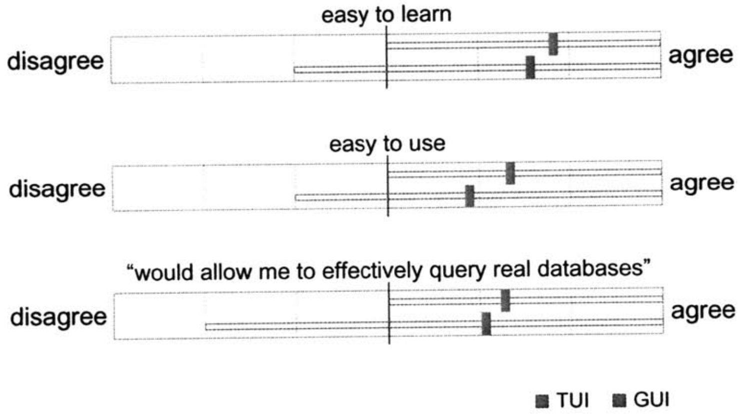


**Figure 6.12: Summary of structured responses to user surveys: comparative GUI/TUI response on selected questions**

*The graph reflects average user response to six different questions (users were asked about their GUI and TUI performance in separate questions).*

#### 6.5.3.4 *Other user comments*

In their free-form comments, users consistently brought up many points that have been referenced elsewhere in the thesis.

Nine users mentioned making use of two-handed interaction with the tangible interface. Example comments include:

- "allows one to control more options at once"
- "ability to modify multiple criteria simultaneously"
- "could tweak things gradually, felt like I had control over variables. Could place wheels in any order (I mirrored the order of the parameters to be extrema-ized)"

Some users mentioned the potential eyes-free aspect of the tangible interfaces, and other kinds of manipulability benefits:

- "clear, easy to use, more intuitive than graphical"
- "didn't have to look at knobs, faster to tweak"
- "ideas of 'increase' and 'decrease' clearer than with the graphical interface"
- "easier to turn the dial (large object) than to position a cursor (small object)"
- "you select the parameters directly instead of having to find them on the screen."

Other subjects commented on corresponding physical drawbacks to interacting with the graphical interface, which are shared with most graphical interfaces:

- "cramped hand on mouse"
- [with the TUI,] "no fumbling with a mouse"
- "grabbing and sliding [with the mouse] is RSI inducing"

On the other hand, seven users mentioned the advantages of familiarity with graphical interfaces. Examples include:

- "it's familiar to lots and lots of people"
- "comfortable and common mouse control"
- "worked like computer interfaces I already know how to use"
- "very familiar with using a mouse"

Some subjects commented on the downside of the TUI setup phase:

- "[weak points of the tangible interface:] the time it took for me to find and place the wheels I needed on the board"
- "[with the GUI,] you don't have to choose the parameter dials"

Some users commented on the need to learn how to operate the TUI:

- "I definitely preferred the tangible interface. I figured it out and mastered the controls quickly (but I already had the graphics controls mastered)."
- "the high tech aspect of [the TUI] makes it more captivating, easier to control, especially once I was familiar with it."

Several users felt that the GUI's static graphical layout was an important strength:

- "you could see all the options in front of you at once"
- "having all the scroll bars visually laid out made it easy to comprehend how the tool worked."
- "I felt it was simpler to keep my selections of price, acreage, etc. stable while viewing the map and my progress bar"

Users had differing issues about the control offered by the wheels vs. the mouse. E.g.:

- TUI: "very precise", "easier to manipulate;" as opposed to…
- "mouse responds more accurately than the turn dials"

Some users felt the TUI delivered too much control:

- " with more practice I imagine one would be quicker with the tangible interface. For beginners, having the control over different options simultaneously can be slightly overwhelming when querying the database. Additionally, I was less likely to look at the numerical values of the quantities (price, acreage, etc.)"
- Another user, speaking with respect to the TUI: "too many things to focus at once can make you lose track"

A final set of comments related to enjoyment aspect of interacting with the tangible interface; e.g.:

- "[the TUI was] more fun, less tiring on the hands than the mouse"

## 6.6 Discussion

### *6.6.1 Evaluation of hypotheses*

As discussed in §6.3, the experiment set out to evaluate three different hypotheses. These were as follows:

**Hypothesis 1:** *Tangible interfaces using physically constrained tokens can provide a feasible approach for expressing simple queries*

In introducing this hypothesis, I described it as a rather simple claim which might be supported through successful execution of the study. In the study, sixteen subjects were presented with a total of 384 tasks, 192 of these involving the parameter wheels query interface. Of these, users successfully completed 189 (98%) of the TUI tasks, including all but one task from the final eight timed questions.

Moreover, users on average preferred using the tangible interfaces, with seven (44%) of the subjects indicating a moderate to strong preference for tangible interface over the comparable GUI (ratings of 6 or 7 on a seven-point Likert scale). Users also gave a slightly higher rating to the TUI over the GUI both for ease of learning, ease of use, and also in its potential for effectively querying real databases. It is also worth repeating that these experiments were conducted by undergraduate assistant Anna Lee, reducing another potential bias.

I believe these observations provide strong confirmation of hypothesis #1.

**Hypothesis 2:** *TUIs facilitate and elicit parallel two-handed interactions within querying tasks.*

Thirteen of the sixteen users were observed using both hands to manipulate the tokens of the tangible interface. In freeform written response about the TUI's strengths, nine users explicitly mentioned two-handed interaction as one of the major benefits of the tangible interface. Seven users explicitly mentioned simultaneously modifying multiple parameters as a valuable property. The possibility of using both hands was mentioned to several of the first seven users, but beyond this, all two-handed usage by subjects was completely unprompted.

These observations provide strong confirmation of hypothesis #2.

**Hypothesis 3:** *TUI is faster than GUI for a range of simple querying tasks.*

The most ambitious hypothesis related to time-based performance over the GUI. While the timing results of four users in the first pilot indicated a 29% raw time-average performance advantage of the TUI over the GUI, the main experiment, with its different task sequence, was unable to repeat this result. Instead, as discussed in §6.5.3, the TUI's raw performance results averaged between 3% and 29% slower than the GUI, depending on the method of aggregation used. When the TUI results were adjusted to exclude their setup phase, their speed on the same tasks ranged from 0.5% slower than the GUI, to 45% faster than the GUI.

The evaluation produced one statistically significant result, and this result does support the time-based performance advantage of tangible interfaces over graphical interfaces under specific conditions. However, these conditions are rather selective, relating to the tangible interface times without their setup phases, and without inclusion of times for tasks #20 and #24.

In short, while selected aspects of the experiment support hypothesis #3, the overall results are inconclusive and do not support this speed-based hypothesis. The discrepancy between the results of the first pilot experiment and the main experiment can be attributed in significant part to differences in experimental procedure. This raises the question of which procedure more accurately reflects the true performance of the two techniques.

The first pilot included a series of back-to-back tasks. As discussed in §6.5.1 and §6.5.2, I believe the original pilot's task sequence both more accurately reflects "typical" usage of the querying techniques, highlighting several TUI strengths, reducing several TUI weaknesses, and highlighting several possible GUI weaknesses. These include the following factors:

1) Back-to-back tasks with some repeated query parameters helps minimize the cost of the TUI's "setup phase."

2) Back-to-back tasks required GUI users to manually "clear" the settings of unused range sliders. This incurs a GUI performance penalty that is more time consuming and distracting than the corresponding removal of parameter wheels from the query rack.

3) Clearing the GUI range sliders also destroys valuable digital state. In comparison, when TUI parameter wheels are returned to the query rack, they retain their previous settings, which are often near the values necessary for answering successive tasks.

4) The first pilot did not include questions similar to the main experiment's #20 and #24, which were especially problematic for TUI users.

At some level, the features of the points #2 and #3 could be supported on the graphical interface with a "disable parameter" function. While I do not know of this feature existing in experimental or mainstream interfaces, its feasibility limits the weight of factors #2 and #3.

The first factor relates to two important issues: what is in fact a "typical" or "representative" task sequence for the tangible query interfaces; and what is the fairest and most accurate way to account for the TUI's setup phase? At some level, these questions are best (and perhaps only) answerable through real-world use of the tangible query interfaces, perhaps accompanied by ethnographic study.

It is also important to note that the experiment involves a significantly "stripped down" version of the main parameter wheel-based tangible interface, which arguably leaves out many of its most powerful features. In the full interface described in Chapter 5, placing parameter wheels upon the query rack specifies not only which parameter is to be queried, but also how and where the results of this query are to be visualized. While a GUI simulation of this same approach is possible, the equivalent functionality has been seen as more difficult to realize in the GUI context (cf. [Ahlberg and Wistrand 1995]). This suggests that the "setup phase" can encompass more salient interactions with the interface than investigated within the experiment.

The fourth factor – whether tasks such as #20 and #24 are exceptional, or whether they reflect accurately on a "typical" kind of task – is again difficult to answer without more "real world" experience with the interfaces. However, these questions suggests at least anecdotally that allowing people to solve certain tasks by literally and figuratively "playing with the knobs" can potentially reduce reflection.

Moreover, the tangible interface's configuration, which made it more difficult to simultaneously view the parameter wheels and display results, probably also played a role in these difficulties. I believe that integrating the parameter wheels with haptic feedback, perhaps in a fashion analogous to the "tagged handles" of [MacLean et al. 2000, Snibbe et al. 2001], could be a powerful added support. Such haptic support could provide mechanical enforcement of lower and upper bounds, haptic indications of query density, and other important kinds of feedback.

### 6.6.2 Issues with parameter bar implementation

The user study examined one of the two query interfaces that were realized: the parameter wheels. No experimental evaluation was made of the parameter bars. Simultaneously, as discussed in §6.2, a major motivation for the design and implementation of the parameter bars was to help conduct user studies.

While a working implementation of parameter bars was realized, in practice they had several properties that made them difficult to employ them in user studies. As discussed in §5.7.2, some of these related to physical form factor and the overall style of interaction. Others related to reliability and performance. While the custom near-field RF communications link was intended to have a communications range of ~1 cm, the realized range was about half this distance. Combined with mechanical issues, this requires parameter bars to be frequently adjusted on the query rack to enable successful pick-up. Moreover, the movement of some parameter bar sliders was a bit "sticky;" and the bandwidth of the parameter bars' communication link (3 to 4 updates per second) was slower than desired. These issues complicate execution of performance-based user studies.

Finally, as discussed in Chapter 5, the parameter wheels seemed a conceptually cleaner implementation, closer to the tangible interface ideal. I believe the parameter bars illustrate one path for "hybrid" approaches that bridge the space between tangible interfaces and traditional handheld computers. At the same time, the parameter bars also seem to indicate some of the potential complexity that this tradeoff can bring about.

### 6.6.3 Closing observations

The experiment took a fairly mainstream approach to evaluation. In the process, from the thesis perspective, many of the most interesting usage dimensions were left unexplored. For example, while the study involved participation from a large range of age and demographics than the local undergraduate population, it did not evaluate use by children or teenagers; by the population of users over 50 years old; or by users with low inclination toward computer use.

In many respects, these are the very base of users who are perhaps most promising for the interfaces of the thesis. A number of subjects commented on the tangible interface being more enjoyable, engaging, and "fun to use" than the graphical interface. I believe that these factors could play an especially strong role for usage by children and teenagers. This intuition is also supported by the success of tangible interfaces in a number of child-use contexts [Perlman 1976; Suzuki and Kato 1993; Glos and Umaschi 1997; Resnick et al. 1998; Ananny 2001]. I also believe that the potential ease of use and "low barrier to entry" benefits of the tangible interface would be attractive both to older populations, and also to populations with less inclination toward computers. However, due to time limitations, these intuitions remain for future evaluation.

Multi-user interaction is another area of key interest that again remains to be evaluated for the thesis interfaces. One of the challenges with multi-user interfaces is determining appropriate metrics for evaluation. While performance time remains one possible metric, it is likely even less appropriate for multi-user interfaces than single-user interfaces. For example, classroom experiences with the urban planning system of [Underkoffler and Ishii 1999] suggest that the interface's strongest role may lie in stimulating and supporting conversations about the problem domain. Here, "speed of completion" could be expected to have an inverse correlation with the quality (or at least length) of associated conversations.

Simultaneously, I believe the scoring approach presented in this chapter provides one potential mechanism for exploring multi-user interaction. I considered evaluating two different approaches. In one variation, two-user teams might work together to complete the same tasks used in this chapter's single-user experiment, both using the TUI and a single-display groupware GUI configuration (e.g., [Hourcade and Bederson 1999]). However, each user could be restricted to manipulate only three of the experiment's six parameters. This approach would allow comparison of completion time both between the two-user and single-user experiments, and between TUI and GUI use. In another variation, two users might be given a common task. Separately, the users might be instructed to optimize several competing criteria, with per-user scoring both on the meeting their individual and shared task objectives.

However, both of these variations have the downside of minimizing the opportunity for meaningful conversation, and seeking a speed result where a TUI speed increase may be unlikely. Indeed, several authors who have evaluated multi-user tangible interfaces, including [Cohen et al. 1999] and [Hornecker 2002], have seen high subjective satisfaction, but without measurable increases in task performance. In addition, to optimize for group usage, parts of the query interfaces would likely benefit from further design iterations (e.g., cf. [Gutwin and Greenberg 1998]). Correspondingly, evaluation of multi-user interactions has again been left to future work.

As one other observation, after seeing the query interfaces, at least three interface experts have independently urged evaluation of the interface on the basis of user retention. These experts believed the tangible interface would strongly facilitate users' memory of the task dataspace. While I believe this is a promising direction, the within-subjects design of the experiment (important for allowing users to subjectively compare the tangible and graphical interfaces) precluded this kind of evaluation for the thesis experiments.

At a higher level, the evaluation of this chapter implicitly assumes that aside from the GUI vs. TUI embodiment, "all else is equal" between the two interfaces. However, there is a great deal of difference to be found in the GUI vs. TUI embodiments, and "all else" is definitely not equal. Perhaps most obviously, the dynamic query technique can be realized on most of the hundreds of millions of preexisting graphical interfaces; the TUI requires special hardware. Also, the TUI's scalability, at least in terms of simultaneously active parameters, is likely considerably more limited than the graphical interface.

Nonetheless, I was pleased by the number and variety of dimensions in which subjects preferred the experiment's tangible interface over the graphical interface, even in the reduced form used for experimental purposes. Moreover, I believe the tangible interface's nearness of performance to the GUI by many metrics, and superiority in performance by several, further establishes it as a viable alternative approach. As discussed in this chapter's opening, the history of graphical interfaces suggests their popularity was not predicated on performance improvements over character-based interfaces, but rather that people enjoyed and appreciated the new way of interacting with and experiencing digital content that it offered. I believe this chapter's evaluation lends support to the belief that tangible interfaces may also find applicability and appreciation by people for use within a variety of interaction contexts.

# chapter 7

# discussion

# 7 Discussion

> *We next went to the School of Languages.... The [second project] was a Scheme for entirely abolishing all Words whatsoever.... since Words are only Names for Things, it would be more convenient for all Men to carry about them, such Things as were necessary to express the particular Business they are to discourse on....*
>
> *However, many of the most Learned and Wise adhere to the New Scheme of expressing themselves by Things, which hath only this Inconvenience attending it, that if a Man's Business be very great, and of various kinds, he must be obliged in Proportion to carry a greater bundle of Things upon his Back, unless he can afford one or two strong Servants to attend him. I have often beheld two of those Sages almost sinking under the Weight of their Packs, like Pedlars among us...*
>
> *But for short Conversations a Man may carry Implements in his Pockets and under his Arms, enough to supply him, and in his House he cannot be at a loss: Therefore the Room where Company meet who practise this Art, is full of all Things ready at Hand, requisite to furnish Matter for this kind of artificial Converse.* —*Gulliver's Travels, Part III, Chapter 5: The Sages of Lagado [Swift 1726]*

This thesis has worked to identify and articulate a thread of research that takes a different paradigm for human engagement with computational systems, making steps toward the usage of physically constrained tokens to facilitate interaction with aggregates of digital information.

This chapter begins by discussing several paths that might extend the expressiveness and range of applications of the physically constrained tokens approach. It then considers some of the alternate design spaces that are opened by reversing some of the TUI properties introduced in §1.2. Next, the chapter explores the challenging question of "what should be physical, and what digital," both in terms of the broad tasks and specific functions that can best profit from physical embodiment. The chapter also considers possible avenues for the use of tangible interfaces in commercial and other applied contexts, and revisits the high-level relationships between the thesis work and several other interface approaches. Finally, the chapter relates a more subjective discussion of the importance of physicality, drawing from personal experiences that helped to motivate the thesis work.

## 7.1 Extensions and variations of the thesis approach

### *7.1.1 Tokens and interpretive constraints: more expressive relationships*

The thesis approach can be extended in a number of different ways. One possible trajectory is to introduce more expressive relationships between tokens and interpretive constraints. Both the mediaBlocks and tangible query interfaces have been built primarily around single kinds of tokens. One prospect lies in creating new kinds of tokens and supporting the composition of more powerful object "sentences."

Aspects of this have been preliminarily explored. For instance, in addition to the mediaBlocks "container" token, an additional kind of "conduit" token was described in §4.10.1. Conduit tokens were conceived as "containing" one or more live, streaming channels of content; e.g., live video or audio. Alternately, the query interfaces implemented a container token for referencing source databases and (potentially) for capturing and manipulating query results.

In the preliminary explorations of these new token concepts, the additional tokens could not be directly "combined" in expressions involving the primary tokens. In the case of mediaBlocks, either a container or a conduit could be present within a slot, but not both; and the interaction of conduits with the sequencer's racks were never well defined. In the query interfaces, the dataset containers and query bars each engaged with separate interpretive constraints, and were made physically "incompatible" with each other.

For the query interfaces, an early prototype iteration did support grammars involving both dataset containers and query bars on the same query rack. While the dataset containers and query bars were intentionally given different physical shapes, they were initially designed to allow their combination on the query racks. The intention was to allow multiple datasets or databases to be combined and simultaneously manipulated, perhaps following the relationship illustrated in Figure 2.20.6. However, it was unclear how to cleanly resolve this extension with the existing adjacency-based Boolean interpretation of the racks. Therefore, this functionality was dropped, and the physical forms were altered to discourage this physical combination.

From a purely syntactic perspective, Figure 2.20 illustrates a number of new grammatical possibilities combining multiple kinds of tokens and racks that remain to future exploration. Where the thesis projects use tokens to represent operands, additional token types could also represent operators, with interpretive constraints used to structure the syntax of this combination.

While the prospect for increasingly open-ended object vocabularies is intuitively compelling, it is important to keep in mind the lessons of Swift's Sages of Lagado. Where tangible interfaces such as the slot machines [Perlman 1976], AlgoBlock [1993], and Programming Bricks [2001] have demonstrated rudimentary object-languages, the Sages suggest the scalability of these approaches are likely to be limited.

Simultaneously, the DataTiles project (in which I was a collaborator) illustrated one approach where the malleability of computer graphics and gesture-based interactions was combined with the physicality of transparent tiles to produce a potentially open ended tile-vocabulary. Here, tiles in many cases were used to represent classes, rather than instances, of operations, thus significantly extending the scalability of this approach. DataTiles are discussed further in §3.5 and pictured in 3.47, while several examples of "tile grammars" are illustrated in Figure 7.1.

It is also worth reiterating that interpretive constraints can be used in combination with other TUI elements and approaches. For example, the TouchCounters system embedded physical storage containers with sensors and local LED displays [Yarin and Ishii 1999]. Shelves with large numbers of these containers were used as a "distributed visualization" for the containers' usage history. Tangible query interfaces could provide a strong approach for querying such a system, with results displayed directly upon the containers.

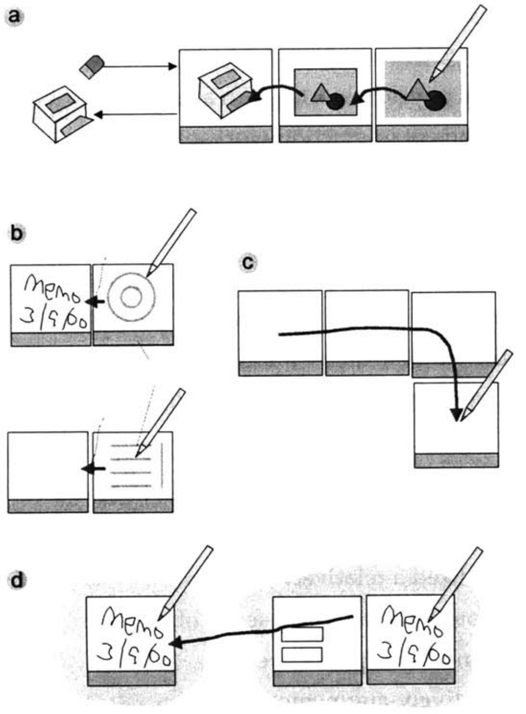


**Figure 5: Examples of tiles and tile combinations. (a) An image from an application tile (right) is stored in a container tile (middle), and then transmitted to the portal tile. The portal tile represents a real world object (a printer in this example). (b) Parameter tiles can be used to specify various types of parameters. (c) Concatenates three video clips and stores item in a container tile. (c) Remote tiles are used to connect distributed tile trays. In this example, a shared drawing environment has been constructed.**

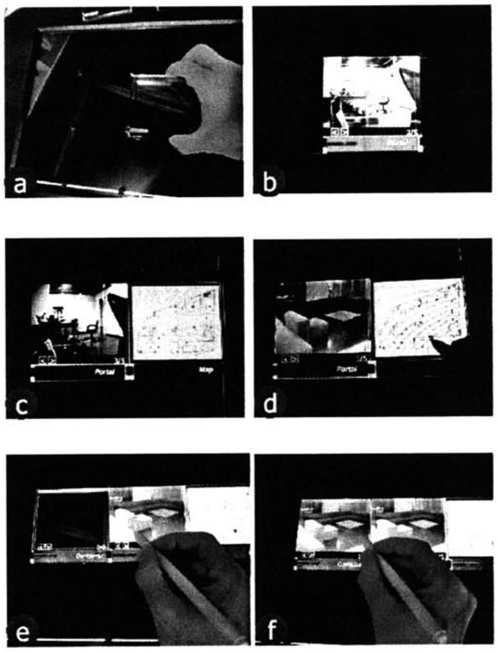


**Figure 6: Examples of tile combination: (a) When a user places a portal tile on the tray, (b) an associated webcam image appears on the tile. (c) Then the user places a map tile, and the map displays locations of webcams. (d) The user clicks on a spot on the map to select another webcam. (e, f) Then the user makes an inter-tile gesture (from portal tile to the container tile) to store a snapshot image in the container tile.**

**Figure 7.1: Illustrations of grammars from the DataTiles system [Rekimoto et al. 2001]**

As another idea, the query interfaces could navigate census information in combination with the Urp urban planning simulator [Underkoffler and Ishii 1999], with query results displayed directly upon Urp's interactive graphical workbench. In yet another variation, the tokens of the Senseboard system [Jacob et al. 2002] lend themselves to describing and manipulating the discrete elements of complex systems (e.g., flow charts). However, Senseboard does not support a mechanism for adjusting continuous parameters that may be associated with these discrete elements. Query rack-style trays could be mounted to the base of the Senseboard, with parameter wheels used to manipulate these continuous values.

### *7.1.2 Integration of interpretive constraints into representational physical structures*

A major theme of tangible interfaces is the physical integration of both representation and control behaviors, blurring the "input/output" distinction that has long dominated human-computer interaction. While parameter wheels employ some level of integration between physical representation and projective display, the mediaBlocks and tangible query interfaces rely heavily on graphical display surfaces for representing the results of tangible interactions.

For more than a year, my thesis work focused on an alternative approach that sought to address this imbalance. Motivated in part by the fabrication discussed in §7.5.2 and §A.4, I sought to push new kinds of interactive information visualizations into the physical world, with a vision of transforming large-scale physical representations into new kinds of interaction workspaces. This effort was called "Strata."

The Strata project developed a series of tangible interfaces for the representation and manipulation of layered information structures. In the end, the Strata effort led to a tension between two distinct ideas: an approach for realizing highly representational tangible interfaces; and an approach for querying that formed the basis of the tangible query interfaces work. Ishii and I decided to focus upon the query interfaces as the project facet that best lent itself to near-term evaluation.

Nonetheless, Strata represented an important component of the thesis efforts to balance between physical and graphical representations of digital information. The mediaBlocks and query interfaces, like many "interactive surface" approaches, realized a relatively flat, "two dimensional" approach to interaction. In contrast, the Strata prototypes layered a series of illuminated 2D surfaces extending into the third dimension. Some of the prototypes incorporated a series of layered flat panel displays; others, projectively-augmented acrylic layers; and in the final prototype, edge-lit acrylic through an array of white LEDs embedded within each layer. Examples of these approaches are illustrated in Figure 7.2.

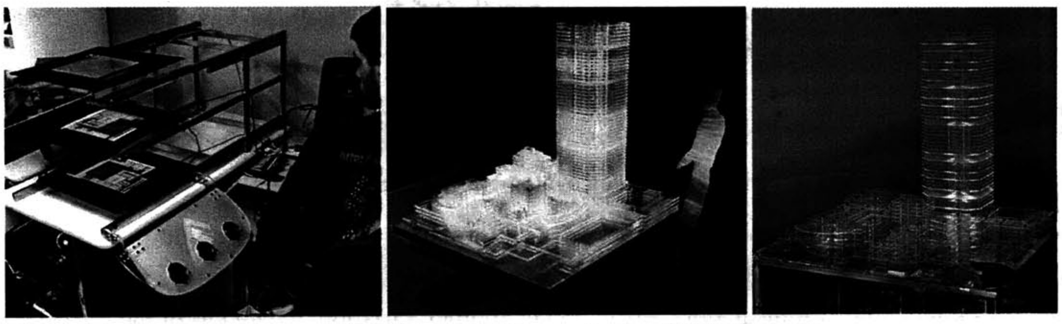


**Figure 7.2: Strata prototypes with LCD, projective, and edge-lit illumination**

The Strata prototypes faced several challenges. First, they were more output-oriented than desired, and too segregated from their mechanisms for interactive control. Secondly, their focus on buildings as a starting domain, while comparable to the initial domain choices of Urp and the metaDESK's "Tangible Geospace," was quite literal. In the absence of a conceptually and visually clean, highly interactive example such as Urp's shadows, this literalness contributed to clouding the project's intended message.

At the same time, Strata made progress toward illustrating several paths for integrating interpretive constraints into more representational physical structures. For example, the primary interactive element of the Strata/ICC installation was a "time wheel." Physical tokens representing building infrastructure could be inserted within this wheel, while the wheel itself could be rotated to select corresponding times of day (Figure 7.3).

Figure 7.3: Content tokens and the time wheel within Strata/ICC

Another variation of the time wheel concept is illustrated in Figure 7.4a. Here, time of day and time of year are represented as concentric ring-like bands, each roughly two meters in diameter. A partial prototype these time-rings was fabricated (Figure 7.4b), which illustrated an interesting property. When the year's 365 days were mapped over a ring of this scale, each day occupied a space several centimeters wide. This suggested that physical tokens might be placed within the ring and manipulated to flag individual days, etc. Smaller scale time rings integrating mechanical constraints were also fabricated (e.g., Figure 7.4c). While conceived to represent time, these ring-shaped interpretive constraints also seemed to have strong potential for domains such as biology; e.g., for representing plasmid rings (used intensively in gene therapy).

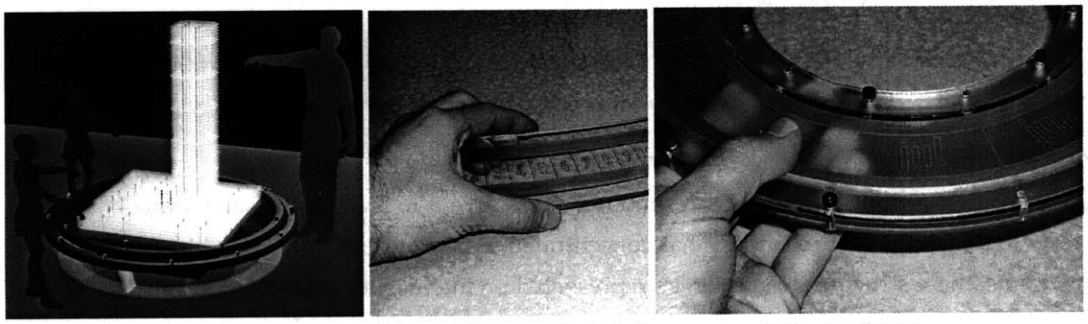


Figure 7.4: Concepts for applying interpretive constraints to representations of time

The concepts of Figure 7.4 remain flat and two dimensional, and again seem to miss some of the potential for tangible interfaces to more fully employ the potentials of 3D physical space. Several further concepts addressing this issue are illustrated in Figure 7.5. Figure 7.5a presents several sketches of an interface for scheduling events at varying scales of time. This concept was most directly in reference to the Unix "cron" daemon, which is used to schedule the execution of Unix processes. Cron is frequently configured to execute recurrent processes one or more times per hour, day, week, month, or year. Figure 7.5a illustrates how physical tokens might be used to represent these processes, and a series of layered time rings might be used to externalize and manipulate a system's configuration.

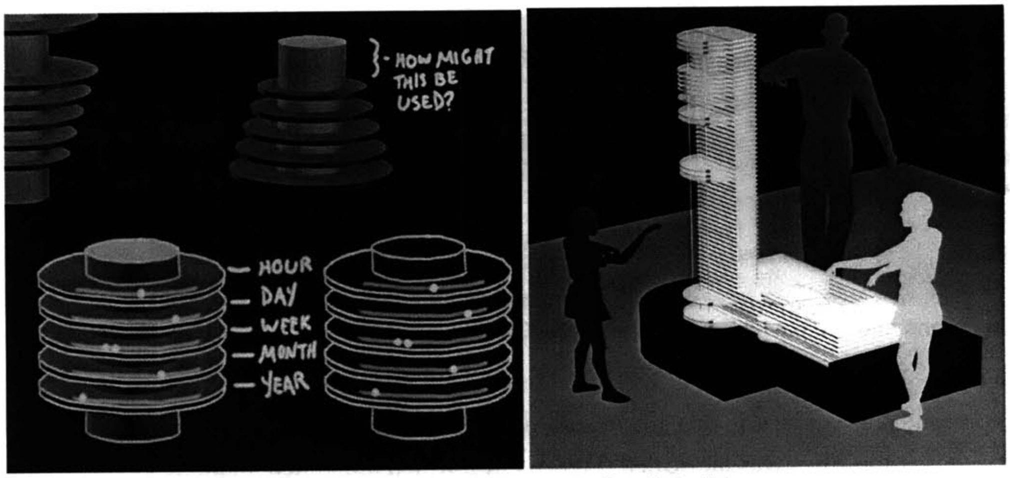


**Figure 7.5: Concepts for extending interpretive constraints into 3D space**

As another concept for balancing between literal and abstract information representations, Figure 7.5b illustrates another variation of the Strata/ICC interface. Here, the building representation is divided into two halves. The right half presents a semi-literal representation of the building. The left half represents a more conceptual representation, which might be used for configuring building-wide simulations or monitoring processes (e.g., trigger conditions for network, power consumption, or temperature "events").

These sketches represent early, partial concepts. However, they suggest paths for integrating interpretive constraints within more highly representational tangible interfaces.

### *7.1.3 Visually structured variations upon interpretive constraints*

Instead of adding increased physical structure, another alternative is to make increased use of graphical representations. Where the thesis has concentrated upon physically embodied interpretive constraints, these structures can often be represented on interactive surfaces in purely graphical form. In a sense, this is a natural relaxation of the interpretive constraint approach, retaining constrained interpretations of (graphically) structured regions, but without mechanical guidance or enforcement.

Several examples of this approach are illustrated in Figure 7.6. The first illustration is an early exploratory sketch from my earliest days of work with the metaDESK. Here, I imagined ringing the desk's perimeter with cells reminiscent of the Monopoly™ board game, while displaying Tangible Geospace's geographical map interpretation in the layout's center. The idea was that placing tokens or cards within these border cells could be used to configure the map visualization, activate the display of different visual overlay layers, and so on.

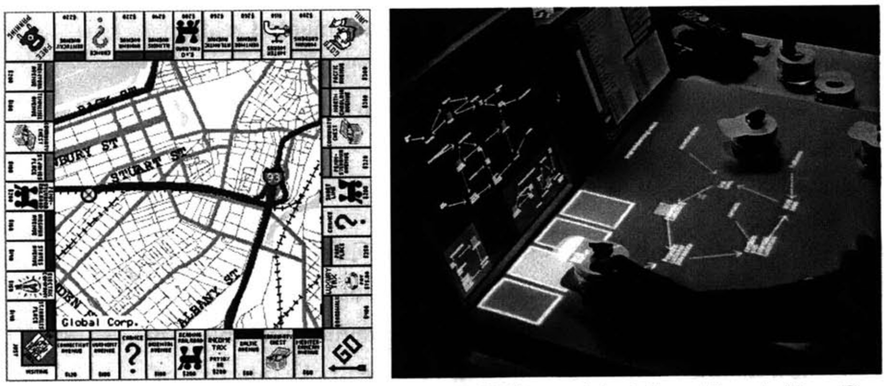


**Figure 7.6a,b: Monopoly mockup on metaDESK; graphical "monitor slot" on Senseboard**

In a more mature implementation, Figure 7.6b shows the use of binding cells for migrating information between an interactive surface and an on-screen display [Patten et al. 2001]. While building conceptually upon the mediaBlocks monitor slot, which utilized a physically-embodied interpretive constraint, the interpretation of Figure 7.6b is likely more practical and flexible for the illustrated screen + workbench configuration.

Realizing interpretive constraints in graphical rather than physical form has several tradeoffs. The costs include:

1) lesser passive haptic feedback;
2) reduced prospects for active force feedback;
3) increased demands for visual attention;
4) decreased kinesthetic awareness and physical legibility;
5) decreased prospects for embedding in niche contexts; and
6) more demanding (and frequently more expensive) sensing requirements.

On the other hand, graphical interpretive constraints can be implemented within "interactive surface" TUIs without adding any additional hardware. Moreover, graphical interpretive constraints benefit from the far greater malleability of the graphical medium, while still leveraging the interface mappings of interpretive constraints.

## 7.2 Implications of other design alternatives

The discussion of §1.2 introduced five properties that are characteristic of tangible interfaces. As noted in §1.3, a number of other related design spaces are highlighted by individually reversing each of these properties. These areas will be explored further in this section.

### *7.2.1 Physically embodied*

As discussed in the first chapter, the broadest property and criteria of tangible interfaces is that digital information or functionality is somehow embodied in physical form. If this criterion is reversed and the other TUI properties are loosely maintained, several adjacent design spaces are highlighted. Virtual reality is broadly concerned with the immersive graphical representation and manipulation of digital information. Artificial realities, as notably explored by Krueger [1983], often provide a "magic mirror" metaphor that "reflect" graphical reinterpretations of physical spaces. Some variations on HMD-based and projective augmented reality support interaction with virtual objects situated within physical spaces. Also, work in haptic holography, such as Plesniak and Pappu's sculpting examples [1998], represents another important class of related research.

### *7.2.2 Physically representational*

The second property of tangible interfaces is the use of artifacts that serve as physical representations (or manifestations) of particular kinds of digital associations. Reversing this property highlights the use of physical/digital artifacts that hold no representational role. In some respects, this can be seen as a manner of degree. For example, the Bricks interface of Fitzmaurice et al. [1995] provides general-purpose physical handles that can be "attached" to virtual objects on a graphical display surface. When removed from the interactive surface, they signify only a generic capacity for control. Therefore, from a tangible interface perspective, these generic handles can be regarded as "non-representational."

Moving even further in this direction, the mouse and other similar input devices are again "non-representational," serving as highly generic tools that can be alternately mapped to virtually every interactive function within a graphical interface. While the mouse mediates control over the GUI's graphical cursor, its function can be equally served by a trackball, joystick, digitizer pen, or other "input peripherals." This invariance differs sharply from tangible interfaces such as Urp, where the interface is closely coupled to the identity and physical configuration of specific, physically representational artifacts.

As discussed in §1.5 and other places, the notion of "representation" is somewhat fluid, and many TUI tangibles can be viewed from a variety of perspectives. For example, much of the thesis work involves physical containers which are neither literally or iconically representational. However, these physical containers are still broadly "representational" in the sense that they are persistently bound to specific elements or aggregates of digital information; and that they serve to mediate a specific set of interactions with this associated information.

### *7.2.3 Physically manipulable*

The third property of tangible interfaces is that their embodied elements are physically manipulable. Reversing this property highlights the space of representational physical/digital artifacts that are not accessible to direct physical manipulation. This class of interfaces has been broadly described as "ambient displays," and is discussed within [Wisneski et al. 1998].

It is worth noting that the manipulability of physical artifacts is in some respects a function of physical proximity, which evolves rapidly in time as people move through space, or exchange control of objects with collaborating partners. In this respect, many approaches for ambient display are also valuable for supporting awareness-at-a-distance of TUI tangibles that are themselves manipulable. Conversely, one of the major challenges with ambient displays is the process of binding them to computational mappings, or probing deeper into the cause of an ambient display's activity. Here, even if the displays themselves are non-manipulable, they may sometimes be strongly complemented with manipulable TUI tangibles that provide interoperability with other tangible or graphical interfaces.

### 7.2.4 Spatially reconfigurable

The fourth property of tangible interfaces is that they are composed of spatially reconfigurable physical elements. Reversing this property yields a space of physically manipulable, spatially fixtured artifacts. In keeping with the description of §1.2.4, this coincides with generations of control panels found in control rooms, cockpits, and other contexts. For the query interfaces, spatially fixturing the parameter wheels and bars yields an approximation of dial boxes and fader boxes, with the added benefit of graphical augmentation and dynamic binding (assuming the latter can be combined with techniques such as Rekimoto's "pick-and-drop" [1997]).

An advantage of spatial fixturing is that control elements are less likely to be misplaced, and their fixed spatial relationships can promote increased kinesthetic awareness. This can be especially advantageous in mission-critical control room contexts, and also in mobile environments (as perhaps indicated from the evolution of the abacus from spatially reconfigurable tokens to beads fixtured upon rods).

However, the spatially fixturing of physical controls compromises or eliminates many of the interface techniques discussed in this thesis. It limits the possibility for composing TUI tangibles into new spatial juxtapositions and relationships, significantly reducing the expressive vocabulary of tangible interfaces. It also limits the possibility for manipulating tangibles in configurations that are "out-of-band" of computational interpretation, which decreases the potential for supporting "epistemic" thinking (as discussed in §2.2.4).

As related in §1.2.4, a corollary of the "spatially reconfigurable" property is that tangible interfaces are generally composed of discrete physical artifacts. Like the term "representational," the meaning of "discrete" is a bit fluid, and worthy of careful consideration. For the large majority of tangible interfaces discussed in Chapter 3, the term needs no clarification. However, several recent tangible interfaces have made strong use of continuously malleable elements of clay [Anderson et al. 2000; Piper et al. 2002].

As discussed in §1.2.4, the intended antonym for "discrete" is "integrated," in the sense that most human-computer interfaces consist of diverse functionality manipulated via a fixed, tightly integrated, and often spatially fixtured set of physical controls. From this perspective, I believe the "continuous" clay elements of interfaces such as Piper, Ratti, and Ishii's Illuminating Clay [2002] can in some senses be considered as "discrete." For example, in Illuminating Clay, the original clay surface is intended to be readily exchanged with other alternative surfaces of LEGO™ and clay, and to be augmented with blocks, cups, and the hands of users. In this sense, while elements of the interface are fluidly continuous in form, they are not spatially fixtured, monolithically integrated components of a general-purpose user interface.

## 7.3 What should be physical, and what digital

One of the "simplest," yet most challenging and important questions that can be asked of work with tangible interfaces is "what should be physical, and what should be digital?" Swift's discussion of the Sages of Lagado, recounted in the chapter's opening, serves as a useful starting point for considering this question.

Clearly, there are limits to the scalability of representing information and concepts with physical objects. The sage's scheme to "abolish all words whatsoever," if words hold the analog of textual and graphical interfaces, is far from the aim of tangible interfaces research. Indeed, to take Swift's discussion more literally as an illustrative example, tangible interfaces seem poorly suited to many contexts involving the creation and manipulation of digital text, which is one of the most central uses of traditional computer interfaces.

To be clear, a number of tangible interfaces have made compelling use of handwriting-annotated paper [Wellner 1993; Mackay et al. 1995; Moran et al. 1999; Stifelman et al. 2001] and sticky notes [McGee et al. 2001; Klemmer et al. 2001]. Such paper-based interactions can lend powerful digital support to equally powerful pre-existing physical work practices, suggesting such uses as a promising avenue for TUI development.

Also, several systems have demonstrated TUI-based manipulation of pre-existing digital text [Small 1996, 1999; Jacob et al. 2002]. But when broadly viewed, the spatial reconfiguration of physical objects seems better suited to the shaping of image and sound, of communications and simulations, than to the processing of words and texts; and this suggests one of many domains where graphical and character-based interfaces may often remain preferable.

### *7.3.1 Key objects of interest*

When considering which elements of a computational system may be best served by physical embodiment, one kind of heuristic is to give physical form to the "key objects of interest." These "key objects" can be seen as the central tools and data abstractions that naturally persist through time and space and elicit active manipulation. If multi-person interactions are to be supported, the key objects of interest may often be those of mutual concern, with manipulation serving to mediate interaction both with the computer and with other people.

It is worth emphasizing the role of "persistence." For better and for worse, even given phase changes and exotic future nanomaterials, physical objects share neither the ability nor predisposition of graphical objects to blink into and out of existence. Correspondingly, TUI tangibles are well suited to offering more stable points of physical control that are amplified through dynamic graphics, audio, and so forth.

One of the many arts of tangible interface design – and an art very much in its infancy – is a careful "impedance matching" between application dynamics and physical embodiment. When an interaction evolves slowly and deliberately, designers may take more liberties in determining the balance of physical and graphical representation.

Conversely, when an interaction evolves rapidly in space and time, more elements of the interaction may lend themselves to visualization using dynamic graphics. Here, physical objects may often serve as containers, parameters, or control points, with visual dynamics born out by intangible graphics. Some of this balance is also likely to change with improvement and dissemination of actuation technology (cf. [Pangaro et al. 2002, Reznick and Canny 2001, Yim et al. 2000, and MacLean et al. 2000]).

The physical materials used within tangible interfaces may also reflect varying interface dynamics and levels of persistence. Materials like corrugated cardboard are inexpensive, environmentally friendly, and afford frequent physical alterations and hand annotations. Alternately, materials like acrylic, foamcore, wood, aluminum, and marble have very different properties, and lend themselves to very different trajectories through time and space.

### *7.3.2 Colocated collaboration*

Another way to approach the question of "what should be physical, and what digital" is by considering the kinds of tasks and functions that may benefit from physical embodiment. Very broadly viewed, tangible interfaces seem to hold strongest relevance for use within situated physical and social contexts, broadly toward the ends of communication and education.

Colocated collaboration is a widely cited benefit of tangible interfaces, which has been demonstrated in work including [Suzuki and Kato 1993; Cohen et al. 1999; Fjeld et al. 2002; and Hornecker 2002]. This aspect has not been directly highlighted by the projects of this thesis, and the physical scale of the thesis prototypes somewhat restricts their suitability for group use. Nonetheless, I believe that the concepts and techniques underlying these systems speak strongly to colocated collaboration, and that this is one of the most promising avenues for future work.

I believe a key aspect of tangible interfaces for colocated use is their "focusing" property, relating to the previous section's discussion of the "key objects of interest." For example, as Durham remarks in the domain of systems design,

> As we teach students in elementary systems design, the crucial thing in all information systems design systems is that they focus attention on and permit the rapid and natural manipulation of the "right things"... Your work is pointed in this crucial direction. [Durham 2002b]

As another example, a number of people have informally reported policies banning the use of laptop computers from a variety of classroom and meeting contexts. As one instance, taken from a syllabus banning laptop use in the classroom [Dunsmoir 2002a,b]:

> [Laptop computers] are excellent tools for individual study but the use of laptops in group meetings has been found to have more negative than positive learning effects, to the extent that large corporations (e.g., IBM) have banned them from meetings and conferences.

I believe this frustration with laptop computers in group situations can be partially attributed to some of computers' strongest strengths. Laptop, palmtop, and wearable computers can be marvelously effective in transporting the user into a separate digital world, often divorcing the user from his physical neighbors and surrounds. The laptop computer and its kin are opaque to those without view of their screens, making it very difficult to determine if the user is diligently taking notes, or escaping into a world of email and web surfing.

In lacking the incredible malleability of graphical and textual interfaces, tangible interfaces help address this kind of usage dilemma. Tangible interfaces tend to limit and channel interaction to a narrower context, and to externalize key elements of representation and interaction in fashions that can be more legible to group members than in GUIs. I believe these properties will help tangible interfaces grow into powerful tools for a range of social contexts, including educational settings, business meetings, scientific collaborations, and many others.

### *7.3.3 Physically situated tasks and contexts*

The notion of *situated* interactions has strongly influenced HCI research through the work of Suchman [1987] and others, with more recent consideration in works such as [Mills and Scholtz 2001]. Websters defines the meaning of "situated" as "having a site, situation, or location" [Websters 2002]. While likely intended in irony, Swift references physically situated use in the chapter's leading quote, saying "in his House [the user of object languages] cannot be at a loss: Therefore the Room where Company meet who practise this Art, is full of all Things ready at Hand, requisite to furnish Matter for this kind of artificial Converse" [Swift 1726].

From the perspective of tangible interfaces, the user's "House" not only includes domestic spaces, but also workplaces, vehicles, and other locales. One of the key properties of physical spaces is the differentiated uses characteristic of different locales. For example, distinct ecologies of objects exist within the kitchen, bathroom, bedroom, office, living room, garage, and other rooms of domestic spaces; and the offices, meeting rooms, laboratories, corridors, auditoriums, and exhibition spaces of workplaces. Each of these locales speaks to a distinct set of physically and socially situated tasks and contexts, which each can lend shape to supporting ecologies of tangible interfaces.

One promising related avenue lies in bringing new computational powers into pre-existing physical interaction contexts. For example, mediaBlocks' transport functionality serves an important role in the context of linking together media devices spread throughout the physical environment. As another example, the TouchCounters system embedded physical storage containers with sensors and local LED displays [Yarin and Ishii 1999]. Shelves with large numbers of these containers were used as a "distributed visualization" for the containers' usage history. Where the TouchCounters system explored a graphical interface for expressing queries over the containers, this had the disadvantage of requiring visual attention on the screen, rather than on the target storage containers. Tangible query interfaces could provide a strong approach for querying such a system, with results displayed directly upon the containers.

### *7.3.4 "Don't be dogmatic"*

In reflecting on lessons learned from the pioneering Xerox Star graphical interface, its creators offered the following caution:

> *Don't be dogmatic about the Desktop metaphor and Direct Manipulation*
>
> Direct Manipulation and the Desktop metaphor aren't the best way to do everything. Remembering and typing is sometimes better than seeing and pointing. For example, if a user wants to open a file that is one of several hundred in a directory (folder), the system should let users type its name rather than forcing them to scroll through the directory trying to spot it so they can select it. [Johnson et al. 1989]

This remark has obvious, and important, relevance to tangible interfaces. Tangible interfaces clearly "aren't the best way to do everything." A number of even more closely relevant lessons can be found from the history of visual languages for graphical interfaces (discussed in §3.x). For example, Citrin [1996] writes:

> Part of the resistance to adoption of visual languages lies in three areas, and this is where strategic research should be targeted. These three areas are (1) appropriateness of mapping, that is, deciding what aspects of problems map naturally into visual representations and what aspects are best left in textual form...
>
> When designing a visual programming language, it is natural for the designer to wish to represent as much as possible by visual means. Unfortunately, this may not yield the most effective languages, and in fact may yield confusing languages that users are reluctant to adopt. ... Heterogeneous visual languages, which mix graphical and textual elements, are a model that should be extensively investigated in the future.

Citrin's question of what should be done visually, and what in text, closely echoes the motivating question of this section. Burnett [1999] offers closely related comments that are both descriptive and prescriptive in relation to tangible interface:

> Many of these early systems had advantages that seemed exciting and intuitive when demonstrated with "toy" programs, but ran into difficult problems when attempts were made to extend them to more realistically-sized programs. These problems led to an early disenchantment with visual programming, causing many to believe that visual programming was inherently unsuited to "real" work – that it was just an academic exercise.
>
> To overcome these problems, visual programming researchers began to develop ways to use visual programming for only selected parts of software development, thereby increasing the number of projects in which visual programming could help. [Burnett 1999]

Both Citrin and Burnett's analyses speak in part to hybrids between tangible, graphical, and textual approaches. Such hybrids find early precedents in the monitor slots of mediaBlocks and the Senseboard of Patten et al. [2001].

In addition to finding lessons from the "recent" history of graphical interfaces and visual languages, there are also strong lessons to be found from some of the oldest recorded languages of human history: the token-based accounting systems introduced in §2.1.3. In analyzing some of the limitations of this approach, Schmandt-Besserat reported:

> …Three-dimensionality gave the [token system] the advantage of being tangible and easy to manipulate. On the other hand, the volume of the tokens constituted a major shortcoming. Although they were small, the counters were also cumbersome when used in large quantities. Consequently, as is illustrated by the small number of tokens held in each [clay] envelope, the system was restricted to keeping track of small amount of goods. The tokens were also difficult to use for permanent records, since a group of small objects can easily be separated and can hardly be kept in a particular order for any length of time. Finally, the system was inefficient because each commodity was expressed by a special token and thus required an ever-growing repertory of counters. In short, because the token system consisted of loose, three-dimensional counters, it was sufficient to record transactions dealing with small quantities of various goods, but ill-suited for communicating more complex messages. [1996, p. 98]

Of course, with the passing of 5000 years much has changed, and the dynamics of computational interpretation and augmentation bring to life many new capabilities and potential applications. Nonetheless, many of the above reservations, together with further characteristics and limitations that appear in Appendix C, still hold relevance and are worthy of consideration.

## 7.4 Applications of tangible interfaces

Tangible interfaces are broadly applicable to a range of applications in science, engineering, commerce, industry, and other professional domains, as well as for uses situated within more personal and domestic contexts. Several example domains are discussed below.

### *Modeling and simulation*

As discussed in §1.1.1 and elsewhere, one of the most popular applications of tangible interfaces has been the use of physical objects to model and simulate various kinds of physical systems. Examples include a number of the interactive workbench systems described in §3.2.1, as well as many of the constructive assembly-style interfaces of §3.2.2.

Where interactive workbenches and constructive assemblies have generally emphasized applications based upon geometrical relationships, the token+constraint systems of this thesis (as well as in prior work) have tended to model more abstract logical relationships. The mediaBlocks and query interfaces systems also can be seen as providing an approach for physically modeling relationships between and across aggregates of information. While not explicitly demonstrated within the thesis, the token+constraint approach also holds the potential to control and steer a variety of simulations.

*Visualization*

As discussed in §2.2, tangible interfaces broadly relate to the intersection of computation and external cognition. As such, they share common ground with the areas of scientific and information visualization [DeFanti and Brown 1991; Card et al. 1999]. TUIs present paths for physically describing the identity of contents to be viewed; the spatial and relational transformation of these contents; and the identity and configuration of associated computational operations. TUIs potentially offer richer representations and broader pathways for input and interactive control than GUIs, trading off increased specialization at the cost of general-purpose flexibility.

A number of tangible interfaces have illustrated properties relating to scientific and information visualization. Particularly suggestive examples include Urp [Underkoffler and Ishii 1999], neurosurgical props [Hinckley et al. 1994], the Universal Constructor and intelligent modeling systems [Frazer 1982, 1995], GDP [Anagnostou et al. 1989], and Tiles [Kramer 1998].

Both the mediaBlocks system and the tangible query interfaces have also explicitly provided interfaces to information and scientific visualizations. These have the visualizations of the mediaBlocks sequencer (including techniques such as the perspective wall of [Mackinlay , as well as the geographical and scatterplot (or starfield) views of the query interfaces.

I have also discussed early work upon more physically representational visualizations in §7.1.2. I am excited by prospects for further development of these techniques towards interfaces for biotechnology, distributed systems, media manipulation, and other domains.

*Information storage, retrieval, and manipulation*

Perhaps the largest class of TUI applications is the use of tangibles as manipulable containers for digital media. Examples include mediaBlocks, tangible query interfaces, DataTiles [Rekimoto et al. 2001], musicBottles [Ishii et al. 1999, 2001], Voice Boxes [Jeremijenko 1996], Triangles [Gorbet et al. 1998], the marble answering machine [Polynor 1995], the Paper Palette [Nelson et al. 1999], LegoWall [Fitzmaurice 1996], InfoBinder [Siio 1995], LogJam [Cohen et al. 1999], ToonTown [Singer et al. 1999], InteractiveDesk [Arai et al. 1995], Passage [Streitz et al. 1999], POEMs [Ullmer 1997], Rosebud [Glos 1997a,b], Sage [Umaschi 1997], and WebStickers [Ljungstrand and Holmquist 1999].

*Systems management, monitoring, configuration, and control*

Several TUIs illustrate the broad capacity for manipulating and controlling complex systems such as video networks, industrial plants, etc. Examples include mediaBlocks, Triangles [Gorbet et al. 1998], LegoWall [Fitzmaurice 1996], Twin Objects [Schäfer et al. 1997], AlgoBlocks [Suzuki and Kato 1993], ToonTown [Cohen et al. 1999], and LogJam [Singer et al. 1999].

*Education*

Another major grouping of TUIs relates to the education domain. Beyond the above simulator examples, related TUIs include the Slot Machine [Perlman 1976], AlgoBlock [Suzuki and Kato

1993], Triangles [Gorbet et al. 1998], and Resnick's longstanding work with digital manipulatives and programmable bricks [Resnick et al. 1998].

Several tangible interfaces have demonstrated techniques for programming algorithmic systems with physical objects, usually in the context of elementary education. Examples include the Slot Machine, AlgoBlock, and programming bricks [McNerney 2000].

*Remote communication and awareness*

Another application domain relates to systems that facilitate remote communication and awareness at the periphery of users' attention. Here, we relax TUI's requirements for physical control, and consider interfaces employing "ambient media" [Wisneski et al. 1998].

Early examples included the Benches system [Dunne and Raby 1994], which coupled physically remote benches through temperature and sound; and Live Wire [Weiser and Brown 1995], which expressed network activity through the spinning of a long "dangling string". Other ambient media examples include the ambientROOM [Wisneski et al. 1998], AROMA [Pedersen and Sokoler 1997], Pinwheels and the Water Lamp [Dahley et al. 1998], digital/physical surrogates [Kuzuoka and Greenberg 1999], and personal ambient displays [Wisneski 1999].

Another kind of interface in the broad domain of remote communications is inTouch [Brave et al. 1998]. The inTouch prototype supports haptic gestural communication between physically remote parties through a "synchronous distributed physical object."

*Entertainment*

As with many new technologies, tangible interfaces have potential in the entertainment domain. Examples include the MusicBlocks [Neurosmith 1998] and Zowie [Francetic and Shwe 2000] products, as well as research systems such as curlybot [Frei et al. 2000], Nami [Heaton et al. 1999, 2000], Triangles [Gorbet et al. 1998], Blocks [Anderson et al. 2000], Digital Manipulatives [Resnick et al. 1998], and Programming Bricks [McNerney 2001].

*Artistic expression*

A number of tangible interfaces have been motivated strongly (or even predominantly) by artistic concerns, bringing to life expressive physical forms with the powers of computational mediation. Examples include Benches [Dunne and Raby 1994], pinwheels [Dahley et al. 1998], musicBottles [Ishii et al. 1999, 2001], Voice Boxes [Jeremijenko 1996], Triangles [Gorbet et al. 1998], and Surface Drawing [Schkolne et al. 2001].

*Spirituality*

Tangible interfaces also have the potential to play a strong role within spiritual contexts and in manners of religious expression. This potential seems especially strong for facilitating communications among physically distributed communities, as mediated through physical artifacts that provide distance from incongruous technology-centric devices. This area of work has been broadly motivated by [Muller et al. 2001], with the Candle Altar [Bayley and White 1997] and AltarNation [Hlubinka et al. 2002] offering early supporting instances.

## 7.5 Paths toward applied use

When topics are pursued within the context of doctoral research, several unusual resources are often present: specialized facilities, expert colleagues, high motivation, and an extended duration of time. In concluding such work, it is interesting to explore possible paths by which this research might find its way into a world of more widespread use.

Tangible interfaces depend centrally upon physical objects. The tangibles of the thesis systems, as well as those of many other tangible interfaces, are specially crafted artifacts embedded with special-purpose electronic components. Thus, where new graphical interface techniques can be realized in their entirety with widespread computer languages and new GUI applications can be rapidly distributed over computer networks, the dissemination of tangible interfaces depend on the fabrication and distribution of new kinds of physical/digital artifacts.

Some of the details of producing these kinds of artifacts are discussed in Chapters 4, 5, and Appendix A. This section considers some of the higher-level implications of TUI fabrication upon prospects for more widespread creation, dissemination, and use of tangible interfaces.

### *7.5.1 Mass production*

Over the past half-century, digital computing technology has yielded a steady progression of ever smaller, cheaper, and more highly integrated devices, at ever larger scales of mass production and consumption. One possible trajectory for tangible interfaces is to join this trend. Indeed, two of the first commercial products to incorporate tangible interfaces – Neurosmith's Music Blocks and Zowie Interactive's play sets (both commercially released in 1999) – are toys that retail for on the order of US$50 [Neurosmith 1999; Francetic and Shwe 2000].

Several technology trends support the mass production of tangible interfaces. Perhaps the broadest of these is the burgeoning increase of devices embedded with computation. As these devices come to support more complex behaviors, this functionality is most often exposed to users in the form of graphical interfaces. The trend toward increasingly numerous devices is paralleled by an increasing breadth of electronic content offerings (both in audio, video, and textual form), and by increasing penetration and bandwidths of wired and wireless networks.

As noted in the discussion of mediaBlocks, the prevailing methods for interacting with networked content are the graphical traversal of hyperlinks and the typing of keywords or URLs. Especially when framed in the context of embedded devices with small displays and limited input modalities, combined with user expectations for simple operation, both of these alternatives present major navigational and usability challenges.

Taken together, these device, content, and connectivity trends present both a critical mass and strong motivation for developing alternative interaction approaches. Tangible interfaces offer a promising family of candidates.

The mass production of tangible interfaces is also supported by several more specific technology trends. One is the increasing use and diminishing costs of tagging technologies. Of special relevance is wireless RFID technology. Among many other application areas, this technology is being embraced as a digitally interrogable replacement for barcode UPC labels, which might be pervasively embedded in a wide range of commercial products [Brock 2001].

These tagging technologies, combined with pervasive networking, are the core underlying technologies behind the mediaBlocks and tangible query interfaces prototypes. (The parameter wheels use RFID tags, while mediaBlocks use a "1-Wire digital serial number" tag.) The increasing deployment, lowering cost, and increasing standardization of these tags, combined with similar trends for wireless networking, provide a powerful foundation for the growth of tangible interfaces.

Another trend favorable to tangible interfaces is the growth of memory and function modules such as CompactFlash (`www.compactflash.org`) and MemoryStick (`www.memorystick.org`). Introduced in 1994 and 1998, respectively, these new digital mediums have grown dramatically in usage alongside the rapid adoption of digital camera, and to a lesser extent, MP3 music players.

The Memory Stick in particular, intended for "linking [a] variety of devices, mainly among AV devices" [Sony 2002], shares common ground with the mediaBlocks system (developed in 1997). As a key difference, where the thesis systems depend upon both "associate" and "manipulate" phases of interaction (§1.1.2), basic Memory Sticks support only the "associate" phase.

One issue is that Memory Sticks currently rely on electromechanical connectors. While their 10-pin serial connectors are a better match for tangible interfaces than the 50-pin ATA-protocol CompactFlash connectors, electromechanical connectors consistently have been found unreliable in tangible interfaces supporting a manipulation phase (§A.1.2). However, future Memory Sticks could be embedded with dual-ported RFID technologies such as [IBM 2000, Atmel 2002], which could significantly aid the use of Memory Sticks within tangible interfaces. The "Magic Gate" variation of Memory Sticks also includes a unique ID, which could aid in associating Memory Sticks with online information (another central feature of mediaBlocks).

Both in conjunction with these trends and on the strength of their merits, I believe tangible interfaces will grow to find a significant mass-market presence in the coming years. Building upon the above trends and the eventual forces of standardization, one of the most exciting prospects for tangible interfaces lies in the emergence of ecologies of physical/digital objects. This will be discussed further in §8.3.1.

Both mediaBlocks and tangible query interfaces offer examples of systems that could potentially be developed into mass-market products. However, these systems have developed thus far along a very different pathway, which also holds considerable promise for future growth and development: that of craft production.

### 7.5.2 *Craft production*

The production of physical objects is among the earliest of human technologies. However, in the process of moving toward ever smaller size, cheaper cost, and higher integration, many computing components have passed a threshold, and no longer are accessible to fabrication by human hands. For example, many microprocessors and communications components are only available in "ball grid array" packaging, which cannot be attached with manual soldering.

These factors significantly reduce the ability of individuals to create or repair mainstream computer-based devices, largely restricting their creation and production to commercial organizations. When these devices begin to malfunction, increasingly the preferred approach is to throw them away and buy another.

In addition to the production of electronics, tangible interfaces also require the design and fabrication of physical elements. While a somewhat lower barrier to many than the production of electronics, the making of objects is itself an art and craft that traditionally requires significant time, commitment, and resources to master.

#### *7.5.2.1 New components*

Several recent technological advances have begun to counterbalance the above trends. These technologies are making the implementation of both prototype and final-form tangible interfaces widely accessible to individuals even with limited resources and prior experience.

On the electronics front, integrated components and hybrid modules have significantly reduced the "barrier to entry." While references to specific technologies grow quickly outdated, several examples of year 2000-vintage technology include:

- Microchip's PIC 12CXXX family of microcontrollers are available in pea-sized, easy-to-solder (DIP) form. Costing as little as US$1 in single-unit quantities, they can be programmed with the C language and a <$100 programmer, include an analog-to-digital converter, and require no external components other than a power source to function.
- IB Technology's Micro RWD devices provide a fully-integrated RFID reader/writer for roughly US$30 (single quantities) in the size of a postage stamp. Again available in DIP form, these hybrid modules require only an antenna and a power source to function.

These devices allow simple versions of the mediaBlocks and query interface systems to be implemented for <US$50 with as little as two or three components, a dozen wires, a few RFID tags, and (optionally) an entry-level computer. The supporting computer can be replaced with a US$50 3x10cm "TINI" computer / module from Dallas Semiconductor or similarly sized Linux devices, in each case provide Java language support and Ethernet connectivity for communicating with networked devices.

These examples suggest that the electronic aspect of tangible interfaces, while not completely trivial, are becoming accessible for implementing TUIs even by people with minimal electronics experience. Similarly, physical fabrication tools are also beginning to change the nature of manufacturing physical objects.

#### *7.5.2.2 New fabrication technologies*

Manual tools like the saw, drill, and hammer remain powerful instruments for engaging with physical materials. At the same time, a new generation of computer-controlled tools are fundamentally transforming the prospects for "personal fabrication," bringing powerful abilities for fabricating both prototype and final-form physical objects [Gershenfeld 1999]. To again cite several specific examples of year 2000-vintage technology:

- Universal Laser Systems' C200 laser cutter is a desktop-sized system costing as little as US$12K. It supports computer-controlled cutting of a wide variety of materials, including wood, acrylic, paper, cardboard, and rubber, and supports a wide variety of software.
- Roland's Modela MDX-15/20 mini-mill is a milling machine costing as little as US$2K, which is small enough to be carried in a backpack. It supports 2D and 3D computer-controlled cutting and scanning of many materials, and is controlled by simple PC software.

These two tools were used extensively in the final thesis project. The large majority of the mechanical fabrication of the query interfaces was designed using a consumer-level drawing program (CorelDRAW™), and executed using a Universal laser cutter. Additionally, all of the query interfaces' printed circuit boards (the carrier and interconnect boards for electronic components) were fabricated using a Roland mini-mill.

Once again, the identity and details of these specific tools are not of critical importance. Rather, they serve as examples and "existence proofs" for powerful capabilities that can be expected to become increasingly common, significantly easing the process of creating tangible interfaces.

#### *7.5.2.3 New prospects*

The existence and relatively inexpensive availability of the above fabrication technologies has a number of important implications for the future of tangible interfaces. First, these technologies increase the accessibility of tangible interface techniques to researchers and designers from many different backgrounds. While TUI instances extend back decades in time, a great deal remains to be learned and articulated about the principles and process of designing tangible interfaces, and many research questions remain open. The participation and efforts of an active research and design community are critical to the growth and maturing of tangible interfaces.

Secondly, these fabrication techniques enable the potential emergence for new "craft"-based forms of commercial production. Frazer describes this prospect as the "electronic craftsman:"

> The direct relationship between the designer at the computer console and the computer-controlled means of production potentially [enables]... a return to one-off craft technology, but with all the capability of the precision machine tool.... [Frazer 1995].

For more than two centuries, consumers have lived increasingly immersed in a world of mass production, where products exist only if thousands or millions of people are willing to invest in the same artifact. Customization has often been available – the choice of red or blue, leather or vinyl – but the base artifacts remain the same.

The coevolution of new fabrication technologies and tangible interfaces has the potential to change this. As a simple example, for many years consumers took their photographic films to photo developers, and returned hours or days later for their "prints." Similarly, for those with (e.g.) home automation systems, one could imagine people dispatching their physical or electronic blueprints to a "print shop," and receiving back a transparent physical model of their homes (perhaps analogous to the Strata creations of §7.1.2). These artifacts might act as displays for monitoring home systems over varying timescales, and as controls for physically querying and manipulating information within the home context.

While this example is both speculative and simultaneously limited in ambition, the value of this kind of technique for the sciences has been noted for at least fifty years, if not centuries or millennia longer. Writing in the "Double Helix," Watson discussed the crucial role of physical fabrication to the discovery of DNA. Watson described bringing sketches of amino acids to his machine shop collaborators, and waiting for days in anticipation of the machined metal pieces so that these could be used toward solving the three dimensional puzzle that grew into their double helix proposal. When their seminal article was published in Nature, Watson wrote of finally receiving "appreciation that our past hooting about model building represented a serious approach to science" [Watson 1968].

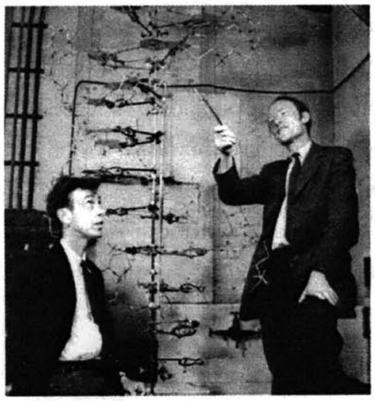


**Figure 7.7: Watson and Crick with the double helix [Watson 1968]**

Today, three-dimensional graphics play a strong role for these kinds of exploration. Nonetheless, as teams of scientists and engineers increasingly struggle with large-scale problems at the bounds of comprehensibility, I believe that new physical approaches to the representation and manipulation of diverse subject domains could have powerful potential. Perhaps no less important, for the home hobbyist and woodworker with digital content of personal significance, the craft production of tangible interfaces also offers a strong path for development, perhaps building upon examples such as the musicbox [Ullmer and Ishii 1999b]. I believe that tangible interfaces have [major implications for both.]

## 7.6 Broader relationships to other interface paradigms

The conceptual background and related work chapters have described some of the background and relationships between tangible interfaces and other approaches in some depth. Given the additional perspectives from the rest of the dissertation, some of these relationships are useful to revisit at a high level.

### *7.6.1 Other tangible interfaces*

In the introduction chapter, the physically constrained tokens approach was described as a kind of middle ground between the interactive surface and constructive assembly approaches. Also, the physically constrained tokens approach was described as building upon two phases of interaction – *associate* and *manipulate* – which corresponded to discrete and continuous modes of interaction.

This balance and tension between discrete and continuous modes of interaction within TUIs can be represented graphically, offering insights on the relationships and tradeoffs between what can be regarded as a continuum between these approaches. This is illustrated in Figure 7.8. Increases in the Y axis roughly correspond to increasing manipulability and physical degrees of freedom, while increases along the X axis map to a heightened presence of mechanical constraints within the interface.

Objects held and manipulated in free space (without any mechanical support) are usually described as having six continuously manipulable degrees of freedom (three translational and three rotational). Correspondingly, two objects held in the hands (as in the neurosurgical props of Hinckley et al. [1994]) offer twelve degrees of freedom, offering a fairly high level of continuously manipulability. However, in the absence of a means of attachment, support, or stable reference, such interfaces can face difficulty in expressing delimiters, distinct commands, stable relationships, or other properties that are associated with discrete relationships. Moreover, it is difficult to hold and independently manipulate more than two objects without taking advantage of more constrained forms of support (a table, pockets, etc.).

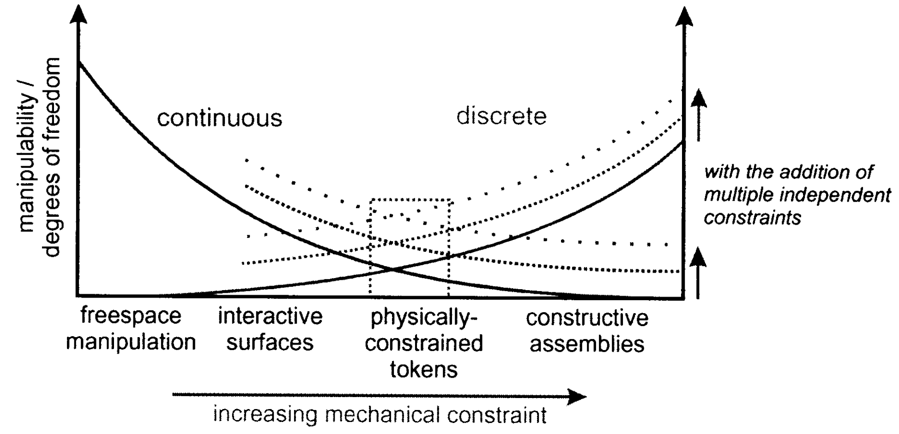


**Figure 7.8: Relationship between continuous and discrete manipulability within tangible interfaces**

On the other end of the spectrum, TUI instances of constructive assemblies have tended to involve purely discrete combinations of numerous modular elements (as discussed in §3.4). However, in almost all such examples to date, the continuous manipulation of physical elements are either disallowed or left uninterpreted (as in the Triangles of Gorbet et al. [1998]).

In order to leverage the strengths of both techniques, it is often beneficial to support both continuous and discrete styles of manipulation. Both interactive surfaces and physically constrained tokens offer this balance at differing levels. Objects upon an interactive surface typically have two translational degrees of freedom, and either one or zero degrees of rotational freedom. If the surface is treated as a continuum, each object can also be regarded as having a single discrete degree of freedom – presence or absence from the surface[1].

In contrast, physically constrained tokens offer fewer continuous degrees of freedom. In return, they also tend to offer increased independence for the mapping of these continuous dimensions, and also increased discrete degrees of freedom. In particular, the constraint structures of the token+constraint approach generally restrict the manipulation of tokens to a single continuous degree of freedom. However, as illustrated in Figure 7.9, token+constraint approaches tend to leverage both a multiplicity of constraint structures, as well as nested token/constraint constructs. From the perspective of Figure 7.8, this has the effect of elevating both the discrete and continuous manipulability of these systems to higher contours.

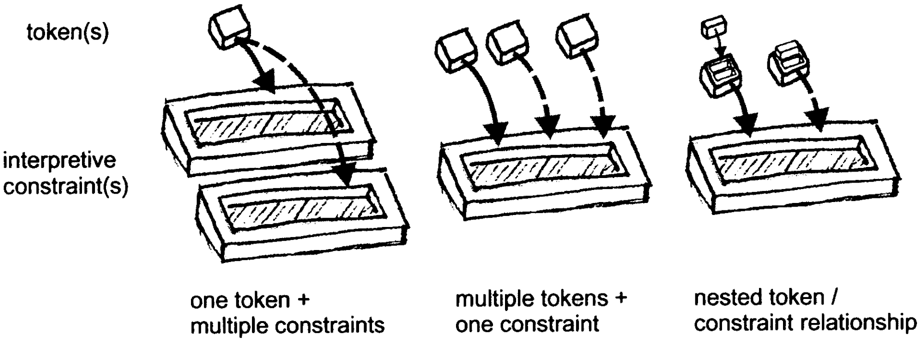


**Figure 7.9 (repeat of Figure 1.5): More complex combinations of tokens and constraints:** *one token + multiple separate constraints; multiple tokens + a single constraint; nested token/constraint relationships*

As illustrated in Figure 7.8, the point of this discussion is that the token+constraints approach offers a kind of balance of both continuous and discrete styles of manipulation. Figure 7.8

---

[1] Some tokens of the Bricks [Fitzmaurize et al. 1994] and SenseTable [Patten et al. 2001] systems have also integrated buttons, introducing a second discrete degree of freedom for these objects. Some SenseTable pucks also integrated support for both discrete and continuously adjustable modifier tokens, thus adding additional continuous and discrete degrees of freedom.

also illustrates a kind of continuum that spans interactive surfaces, physically constrained tokens, and constructive assemblies. Interactive surface systems can offer increased expressiveness through the use of both "hard" (mechanical) and "soft" (graphical) interpretive constraints. This approach has seen early development within the Bricks [Fitzmaurice et al. 1995], Sensetable [Patten et al. 2001], and Audio Pad [Patten et al. 2002] systems. Constructive systems also have the potential for integrating additional continuous degrees of freedom, though the effective mapping of such parameters may be non-trivial.

Finally, it is worth noting that the use of force feedback offers the benefit of software control over mechanical linkages. This can allow linkages to be dynamically transformed between "discrete" and "continuous" controls. This approach has been illustrated by work such as the tagged handles of MacLean et al. [2000], which also highlighted the integration of discrete and continuous manipulation approaches.

### *7.6.2 Graphical user interfaces*

The tangible interfaces of this thesis have made extensive use of computer graphics. While the distinctions of the thesis work from earlier interface approaches have been discussed in Chapter 1 and §7.2, these are worth revisiting. Given the heavy use of computer graphics within the thesis, are tangible interfaces kinds of graphical interfaces? Conversely, are graphical interfaces kinds of tangible interfaces?

I believe the answer to both of these questions is "no." GUIs are built upon engagement with interactive elements that are represented through purely graphical means – e.g., graphical windows, icons, and widgets such as buttons and sliders. These graphical elements may be accessed through the manipulation of one or more mice, styli, touch screens, or other input devices. However, there is never a question of whether these input devices "are" the GUI elements in question; they are purely the mechanisms *through* which the intangible elements of GUIs are made accessible to human manipulation.

The physical artifacts of tangible interfaces go beyond input devices through which the "real" graphical functionality is accessed. Instead, these physical elements are both the fundamental representations and controls through which the interface's functionality is presented and rendered manipulable to the user. TUIs physically expose their language of interaction, making the tacit promise that manipulation of each tangible component will be met with a corresponding computationally mediated response.

Visual representations do play a major role within tangible interfaces in general and the thesis systems in particular. This has also been true in many other interface approaches. For example, mechanical interfaces make use of visual labels, legends, and illuminated indicators. As another variation, many early graphics workstations had two display screens: a non-interactive graphical display screen, and an interactive text-only screen through which graphics could be invoked and textually manipulated. Even though both of these interface approaches make use of "graphics," neither are GUIs.

### *7.6.3 "Invisible interfaces"*

Moving forward, it is useful to consider the relationship of tangible interfaces to several possible trajectories for the future of user interface design. One such prospect relates to the idea of "invisibility," sometimes framed in terms of "invisible interfaces" [Weiser 1994; Fishkin et al. 1998], "invisible computers" [Norman 1998], or "disappearing computers" [Streitz 2001].

In some respects, the high-level goals of these programmes hold much in common with the concept of tangible interfaces. All speak to a decreasing emphasis upon the computer itself (as the stereotypical box containing microprocessor, RAM, and fixed storage); to continuing decreases in the size and cost of computing devices; and to the greater pervasiveness of computing capabilities throughout the physical environment.

In other respects, the TUI design philosophy is somewhat different. In particular, the idea of "tangibility" highlights the elements of these interfaces that are materially present – the physical artifacts that serve as embodiments of digital information and functionality. In referring to the absence (rather than presence) of the subject of interest, Dourish suggests the "invisibility" term is somewhat problematic:

> Invisibility has been held out as a laudable goal for interface design by researchers and interface design critics from a variety of backgrounds. There are various ways, though, in which it is a problematic term. One is that it appears to be at odds with another widely recognised feature of modern HCI practice, the influence of design....
>
> A key [tenet] of interaction design [is that] design should communicate....
> The design perspective seeks to find a new level of engagement between system and user. It reflects an attempt to make interaction engaging and marks a transition from thinking about the user "interface" to thinking about the user "experience." But you cannot be engaged with something that essentially isn't there. Invisibility is not engaging; invisibility does not communicate. Invisibility and the design influence are somewhat at odds. [Dourish 2001]

To be fair, users of the "invisibility" and "disappearance" terms are often oriented toward notions about transparency of usage, as in Heidegger's concept of "present-at-hand." Here, the concept is that when in use, tools such as a hammer – or for that matter, a keyboard or mouse – cease to function as separate objects, and serve instead as an extension of the hand and body [Winograd and Flores 1986; Dourish 2001]. Streitz, for example, highlights this kind of distinction in his discussion of "mental vs. physical disappearance" [2001].

However, Dourish argues that this kind of example does not reflect "invisibility" at all:

> In Heidegger's "hammer" example, the hammer does not become invisible when it is in use. Certainly, it withdraws into a larger frame of activity; but it is, critically, present in the way in which I can act through it and feel through it. Invisibility would be a curse in a situation like that; an invisible artifact, one that does not impinge on my world at all, is not a tool I can use effectively for anything. [Dourish 2001, p. 202]

There are also different kinds of arguments against "invisibility" as an ultimate interface ideal. For example, in referring to people's perception of ambient phenomena such as the temperature of a room, Heschong writes:

> We need an object for our affections, something identifiable on which to focus attention. If there is something very individual and particular that we consider responsible for our well-being – a stove that is the source of all warmth, for example – we can focus appreciation for our comfort on that one thing. But if nothing seems to be responsible for our sense of well-being, then what or whom do we thank? On a lovely spring day we may identify the season itself with our wonderful sense of well-being.... On a tropical isle,... we would probably come to love the island.... But in a typical office building, to what can we attribute the all-pervasive comfort of 70°F, 50 per cent relative humidity? The air diffuser hidden in the hung ceiling panels? The maintenance personnel who work during off-hours? The mechanical equipment down in the basement below the parking garage? [1979]

Projected into the context of human-computer interaction, terms like "well-being" and "object[s] for our affection" may sound ironic; these are terms rarely used in the company of computers. However, implicit in this are several important messages.

First, human-computer interactions often experience "breakdown" – whether in the all-too-familiar technological sense, or Heidegger's more philosophical sense of interactions that cease to respond as anticipated. If the interface is to be "invisible," how, where, what, or to whom should users turn to seek a remedy? Simply saying that interfaces should not break is not productive; even in the most idealistic sense, Heidegger might suggest that "breakdown" is a natural component of use.

However, within the "black box" of computers, "broken code" is typically beyond the reach of human hands; and for an "invisible computer," there is perhaps literally no entity to which hands or eyes can turn. In contrast, Maeda frequently asks what a tangible interface might look like when it breaks [Maeda 2002]. Here, he points to the ways in which the breakage of well-made physical artifacts can expose their problem elements to inspection and repair, even by nonspecialists. Supporting this kind of interaction is a substantial design challenge for TUIs; but it is also an approachable and worthy one, and one that is already beginning to be addressed [Resnick et al. 2000].

The second potential irony relates to the idea that people could come to express genuine affection for an object associated with computation. I would argue that this need not be ironic at all, and elaborate on this subject of §7.7. There are certainly those who would argue that the calm, generally reliable efficiency of Heschong's air conditioner is a kind of ideal for computational systems; and for some users in some contexts this is most likely true. However, especially in contexts such as domestic interactions that are often the subject of "invisible computing" discussion, I believe Redström's question of "how we will *live* with, and not only use, computers" [2001] hints at some of the compelling prospects of tangibility.

### 7.6.4 Artificial intelligence

Another prospect for future user interfaces relates to potential advancements in artificial intelligence. As discussed in §2.5.1, tangible interfaces tend to provide a direct manipulation style of interaction. Elaborating on the role of direct manipulation in the Xerox Star's design, Johnson and Roberts wrote:

> Systems having direct-manipulation user interfaces encourage users to think of them as tools rather than as assistants, agents, or coworkers. Natural-language user interfaces, which are inherently indirect, encourage the reverse. As direct-manipulation interfaces become more prevalent and as progress is made in natural-language understanding and generation, it will be interesting to see which way users prefer to think about their computers. [1989]

Similarly, TUIs can be said to cultivate tool-like, rather than language-like, modalities of interaction. Tangible interfaces may sometimes leverage back-end functionality employing "artificial intelligence" techniques. For example, physical tokens might be used to represent various kinds of automation (e.g., "sleeper searches," or large-scale computations), which might be activated or steered through physical manipulation of the host artifacts. Also, systems like Rasa have already demonstrated the use of speech interfaces in combination with the manipulation of physical artifacts [McGee and Cohen 2001].

Nonetheless, the fundamental emphasis of tangible interfaces is upon human engagement with physical artifacts that deeply intermingle the roles of representation and control, and highlight the causal relationship between manual actions and computational responses. Correspondingly, both as a matter of pragmatics and philosophy, I believe that TUIs will continue to emphasize and celebrate the role of human control and expressive manipulation.

## 7.7 Being physical

As work in support of a doctoral thesis, this dissertation has sought to deliver an objective presentation of the thesis research. In this section, I will briefly describe some of the more subjective personal experiences and beliefs that have motivated these efforts.

In 1995, MIT Media Lab co-founder Nicholas Negroponte published his best-selling book "Being Digital." The text speaks compellingly of the transformative power of digital representations upon our engagement with the world.

In 1996, I had the wonderful opportunity to travel with Negroponte and Ishii in Japan. In one of many memorable experiences, I was struck by Negroponte's comment that cooking was one of his favorite personal hobbies.

In 1997, I had another chance to interact with Negroponte and Ishii, this time in writing a short column [Negroponte et al. 1997]. In ending the text, we revisited the kitchen:

> Year 2020, the curtain opens.... To view the true state of the art, we visit a
> Sicilian kitchen, and look to the center table – only to find...

> Bread. Pasta. Olive oil, and an overripe tomato. Perhaps the bread knives are edged with never-dulling nanoceramic and the oven is fusion-fired. The only glass screens in the kitchen are found on the oven door and garden window (both nanocleaning, of course). The only keyboard is on the faux-vintage typewriter, and all the mice play tag with the cats.
>
> The Sicilian kitchen of 2020 is digital, of course, but it is also intimately and inescapably physical. While a few frantic folk and workers of the midnight hour subsist on energy pills, the Sicilians take pleasure in their food, and embrace the physicality of its substance and preparation. [Negroponte et al. 1997]

The Sicilian kitchen is not a place of instant rice and microwave dinners, but of visceral engagement with once and presently living materials that are raw, moist, uneven, uncooked. It is a place of pleasure in interacting with physical tools – sharp knives, good pans, a responsive range. It is a space of near-unparalleled sensory experience – of sight, sound, taste, and smell, and also, critically, of touch and physical engagement. It is a space concerned with subtlety – spices are an art, and can easily overwhelm and destroy – but also with power and sometimes danger. Sharp blades, fire, and real chemistry are at work.

The Sicilian kitchen also has a temporal dimension. It is not a place of fast food; it is a space of slowness, of long and hard physical labor. To respect the food and this effort, its consumption is also not to be rushed, but to be shared and enjoyed with family through a fullness of time.

The Sicilian kitchen is about being physical.

### 7.7.1 Physical performance

For more than five years, my research focused largely on 3D computer graphics, toward the ends of representing and interacting with large aggregates of information in graphical form. I worked briefly in NCSA's Virtual Reality lab, developed 3D interfaces at Interval Research, and joined MIT's Media Lab to pursue this direction. However, after first beginning to use and develop applications in the "goggles and gloves" paradigm of virtual reality, I began to feel that something was missing.

Virtual reality pioneers Zimmerman and Lanier have described the creation of an "air guitar" that "really worked" as one of their motivating inspirations [Hayward 1993]. I have never played the guitar; but I have played the piano and trombone since childhood, and was convinced that neither of these instruments could ever be done justice in midair with a DataGlove.

A great deal of music making lies in the subtleties of physical engagement with the instrument. In once trying to play two piano pieces, I could plausibly render the Joplin rag on an old upright, and Rachmaninoff on the Media Lab's Bosendorfer. But when trying each piece on the other piano, I as struck that there were passages of each that I was no longer physically capable of playing.

In one sense, this corresponded to the origins and nature of the two pieces. But while neither the upright and Bosendorfer had a greater claim to the name of "piano," the difference of their mechanical actions (forgetting for the moment their acoustics) made each capable of – and limited to – mediating very different human performances. Neither the instrument nor performer could truly stand alone (regardless of the Bosendorfer's digital instrumentations); and surely neither could be replaced by fingers dancing in air, on a table, or on a QWERTY keyboard.

This rift between physical performance and musical evocation holds not only for "classical" musical instruments with deep roots in physical mediums, but also in rendering very recent genres of electronic music. In introducing their tangible interface for the performance of electronic music, Patten, Recht, and Ishii write:

> The late nineties saw the emergence of a new musical performance paradigm. Sitting behind the glowing LCDs on their laptops, electronic musicians could play their music in front of audiences without bringing a truckload of synthesizers and patch cables. However, the transition to laptop based performance created a rift between the performer and the audience as there was almost no stage presence for an onlooker to latch on to. Furthermore, the performers lost much of the real-time expressive power of traditional analog instruments. Their on-the-fly arrangements relied on inputs from their laptop keyboards and therefore lacked nuance, finesse, and improvisational capabilities. [Patten et al. 2002]

Maeda has expressed similar frustrations in a domain as far removed as the expression and debugging of software code [2002]. Maeda related the experience of watching three of his star students the night before a major gallery opening. Each sat next to each other in front of GUI consoles, each performing heroics in battle with intransigent code. Had comparable encounters been rendered in a physical medium, the results might metaphorically have been worthy of a Hollywood action film. But channeled through the medium of keyboard, screen, and mouse, when seen from a distance of two meters, the students' activities were nearly indistinguishable from peers who were playing Solitaire or surfing the web.

### 7.7.2 Aesthetics

The above paragraphs also speak to another subtext of tangible interfaces: that of aesthetics. When I applied to study at the Media Lab, my objective was to design 3D graphical representations of distributed information spaces, as inspired by metaphors of urban spaces and the writings of Lynch [1960]. I had a strong background in software design and 3D graphics programming, and felt ready to tackle this problem domain.

But when I was finally invited by Ron MacNeil to work in the Visible Language Workshop, it was a conditional admittance, conditional on addressing a perceived gap in my experience. I asked which books I should read, and MacNeil said this wasn't sufficient. And so for several months I attended night school in the hills of San Ramon, California – in graphic design. For while software and graphics provided a partial means, the ends of my work had to be to *communicate*. For work that aspired to communicate in a graphical medium, experience in traditional graphic design was perhaps the clearest point of departure.

After joining with Ishii in the Tangible Media group, my efforts turned to the physical medium. I had minimal experience with physical design and fabrication, and relied for several years upon the skills of a collaborator, Dylan Glas. When Glas departed, I decided this "secondhand" approach was untenable, and I sought to develop physical design skills of my own. I initially held strong enthusiasm for 3D printing, but quickly grew disillusioned with the limitations of our laboratory's $100,000 machine – excruciatingly slow production of small objects in coarsely-rastered blue plastic.

In contrast, I found the laser cutter to be an incredibly liberating tool. Where our 3D printer was limited to blue plastic, the laser cutter worked equally well across a wide array of compelling materials – acrylic, wood, paper, cardboard, rubber, eggplant, seaweed, and many others. Where my untrained, unpracticed hands were weak in manually drawing and crafting objects, the laser cutter gave many of the powers of software – repeatable, refinable, rapidly iterable control – but applied to physical mediums. In a sense, the laser cutter brought powers over physical materials analogous to the laser printer's transformation of paper.

And so, with the laser cutter I became a prolific builder of things. After first erring toward making all manners of objects in 1/8" clear acrylic, I broadened my vocabulary of materials, cutting dozens of artifacts out of wood, cardboard, foamcore, vellum, and many others. When it came time to prepare for my general exam with Maeda, I launched into his first exercises with vigor, rapidly producing a stream of 3D CAD sketches and laser-cut prototypes. With the third exercise, the rules changed. For renderings, Maeda restricted me to pencil, paper, and eraser; for prototypes, to knife, sandpaper, and blue foam. Use of the computer was not allowed.

My hands suddenly were as they had been a year before: untrained, unpracticed, unskilled. The drawings and objects I produced were irregular, imperfect, and poorly executed. I continued intensively in this way for a month. My handmade objects from this time never grew into works of art. After laboring for hour upon hour of cutting, sanding, coating, and resanding, the objects still lacked energy, their curves still flat to the eye and rough to the touch.

But despite, and perhaps partly because of, their coarseness, these new objects became lenses to see what I could not see before. CAD and the laser cutter had become my hammers, and my objects heavily reflected the forms these hammers made easy to create. I had grown adept at making layered stacks and pegged puzzles – often functional, sometimes clever, but rarely if ever moving, rarely if ever driven by a vision that transcended the tools.

I have not given up CAD and the laser cutter, and still count them among the most powerful tools I have encountered. But I have also found a new appreciation for objects made by hand. I see the imperfections and eccentricities which creep into handmade objects, when transformed by skilled eyes and hands, as the seeds that make these objects unique, that give them identity and soul. I increasingly admire objects that are artfully fragile, made of delicate materials, that demand and inspire attention, time, and care.

After hearing Ishii say "it's got to be beautiful" for seven years, I realize that he is speaking not only of visual aesthetics, but also of conceptual aesthetics. Despite its encyclopedic origins, I continue to find inspiration in the characterization that architecture "embraces both aesthetic and utilitarian ends that may be distinguished but not separated" [Britannica 2002].
I think this assessment also applies to the design of tangible interfaces, and hope that the work of this thesis is complementary to this architectural ideal.

# chapter 8

# conclusion

# 8 Conclusion

> *In biology, a clone is the opposite of a clade. A clade is a group of populations sharing a common origin but exhibiting genetic diversity so wide that they are barred from interbreeding. A clone is a single population in which all individuals are genetically identical. Clades are the stuff of which great leaps forward in evolution are made. Clones are evolutionary dead ends, slow to adapt and slow to evolve....*
>
> *All this, too, has its analog in the domain of linguistics. A linguistic clone is a monoglot culture, a population with a single language sheltered from alien words and alien thoughts. Its linguistic inheritance, propagated asexually from generation to generation, tends to become gradually impoverished.... Linguistic rejuvenation requires the analog of sexual reproduction, the mixture of languages and cross-fertilization of vocabularies.... In human culture as in biology, a clone is a dead end, a clade is a promise of immortality.*
>
> *Are we to be a clade or a clone? This is perhaps the central problem in humanity's future. In other words, how are we to make our social institutions flexible enough to preserve our precious biological and cultural diversity?* —*Freeman Dyson, "Disturbing the Universe" [Dyson 1979, p. 223]*

Over the last decade there has been a growing consensus that networked, computationally mediated devices will increasingly spread beyond the desktop into niche roles in the physical environment. This observation was perhaps first articulated by Mark Weiser of Xerox PARC under the label of ubiquitous computing [Weiser 1991], and in recent years has been popularly described in terms of "information appliances," "things that think," and "pervasive computing," among others. At the beginning of the new millennium, this movement is well underway.

However, these discussions usually leave unexamined one of the most central issues from the human perspective. While decentralized computing and pervasive networking are key enabling technologies, they say very little about human interaction with end-user devices. In both research settings and commercial practice, the descendants of ubicomp have remained closely tied to the graphical representations and general-purpose devices of the GUI/WIMPs approach.

This trajectory begs the question: with the continuing migration of computers into all manners of things, will all manners of things begin to look and feel like computers? This resonates strongly with this chapter's leading question from Dyson: "are we to be a clade or a clone?"

**Figure 8.1: The Xerox Star, ca. 1980; telephone, watch, and refrigerator GUIs, ca. 2002**

*(images © Xerox, Samsung, Seiko, LG Electronics)*

This thesis has worked to identify and articulate a thread of research that suggests a different paradigm for human engagement with computational systems. I have introduced the idea of "tangible user interfaces," an approach for using systems of spatially reconfigurable physical objects as representations and controls for digital information. In particular, the thesis has focused on approaches for realizing interactive systems composed of physical tokens and constraints. In these interfaces, physical tokens represent digital information elements and aggregates such as data structures and parameterized queries. Physical constraints are then used to map structured compositions of these objects onto a variety of computational interpretations.

Where previous tangible interfaces have made one-to-one mappings between physical objects and digital elements, this thesis uses physical tokens to represent *aggregates* of digital information. Information aggregates have neither an inherent physical representation, nor any intrinsic physical language for their manipulation. Interpretive constraints provide an approach for realizing this mapping. The physical representation of information aggregates significantly increases the scalability of tangible interfaces, allowing a small number of physical elements to manipulate moderate to large collections of digital information. This design choice also potentially allows simple physical actions to apply powerful computational operations over large collections of information.

I have both argued and experimentally supported the thesis claim that physically constrained tokens and interpretive constraints provide an effective approach for interacting with aggregates of information. The argument has been on the basis of the two main thesis projects, mediaBlocks and tangible query interfaces. These systems have applied tokens and interpretive constraints to support physical interaction with online media and database queries. I have also evaluated the parameter wheels query interface in a user study, where they have compared favorably with a best-practice graphical interface alternative.

## 8.1 Contributions

In supporting the thesis statement, the dissertation has made a number of specific contributions. These include:

1) Identification and characterization of tangible user interfaces as a distinct and cohesive stream of research.
2) Identification and demonstration of the token+constraint approach, providing physically embodied syntax for the structured composition of physical/digital elements.
3) Proposal and realization of techniques for physically representing and manipulating aggregates of digital information.
4) The mediaBlocks system, developing new techniques for the physical embodiment and physically situated manipulation of online information.

5) Tangible query interfaces, developing new techniques for the physical representation and manipulation of queries across large collections of digital information.

6) Experimental comparison of tangible query interfaces with best-practice graphical interface techniques, empirically supporting the value of tangible interfaces utilizing physically constrained tokens for querying tasks.

## 8.2 Limitations, challenges, and open issues

I believe that tangible interfaces in general and interpretive constraint approaches in particular provide a promising, effective means for people to interact with computational systems. At the same time, people interact with computing systems in an enormous and rapidly growing number and range of contexts. While this has provided a major motivation for the thesis, it also begs consideration of the limitations, challenges, and open issues faced by the thesis approach. Many of these issues have been considered within the introduction and discussion chapters, among other places; this section will extend and summarize these earlier discussions.

### *8.2.1 Balance between tangible and intangible interactions*

Likely the single largest question surrounding tangible interfaces is "what should be physical, and what should be digital." This question was considered at some length in the discussion of §8.4. This text noted two implicit sub-questions:

1) Which interactions are best suited to tangible user interfaces; and which are better served by graphical user interfaces or other approaches?

2) Within tangible interfaces, which elements should be represented in physical (tangible) form, and which are better represented intangibly (e.g., with dynamic graphics).

#### *8.2.1.1 Tradeoffs between tangible and graphical interfaces*

The suitability of interaction techniques – whether tangible, graphical, character-based, speech, or otherwise – heavily depends upon for whom, in what contexts, and toward what ends the interface is intended to support. Moreover, individual people's skills, preferences, and work practices also play strong roles in determining the most suitable technique.

Broadly speaking, I believe that tangible interfaces are likely to hold strongest value within a range of physical, educational, social, and personal contexts that lie outside of the stereotypical single-user, office-based, productivity-oriented task orientation – although our user study implicitly falls within this latter context. Support for colocated collaboration is a key strength of TUIs, though single-display groupware may sometimes offer a strong alternative. Interactions where the eyes and ears are busy, and contexts where the added control of two-handed manipulation and kinesthetic engagement are valued, can also lend themselves well to TUI techniques. Single user or group activities where attention is to be focused on a few key objects of interest can be good matches for tangible interfaces. People with varying kinds of disabilities are another special user community that may benefit from tangible interfaces.

Conversely, graphical user interfaces are unmatched in sheer malleability, and are very likely to remain the interface style of choice for many kinds of interactions with digital information. Interactions where diverse functionality must frequently be accessed – especially in mobile environments – are strong natural candidates for graphical interfaces. Similarly, interactions where there is a large or rapidly changing "vocabulary" of interaction – whether through textual terms, large numbers of parameters, etc. – also seem better oriented to textual or graphical techniques. Interactions with digital text broadly seems to favor graphical interfaces over tangible alternatives – though paper remains a highly effective medium, and digital interactions with paper-based text have been the basis for important TUI-related interfaces such as Wellner's DigitalDesk [1993] and Stifelman et al.'s audio notebook [1996, 2001].

#### *8.2.1.2 Tradeoffs between tangible and graphical representations*

Tangible interfaces generally rely upon a careful balance and integration between tangible and intangible (frequently graphical) representations. Among many other properties, physical persistence is one of the most fundamental aspects of TUI tangibles. This property has many implications, including for the number and kinds of elements that can profitably be embodied within a given volume of space.

As one kind of heuristic, §8.4 suggested physically embodying the "key objects of interest" – the central tools and data abstractions that persist through time and space within an interaction and elicit active manipulation. If multi-person interactions are to be supported, the key objects of interest may often be those of mutual concern, with manipulation serving to mediate interaction both with the computer and with other people. Conversely, the representation of highly dynamic or rapidly changing content often lends itself to visualization using dynamic graphics. Here, physical objects may often serve as containers, parameters, or control points, with visual dynamics born out by intangible graphics.

One of the largest outstanding issues for tangible interfaces involving non-geometric information lies in the integration and coincidence of "input" and "output" spaces. One of the major goals of tangible interfaces is to blur or even eliminate the distinction between "input" and "output," focusing emphasis on the integration between tangible and intangible representations. This integration is admirably achieved by works such as Urp [Underkoffler et al. 1999] and Illuminating Clay [Piper et al. 2002], which build upon inherently geometric domains.

While parameter wheels realize a kind of integration of physical representation and projective display, the mediaBlocks and query interfaces rely heavily on graphical display surfaces for representing the results of tangible interactions. In addition to falling short of the tangible interface ideal, this also can lead to mixed user expectations, such as whether touch screen-style interaction is also supported. While this kind of hybrid approach is possible, work such as table compositions of Iwai [1999] illustrates the potential for tighter integration. Another path relates to tighter integration with more physically representational approaches such as the layered structures of [Ullmer et al. 2001]. While some of this early work was more "output-oriented," I am optimistic about prospects for integrating these approaches with descendants of the more "input-oriented" systems from this thesis.

### 8.2.2 Physical clutter

One potential concern for tangible interfaces is the introduction of more physical artifacts into a world of cluttered desks and littered floors, already overbrimming with objects. Speaking of this, Cohen et al. write:

> Even when a TUI seems called for, some people argue that the work is doable "virtually," all within a GUI. They will say "I don't want any more stuff in my life" or "What happens if I lose the pieces?" Though we respond by pointing out their own successful real-world practice with "stuff," we suspect their arguments are not really intended to be answered. Perhaps they are simply reluctant to change interfaces, like those who balked at the introduction of the desktop metaphor. [Cohen et al. 1999]

Writing in the earliest days of the desktop metaphor, Malone studied people's work practices with physical desks, clutter and all. The first of his two major conclusions was that "a very important function of desk organization is to *remind* the user of things to do, not just to help the user *find* desired information." With the GUI "desktop," organization of information (or lack thereof) can be swept away with a single command, or minimized indefinitely. For better and for worse, these properties are not shared with the physical world.

Simultaneously, the scarcity of physical space, and the persistence of physical objects sited within it, potentially heightens their reminding function. As Cohen observes in the earlier quote, people also have extensive lived experience with the management of physical things. With tangible interfaces, these same kinds of physical strategies can in turn be shared and applied for interactions with digital information. Moreover, since the physical artifacts of tangible interfaces are usually externalizations of information that remains constantly online, a lost object does not necessarily mean the loss of the associated information or functionality.

Another key property of physical spaces is the differentiated use characteristic of the various rooms and working surfaces of buildings. For example, distinct ecologies of objects exist within the kitchen, bathroom, bedroom, office, living room, garage, and other rooms of domestic spaces. While there is interplay between these object ecologies, this is often rather limited, with eating utensils rarely wandering into bathrooms, and so forth. While surely not eliminating clutter outright, this observation has significant implications for TUIs and plays a role in compartmentalizing and bounding the tides of detritus.

As per Larkin and Simon's observations about diagrammatic representation [1987], just as a picture is not always worth 10,000 words, a tangible object is not always preferable over purely intangible words or pictures. And as related in §2.2.2 by Larkin, Simon, and Petre for the case of diagrams, the potential success of tangible interfaces is not an inherent product of tangibility, but largely a consequence of good design. While this thesis has taken early steps toward exploring this design space, articulation of a set of design principles remains to future work.

### 8.2.3 Generalizability and scalability

An implicit claim of the thesis involves the generalizability of interpretive constraints as a technique for interacting with aggregates of digital information. I believe the thesis has illustrated how some of its specific interface devices and techniques (e.g., mediaBlocks and parameter wheels) can be applied effectively to a variety of application domains. Equally important, the thesis has argued how physically constrained tokens and interpretive constraints can be used as the basis for new techniques for interacting with both information aggregates and other kinds of information.

At the same time, there are presently several kinds of limits to this generalizability. Some of these limitations relate to scalability. For example, when applied to the selection of individual data elements, the mediaBlocks sequencers' "index rack" approach seems best suited for relatively small aggregates of information. Similarly, when applied to discrete values, the parameter wheels seem best limited to small numbers of values; although the pie-menu TUI techniques of [Patten 2002] illustrate one path for further scalability, including the ability to navigate data hierarchies.

The mouse and keyboard of graphical and character-based interfaces open the door to rudimentary gesture and language based interactions, which are very powerful mediums that are unlikely to be displaced by tangible interaction techniques. Nonetheless, I believe that tangible interface techniques both can stand on their own, and also can effectively complement and interoperate with gesture- and language-based approaches. For example, the mediaBlocks "monitor slot" illustrates one mechanism by which information can be rapidly migrated between tangible and graphical interfaces. In practice, hybrid interfaces combining TUI elements with graphical and language-based approaches are likely to form an important part of tangible interfaces' future.

## 8.3 Future work

### 8.3.1 De-integration, interoperability, and ecologies of objects

One hallmark of digital technology has been a progression toward products with ever more highly integrated functionality. Personal computers are the most evident example, with some estimates of 70,000 distinct applications existing for the Microsoft Windows platform. This trend has also accelerated in the realm of personal electronics, with PDA+cell phones, camera+MP3 players, and many other variations growing increasingly common. A frequent motivating complaint is an aversion to carrying many single-purpose devices. However, such devices are sometimes criticized as metaphorical "Swiss army knives" – known for their generality, but frequently inferior to specialized tools for specific tasks.

In contrast, tangible interfaces metaphorically and literally embrace de-integration. Instead of a single device with 70,000 applications, tangible interfaces divide the functionality of individual tasks into systems of discrete, interoperating physical elements. While strengths of this approach have been discussed at length, an obvious possible drawback relates to the number of physical artifacts necessary to address a broad range of tasks.

Instead of TUIs existing as isolated islands of functionality, I believe that a key area for future research relates to interoperability between ecologies of objects. Bishop illustrated such prospects in the early 1990s with his Marble Answering Machine, as one of a series of explorations into families of interoperating physical products [Abrams 1999]. Bishop has also suggested the kitchen as a compelling metaphorical locale. Here, ecologies of scores of physical objects, ranging from forks, spoons, plates, and bowls, to a multitude of more specialized artifacts and edible mediums, provide an inspirational example of "interoperability."

While a metaphorical target that is ambitious in scope, Bishop's explorations, the mediaBlocks, and DataTiles have all taken steps toward rudimentary "object languages" that span families of interoperating products. Powerful systems and artifacts that could form seeds for such languages (including mediaBlocks and parameter wheels) are beginning to emerge. Case histories from the emergence of GUI windowing systems (such as the Xerox Star, with its somewhat monolithic shortcomings [Johnson 1989]) also provide useful lessons. Standards issues will no doubt exist. Nonetheless, key building blocks are already in motion, including the increasing pervasiveness of networking and networked devices; progress and pressures toward RFID standards [Brock 2001]; and progress in related protocols such as Jini [Arnold et al. 1999] and Hive [Minar et al. 1998].

### *8.3.2 Specialized and evolving physical forms*

Another promising direction lies in creating specialized, highly representational interfaces that give physical form to specific processes, data sets, and so forth. Such approaches can build upon the kinds of fabrication technologies used to build the tangible query interfaces, as illustrated by the exploratory beginnings of Strata. As discussed in §7.1.2 and §7.5.2, these technologies lend themselves not only to the rapid creation of physical prototypes, but also to final-form tangible interfaces that may be individually shaped with all the malleability of digital software.

I believe that the fusion of these specialized physical forms with integrated sensing and display capabilities opens the prospect for compelling new kinds of interactive artifacts and spaces. Simultaneously, these prospects are attended by new issues of their own. For instance, processes and datasets change, and the physical forms must evolve to match pace.

In part, this again speaks to the question of what should be physical, and what digital. Evolutions in visual form can be graphically mediated upon static physical forms – whether upon blank surfaces, passively printed surfaces (e.g., the "partial printing" of DataTiles [Rekimoto et al. 2001]), or upon other display surfaces (e.g., the use of projection onto the LCD display of parameter bars). The use of physical materials with widely varying levels of permanence, from paper and cardboard to steel and marble, add another important design dimension.

Physical structures themselves can also evolve across varying timescales, whether mediated manually or via internal actuation, external robotics, or (in the longer term) directed growth. Examples of early macroscale self-modifying physical structures include Negroponte's 1969 gerbil-based "Seek" [Goodman 1987], Price's building-scale "Generator" [1976], and self-modifying tangible interfaces by Frazer [1982] and Aish [1984]. These early efforts, combined with more recent progress in actuation such as [Pangaro et al. 2002, Reznick and Canny 2001, MacLean et al. 2000, and Yim et al. 2000], raise promising prospects for future TUI design.

### *8.3.3 Broader prospects*

In 1997, Ishii and I wrote that the locales in which we engage with computation are "shifting from the desktop in two major directions: onto our skin/bodies, and into the physical environments we inhabit" [Ishii and Ullmer 1997]. I believe that this observation still holds true, and that tangible interfaces have major implications for both of these directions. Where many wearable computing efforts draw people "out of their bodies" and into separate digital spaces, I believe tangible interfaces worn upon the body have special potential for styles of display and interaction that engage people more closely with their physical and social surrounds.

A great many wearable artifacts offer inspiring points of departure for TUI design. These include jewelry such as rings, necklaces, bracelets, earrings, brooches, and pins; and also clothing elements and accessories such as belts, buttons, pockets, scarves, wallets, and gloves. Ceremonial and culturally specific garments are also promising starting points. Equally important, altogether new elements of physical/digital clothing and accessories will likely develop. Uses related to communication and monitoring seem especially compelling, but many more likely exist. A number of early works have begun to explore such prospects, including [Orth 2001, Chan et al. 2001, Tollmar et al. 2000, Post et al. 2000, Wisneski 1999, Ljungstrand et al. 1999b, Borovoy et al. 1998, and Strickland 1998], and I believe there is much future promise.

Architectural spaces also offer fertile ground for the design of tangible interfaces. Ishii et al. [1994] offered an inspiring vision of "new architectural spaces where all the surfaces including walls, ceilings, windows, doors, and desktops become active surfaces through which people can interact with other spaces, both real and virtual." This vision has been inspiring to many interactive surface approaches. More recently, Ishii has spoken of the fertile boundaries between land and sea, taken as a metaphor for the boundaries between physical and digital space. In keeping with this metaphor, it is the structures that partition architectural space and the boundary regions that they create which hold special interest within this thesis.

For example, Ishii's musicBottles draw their magic from a special pedestal, suggesting a range of other instrumented furniture and cabinets where Bottles and their descendants might be stored and used. As another example, early experiments with "ambient fixtures" (e.g., the Water Lamp [Dahley et al. 1998]) hint at prospects for additional varieties of physical/digital fixtures. Simultaneously, ambient fixtures also suggest the need for companion tangible interfaces that express information sources, filters, and controls, and raise questions about where these interfaces might be physically situated. As one possibility, Underkoffler once suggested wainscotting (a kind of waist-level molding) as a prospective linear control element within future "luminous rooms" (e.g., for representing timelines like daily schedules).

These examples suggest many other elements of architectural space that might be transformed into tangible interfaces for digital information. Hand rails, posts, columns, window frames, and doorframes can be transformed into linear display and control surfaces, or as linear or radial interpretive constraints. Room partitions (permanent half-walls) could also offer linear interactive surfaces, as well as partitioning both digital and physical spaces. Shelves and shadow boxes offer three-dimensional locales for organizing and mediating TUI tokens. Drapes and tapestries offer soft surfaces for digitally modulating light and sound, and perhaps for hanging and organizing tokens used for communication and control purposes.

Culturally specific architectural elements and locales can also serve as compelling points of departure for tangible interface design. For instance, in Japanese buildings, the "genkan" is a kind of entryway and transition space just inside of exterior doorways, where shoes are removed and the "dirt" of the outside (literally and figuratively) is left behind [Deai 2002]. This suggests prospects for analogous physical/digital "genkan." As another example without West equivalence, the "ofuro" (Japanese bath) is a kind of bath in which cleaning is forbidden, which instead holds an important role as a place of meditation and communication [JFN 1996]. Ishii has suggested digital mediation of the ofuro as a kind of ambient display [Negroponte et al. 1997]. In keeping with Dyson's leading question at the chapter's beginning, these examples suggest prospects for building upon and "bringing to life" the distinctive physical artifacts, locales, and traditions of diverse cultures as central elements of future interface design.

## 8.4 Closing remarks

This thesis has introduced tangible user interfaces as a topic broadly concerned with giving physical form to digital information. Giving physical form to the intangible is among man's oldest technologies; with their 10,000 year history, the clay tokens discussed in §2.1.3 predate writing, cities, and even the wheel. As with the creation of these ancient tokens, this thesis has been concerned with *giving* form; not with repurposing or augmenting pre-existing "everyday objects," but with designing new physical forms that combine to express new kinds of physical/digital relationships.

The resulting vocabularies of physical elements are both new and very old, as perceived from slightly different perspectives. The vocabularies developed by the thesis newly open some of the most fundamental and powerful abstractions of the computational world – aggregates of digital information – to physical embodiment and manipulation. Simultaneously, the underlying elements of these vocabularies – blocks and disks, racks and pedestals – were known and used by our ancestors five and ten thousand years ago.

I believe that the value in these vocabularies is not simply that they draw from our past, better employ human skills, or offer improved task performance. More fundamentally, I believe that tangible interfaces help reconcile the human experience of both "being digital" and "being physical." As Ishii and I wrote in "Tangible Bits,"

> We live between two realms: our physical environment and cyberspace. Despite our dual citizenship, the absence of seamless couplings between these parallel existences leaves a great divide between the worlds of bits and atoms. At the present, we are torn between these parallel but disjoint spaces. [Ishii and Ullmer 1997]

The benefits of physicality are not realized simply by making bits tangible. I believe that much of the value of tangibility is a product of good design, and find this opinion echoed even by the psychologists of §2.2.2. Good design is more an art than a science; and designing tangible interfaces is likely to remain more difficult to do, and do well, than the design of GUIs. As Underkoffler writes, "the future of reactive, real-world graphics will surely have its own Rands and Tuftes, Leacocks and Gilliams" [Underkoffler et al. 1999b]. I find this analogy – and even more so, the prospects it raises – highly compelling.

My hope is that tangible interfaces can play a role in reshaping how people engage both personally and professionally with digital information. As Redström writes, "the prospect of ubiquitous computing in everyday life urges us to raise basic design issues pertaining to how we will *live* with, and not only use, computers" [2001]. I believe that Bishop's marble answering machine, Ishii's musicBottles, and my own mediaBlocks and musicbox each speak to ways that tangible interfaces can positively reshape people's personal engagement with computation.

Moreover, in a time when advances in biology and other disciplines stand to shape and even alter the very meaning of what it is to be human, mediums for exploring and discussing the implications of these advances holds unprecedented importance. Speaking of one of the earliest tangible interfaces, Aish and Noakes wrote that such interfaces "can be expected to [support] a greater *understanding* by both professional and laypeople of… complex underlying relationships" [1984]. I hope that systems such as tangible query interfaces and Strata can play a positive role in supporting these critical dialogues and decisions.

# appendix a

# implementational approaches and considerations

# Implementational approaches and considerations

> *Industrial production used to be associated with high tooling costs and very large production runs. This is now changing because the computer has paved the way for what I have called 'the electronic craftsman.' The direct relationship between the designer at the computer console and the computer-controlled means of production potentially means not just a dramatic reduction in the production costs of the tools for mass production, and thus shorter economic runs, but a one-to-one control of production and assembly equipment. This is effectively a return to one-off craft technology, but with all the capability of the precision machine tool.... The results are closer to nineteenth-century handicraft than the regimented superblocks of 1965.*
>
> *— John Frazer, "The Electronic Craftsman," in "An Evolutionary Architecture" [Frazer 1995]*

The implementation of tangible interfaces differs from that of traditional graphical interfaces in several major respects. First, tangible interfaces are intimately coupled to their embodying physical elements. This dependence places a high importance on the design and implementation of mechanically and aesthetically plausible physical artifacts. Secondly, these artifacts often must be electronically instrumented to link them to the digital world. Even within the software realm, TUI implementation often differs substantially from that of traditional GUIs.

While these observations are generally true for tangible interfaces, they take on especially strong importance for the token + constraint approaches developed within this thesis. To support and explore this assertion, it is helpful to revisit the tangible interface styles introduced in Figures 1.1 and 1.2. For convenience, these are reproduced below as Figure A.1.

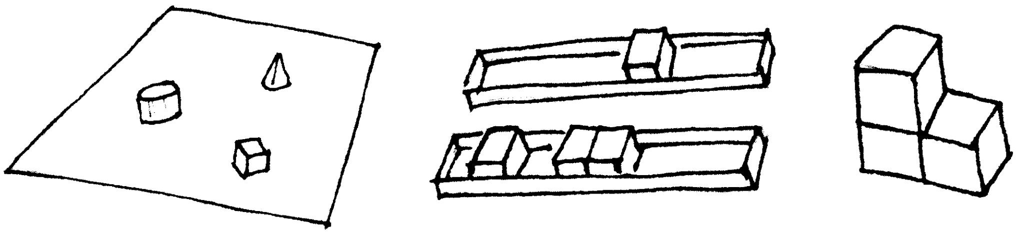


**Figure A.1a,b,c: Illustration of interactive surface, token+constraint, and constructive assembly approaches**

Interfaces employing "interactive surfaces" lend themselves to realization with fairly general purpose hardware implementations – e.g., as a "sensetable" such as described in [Patten et al. 2001a]. To the extent that such a surface is present and the designer is satisfied with its associated functional constraints, interactive surfaces can render the process of TUI implementation relatively similar to that of GUIs. Similarly for constructive approaches, their proponents often advocate the use of modular, "LEGO™-like" kits. While the design of such kits "from scratch" is not a simple proposition, interface design using pre-existing kits is potentially "a small matter of software."

In contrast, one of the driving concepts of the token+constraint approach is the use of constraints' physical/mechanical structure to express and constrain interface syntax. As has been discussed, this implies that the interface's physical structure (and often supporting electronics) is individually shaped reflect to reflect the syntax of different applications. While this does not mean that each aspect of these interfaces must be "built from scratch," it does suggest a heightened role for mechanical and electronic design and fabrication than in other tangible interface approaches.

This chapter discusses implementational approaches and considerations underlying the construction of tangible interfaces, with an emphasis on the implementation of token+constraint systems. These include issues of sensing; display and actuation; interface electronics; physical structure; and software. This chapter does not attempt to be exhaustive. Instead, the intent is to broadly describe the state of the art in the implementation of tangible interfaces, with the hope that these might serve as a starting resource for people wishing to design tangible interfaces. Of course, references to specific technologies grow quickly outdated. An attempt will be made to balance between higher-level approaches and specific grounding examples, but it is taken for granted that most specific instances will appear quaint beyond a few years remove.

## A.1 Sensing

Likely the most fundamental technical requirement for tangible interfaces relates to mechanisms for sensing physical objects. Tangible interfaces generally require the sensing of three kinds of information:

1) Object presence
2) Object identity
3) Object position/configuration

The first task is to determine *object presence*: whether a physical object that can potentially be sensed by the system is within the interface's workspace; and if so, how many. This corresponds loosely to the "entrance" and "exit" events of graphical interfaces. The second task is to determine *object identity*. This often involves attempting to map each physical artifact to a unique ID, which can in turn be resolved to some class and instance of digital information or computational functionality. Finally, tangible interfaces frequently must determine the position and configuration of their composing elements.

The following discussion is broadly divided into consideration of *continuous* sensing approaches, which provide positional information on one or more objects over a spatially continuous sensing field; and "discrete" approaches that detect the presence/absence and (usually) identity of objects within a discrete cell. Some emphasis is placed on the latter class of approaches, as they are the methods used within the thesis projects. The section makes no attempt to be exhaustive; a broad overview of sensing technologies is provided by [Fraden 1997].

It is worth noting that this section's discussion of discrete and continuous *sensing* differs somewhat from the more common consideration of discrete and continuous *sensors*. Individual sensors are often categorized as "continuous" or "discrete" on the basis of the response they are capable of yielding. For example, a simple mechanical switch is (in idealized form) "discrete" in state – either "open" or "closed," "on" or "off." In contrast, the signal from a potentiometer (when used as a voltage divisor) can take on a continuum of values/levels.

This section considers the "discrete" and "continuous" terms from a higher level of abstraction. Instead of referring to the function of individual sensor elements, this section refers to the higher-level sensing systems used to resolve the presence, identity, and spatial configuration of the system's component elements. In particular, to the extent a sensing system can continuously localize the spatial configuration of its component objects, I will consider it a continuous sensing system. Conversely, sensing systems that can detect only the presence or absence of objects within a given sensing cell will be considered as discrete approaches.

In fact, something of a continuum exists between continuous and discrete approaches, with arrays of discrete sensing elements frequently used to approximate a continuous field.
For instance, computer vision systems generally rely upon pixel-based cameras, where thousands or millions of discrete sensing elements approximate a continuous field of view. Nonetheless, from the perspective of designing tangible interface, the classification of sensing systems into continuous and discrete approaches is a useful one.

### *A.1.1 Continuous sensing approaches*

Continuous sensing approaches provide information on one or more objects located within a spatially continuous sensing field. Sensing may take place continuously along a single spatial dimension (whether translational or rotational), or a combination of multiple spatial dimensions. Tangible interfaces utilizing continuous sensing approaches often sense objects with three spatial degrees of freedom – two translational, one rotational. This supports the sensing of objects stably placed upon an interactive workbench or attached to an interactive wall.

#### *A.1.1.1 Computer vision*

One of the oldest and most popular approaches for continuous sensing is the use of computer vision. Part of computer vision's appeal draws from the prospect of sensing the position and orientation of multiple physical artifact through the use of a single inexpensive camera.

Computer vision techniques date back to the work of Roberts [1965] on "blocks worlds." Likely the first use in the context of tangible interfaces was Wellner's pioneering DigitalDesk [1993], discussed in Chapter 2. Since that time, a number of tangible interfaces have employed computer vision, including Video Mosaic [Mackay and Pagani 1994]; the metaDESK [Ullmer and Ishii 1997]; Build-It [Fjeld et al. 1998]; Illumating Light and Urp [Underkoffler et al. 1999]; Augmented Surfaces [Rekimoto and Saitoh 1999]; active phicons [Moore et al. 1999]; Collaborage [Moran et al. 1999]; Round Table [Broll et al. 2000]; and the Designers' Outpost [Klemmer et al. 2001].

Computer vision systems for TUIs have used both visible and infrared light approaches. In some cases, the target objects have been tagged; in other cases, untagged. Some of the tagged approaches have used active electronics (e.g., [Moore et al. 1999]), while others have used passive fiducials (e.g., [Underkoffler et al. 1999; Kato et al. 2000]).

While the promise of computer vision is compelling, in practice the technique poses many difficult challenges. Changes in lighting – even stemming from the movement of users in proximity to the interface – are often a major challenge for computer vision algorithms. Occlusions stemming from user interactions and inter-object interactions pose another frequent challenge. Moreover, it can be difficult to distinguish between "in-band" artifacts that should be interpreted by the system, and "out-of-band" elements that should be ignored (e.g., hands and miscellaneous objects). Some of these conditions can be partially controlled by careful camera placement and other techniques. E.g., the metaDESK [Ullmer and Ishii 1997] and Designers' Outpost [Klemmer et al. 2001] systems used infrared cameras and lamps located behind a rear-projected display surface to minimize occlusions and lighting variation.

#### *A.1.1.2 Free-space electromagnetic tracking*

The use of electromagnetic six-degree-of-freedom (6DOF) position trackers was also popular in early tangible interfaces. Devices like the Polhemus FastTrack™ and Ascension Flock of Birds™ first found popularity for head and hand tracking in virtual reality systems. These technologies have been used in tangible interfaces including neurosurgical props [Hinckley et al. 1994]; Bricks [Fitzmaurice et al. 1995]; the metaDESK [Ullmer and Ishii 1997]; and Surface Drawing [Schkolne et al. 2001].

The popularity of electromagnetic 6DOF trackings has been limited by several significant disadvantages. One is that they have generally been available for interactive use only in tethered form (with the possible recent exception of motion capture systems, which typically utilize bulky transmitters). The associated cords often complicate the use of multiple discrete physical objects. Moreover, this technology has remained relatively expensive (partially reflecting low sales volumes), with entry level prices throughout the 1980's and 1990's staying on the order of US$5000.

In addition to electromagnetic approaches, free-space (6DOF) sensing technologies based upon ultrasound, infrared beacons, and inertial techniques have been developed and commercialized. Ultrasound and infrared beacon approaches have sometimes offered cost benefits, but are sensitive to occlusion. Inertial techniques offer rapid response and modest cost, but are susceptible to drift over time. Several sensing technologies have developed hybrids of multiple sensor technologies, such as hybrids of inertial and ultrasonic techniques [Foxlin et al. 1998].

#### *A.1.1.3 Surface-based electromagnetic tracking*

Where electromagnetic 6DOF tracking technologies are designed to track targets in free space, another class of technologies track objects on or near 2D surfaces. These approaches broadly draw from computer tablet technologies. Electromagnetically-sensed tablet technologies originated with the Rand tablet, first developed in the early 1960s for handwriting input and other interactive applications [Davis and Ellis 1964]. Now widely commercialized, such tablet technologies can be applied to object tracking in TUIs by embedding their sensing elements into physical objects. This approach appears to have been pioneered by researchers at Wacom [Fukuzaki 1993, described in Fitzmaurice 1996, §3.3.10]), and has been further developed in research including [Fitmaurice et al. 1995, Kurtenbach et al. 1997, Rekimoto 2000, Patten et al. 2001].

One limitation of tablet technologies, when reconfigured for tangible interface uses, was has been a limitation to tracking a small number of objects. Generally only one or two elements are able to be simultaneously tracked, reflecting the nature of traditional stylus input tasks. This situation has progressed with the arrival of tablet devices embedded with digital device ID numbers such as Wacom's Intuos™ family. As one variant, these digital IDs allowed the development of interfaces that involve many sensing elements, only one or two of which are simultaneously active. Examples include the FlipBrick [Fitzmaurice 1996] and ToolStone [Rekimoto 2000], which used objects with Wacom styli embedded in each of six faces.

As another variant, Patten et al. have developed techniques for sensing extended number of simultaneously objects [2001]. One approach involves a workspace built of multiple tiled Wacom Intuos™ tablets. Intuos sensing elements, each with distinctive digital IDs, are embedded into sensing pucks. The Intuos elements are cyclically enabled and disabled, allowing multiple pucks to be sensed, at the cost of reduced per-puck performance.

While a viable alternative for some uses, Patten, Mazalek [2001], and others have developed a more attractive sensing approach through the tagged sensing system present within the Zowie™ toy [Francetic and Shwe 2000]. The Zowie toy was a low-cost (< US$100) toy embedded with a rectangular sensing antenna, which was capable of tracking nine centimeter-scale electromagnetic resonator tags with high accuracy and update rate. Several experimental systems have been implemented, some of them tiling multiple Zowie antennas to yield an extended sensing surface. This approach shows much promise, and is complicated primarily by the exit of the Zowie company and product from the market. The underlying Spiral™ technology, originally developed by Scientific Generics, is now owned by Synaptics, an input device manufacturer.

### A.1.2 Discrete sensing approaches

In the context of tangible interfaces, discrete sensing approaches provide information on the presence and (frequently) identity of one or more objects located within proximity or contact to a discrete sensing cell. These sensing cells may be used individually, or multiple sensing cells may be arranged into various kinds of arrays.

Discrete sensing approaches for tangible interfaces can be grouped into several major categories. First, they can be divided into contact and non-contact sensing approaches, on the basis of whether their operation depends upon mechanical and (frequently) electrical contact with the target object. Secondly, the target objects can be either untagged, tagged, or electronically instrumented.

#### A.1.2.1 Contact-based sensing

In contact-based discrete sensing approaches, the TUI's target objects are detected through mechanical and (frequently) electrical contact with the target object. The target objects themselves may be either untagged, tagged, or electronically instrumented.

*Untagged objects*

Technologically, perhaps the simplest form of sensing for tangible interfaces is the contact-based sensing of untagged objects. For example, a minimalistic approach might be to attach a mechanical switch to some form of rigid or semi-rigid "sensing surface." When an object of sufficient mass (say, a wooden block) is placed upon the "sensing surface," the switch is closed. When the object is removed, the switch is opened.

This approach has several immediate limitations. First, a switch generally cannot discern the identity of the object. If multiple objects are present, this also cannot be detected. Moreover, a switch generally cannot discern whether the object is "in-band" (intended for interpretation by the system), or "out-of-band" (e.g., with the switch triggered by a user's hand).

Another simple approach is to weigh the target object's mass. This has been effectively employed in the Passage system of Streitz et al. [1999], where a sensitive electronic scale allowed the binding of information to everyday objects (e.g., a user's key ring).
While well-implemented in the Passage system, this approach has several limitations.
One is speed; e.g., the Passage system's sensitive scale required several seconds to stabilize. Calibration, robustness, lack of unique identity, size, and cost also pose additional limitations.

*Tagged objects*

A more common form of contact-based sensing is to employ some kind of tag. This tag can be as simple as a conductive wire or surface that makes electrical contact when an object is placed upon a sensing cell. However, this example does not generally support the sensing of object identity.

Another simple approach is to tag objects with a resistor of known value. This approach was used in the earliest version of the mediaBlocks, and is used commercially by the Music Blocks product [Neurosmith 1999]. This approach allows some identity information to be provided, depending upon the accuracy of the resistor valuation and the sensitivity of the reader. However, like many electronic contact-based approaches, resistor-based IDs require a stable, reliable electronic connection, which is somewhat difficult to achieve in the context of TUI interactions. To compensate for this, the Music Blocks product utilizes a carefully designed electromechanical connector, and includes electrical contacts that flex under the weight of the musicBlocks, providing pressure to form a reliable connection.

As another alternative, wired "digital serial numbers," such as the 1Wire/iButton family of products from Dallas Semiconductor, support low-cost, unique-ID tagging of physical objects. The iButton family of products were used in the mediaBlocks, LogJam [Cohen et al. 1999], and ToonTown [Singer et al. 1999] tangible interfaces. However, these objects must support stable, reliable electromechanical contact with two leads, which in practice is non-trivial to implement – especially for token/constraint systems supporting a "manipulate" phase of interaction (§1.1.2). LogJam and ToonTown used magnets to improve the electrical connection. MediaBlocks used a "fuzzy conductive Velcro" material first employed by the Triangles system [Gorbet et al. 1998]. However, each of these systems suffered from unreliable performance.

*Electronically instrumented objects*

A third contact-based approach is to embed TUI tangibles with custom electronics, and to again pass data over a wired connection. This approach is especially popular in TUIs employing constructive assemblies – e.g., [Frazer 1982; Aish 1984; Suzuki and Kato 1993; Gorbet et al. 1998; Anderson et al. 2000; McNerney 2000; Heaton 2000], among many others. However, these systems have again frequently struggled with establishing and maintaining reliable electromechanical connections. Some systems utilizing specialty connectors have achieved better performance (e.g., [McNerney 2000; Heaton 2000]). However, as such systems are scaled up to include many elements, even specialty connectors have proven problematic (e.g., see [Anderson et al. 2000]).

### *A.1.2.2 Non-contact sensing*

In non-contact discrete sensing approaches, TUI tangibles are detected through optical, electromagnetic, or other wireless interactions with the target object. To be clear, the target objects and sensing systems of non-contact sensing schemes often involve physical contact (e.g., an RFID tag placed upon a tag reader). However, these sensing schemes do not rely upon a contact-based mechanical or electrical connection, and consequently can often deliver better reliability. The target objects themselves may again be untagged, tagged, or electronically instrumented.

*Untagged objects*

Untagged, uninstrumented physical objects can be wirelessly sensed in a variety of fashions. Optical proximity sensors (often involving an infrared transmitter/receiver pair) are a common approach. Capacitive sensing provides another possible path. Object identity is again difficult to determine with such approaches. However, these approaches can be usefully integrated with other sensing technologies in sensor fusion approaches that offer increased sensing robustness and decreased power consumption (e.g., using RFID sensing circuitry that is activated only when an object is proximal).

*Tagged objects*

Perhaps the most common kind of object tag is the barcode. Barcodes have been used in tangible interfaces including the transBOARD [Ullmer 1997], Webstickers [Ljungstrand and

Holmquist 1999], and the Paper Palette [Nelson et al. 1999]. However, barcodes require line-of-site interrogation of the barcode label.

One of the most promising sensing technologies for TUIs are RFID (radiofrequency identification) tags. These are small, wireless tags that are inductively or capacitively coupled to a tag reader. Most RFID tags are encoded with unique IDs, with some tags also offering user-writeable nonvolatile memory. RFID technologies have become increasingly popular in recent years. Among many other application areas, this technology is being embraced as a digitally-interrogable replacement for barcode UPC labels, which might be pervasively embedded within a wide range of commercial products [Brock 2001].

RFID tags have been used in a variety of tangible interfaces, including [Want et al. 1999; Yarin and Ishii 1999; Rekimoto et al. 2000], and tangible query interfaces, among others. One potential limitation of RFID tags involves their integration within objects that are electronically instrumented (e.g., that include active displays). However, dual-ported RFID technologies such as [IBM 2000, Atmel 2002] show promise for integrating such active devices with standard RFID protocols and readers.

*Electronically instrumented objects*

A final non-contact sensing approach is to embed TUI tangibles with custom electronics and to track presence, identity, and location using wireless mediums. This medium can be an optical link, as with the GDP cubes of Anagnostou et al. [1989], the Tiles of Kramer [1999], and the phicons of Moore et al. [1999]. Inductive communications offer another approach, as with Kramer's beads [Resnick et al. 1998], the tangible query interfaces, and dual-ported RFIDs.

## A.2 Display and actuation

In order to provide computationally-mediated feedback to users, tangible interfaces must integrate some form of display or actuation. The most common approach is to provide dynamic graphical feedback through back projection, front projection, embedded displays, standalone screens, or bistable surfaces.

Front projected surfaces have been one of the most popular methods for the illumination of interactive workbench and interactive wall systems. Tangible interfaces utilizing this approach include the DigitalDesk [Wellner 1993], Video Mosaic [Mackay 1994], Build-It [Fjeld 1998], Urp [Underkoffler and Ishii 1999], Sensetable [Patten et al. 2001], Senseboard [Jacob et al. 2002], and the tangible query interfaces.

These systems have most often utilized ceiling-mounted projectors, which in the present day raises issues of cost and mobility. However, the projection jig used by tangible query interfaces illustrates how these constraints can be reduced. Front projection systems also face issues of occlusion by the hands and bodies of users, which can be especially problematic in interactive wall systems (e.g., [Jacob et al. 2002]). However, the trapezoidal anti-keystoning optics of most present-day projectors help ameliorate this issue, as discussed in [Underkoffler et al. 1999].

As another alternative, back projected surfaces have been used by a variety of interactive workbench and wall systems, including Bricks [Fitzmaurice et al. 1995], the metaDESK, Surface Drawing [Schkolne et al. 2001], and the Designer's Outpost [Klemmer et al. 2001]. Perhaps the main disadvantages of back-projection to date has been the large physical space necessary to accommodate the throw path of the supporting projector.

Embedded visual displays ranging from single lamps and LEDs to high-resolution embedded flat panels have also been used within TUIs. Embedded lamps and LEDs have been among the oldest methods of TUI visual display, used in systems including the Slot Machine [Perlman 1976]; the Universal Constructor [Frazer 1982, 1995]; AlgoBlocks [Suzuki and Perlman 1993]; Stackables [Kramer and Nelson 1999]; Tiles [Kramer 1999]; Peano [Heaton 2000]; and others. LED matrices have also been used in systems such as Yarin's TouchCounters [1999]. Embedded high-resolution LCD displays have been used in systems such as the metaDESK and mediaBlocks, with smaller LCD displays embedded within the query interfaces' parameter bars. Many tangible interfaces have also made use of standalone computer screens.

A promising approach for the future is the use of bistable display surfaces. Technologies such as "electronic ink" [Jacobson et al. 1997] have begun to enter commercial use, and in the medium term have the potential to substantially impact the use of tangible interfaces. Among the virtues of this technology are low power consumption and the prospects for printing onto surfaces of widely varying shape [Comiskey et al. 1998].

Another variation of this approach that has been developed by Prof. Joe Jacobson is "reusable paper." Here, the appearance of thermochromic inks can be reversably changed through exposure to a thermal printhead. For example, I experimented with surfacing the parameter wheels with prototype "reusable paper," in the hope that the surface of parameter wheels could quickly be "reprinted" without embedding any electronics beyond an RFID tag. This approach was complicated by the fact that most thermal printheads are designed for use against a cylindrical platen, but showed promise for future development.

While the above text has focused upon visual augmentations, audio has also been a popular medium within tangible interfaces. TUIs that have relied upon audio as the primary display medium include Dr. Legohead [1995]; Voice Boxes [Jeremijenko 1996]; Stifelman's audio notebook [1996, 2001]; Triangles [Gorbet et al. 1998]; Music Bottles [Ishii et al. 1999, 2001]; Audio Shapers [Weinberg 2000]; Rasa [2001]; and TellTale [Ananny 2001].

Mechanical actuation is another major class of functionality that has significant implications for tangible interfaces. TUIs that have made use of actuation include inTouch [Brave et al. 1998]; PsyBench [Brave et al. 1998]; tagged handles [MacLean et al. 2000]; haptic media controls [Snibbe et al. 2001]; behavioral kinetic sculptures [Reas 2001]; Phidgets [Greenberg and Fitchett 2001]; the Universal Planar Manipulator [Reznick and Canny 2001]; and the Actuated Workbench [Pangaro et al. 2002].

Other potential display mediums for tangible interfaces include vibration [Holmquist et al. 1998; Wisneski 1999]; warmth [Wisneski 1999; Tollmar et al. 2000]; scent [Kaye 1999]; and taste (e.g., "edibles").

## A.3 Physical structures

> *Every scientific apparatus, even a device that is fundamentally electronic or optical in nature, requires a mechanical structure. The design of this structure determines to a large extent the usefulness of the apparatus, and thus a successful scientist must acquire many of the skills of the mechanical engineer in order to proceed rapidly with an experimental investigation. [Moore 1989]*

Given the centrality of physical objects within tangible interfaces, techniques for physical fabrication are a key topic. The choice of physical tools has a great impact on the physical structures that can be fabricated by the TUI designer, and consequently on the kinds and varieties of interface possibilities that can be realized.

One approach is to embrace the use of pre-existing "everyday objects." This is the path that I favored during my master's thesis [Ullmer 1997], and appears to be a common instinct for many implementors with backgrounds in HCI. However, there are also significant conceptual and technical limitations to this approach. Conceptually, the use of "everyday objects" in TUIs is in some respects analogous to using photographs of "everyday" things as replacements for GUI icons. While this analogy is somewhat flawed, it suggests some of the problems of representing digital information and operations with "overloaded" forms from the physical world. From a technical standpoint, developing reliable sensing strategies is non-trivial; computer vision is a tricky beast, and even 6DOF sensors and RFID tags require well-planned placement.

Another popular approach is the use of modular physical construction kits, with LEGO™ as the most popular example. LEGO™ can be a useful rapid prototyping medium, especially for structures that rely upon linkages of racks, gears, and other elements. However, LEGO™ is also a rather limiting medium, and is generally inappropriate for TUI tangibles that have gone beyond the "early concept prototype" stage.

A next level of implementation involves the handcrafting of light craft materials such as balsa, cardboard, foam, and foamcore. While these materials have been used in prototyping both the mediaBlocks and tangible query interface systems, their lack of rigidity and the imprecision of hand cutting make them difficult to use in functional prototypes of token+constraint systems.

Traditional woodworking and machining offer the most traditional professional prototyping approach. However, these approaches often draw upon skills and machinery that are not widely present among HCI practitioners. Also, traditional woodworking and machining can be time-intensive, making these approaches challenging in the face of frequent design iterations. CNC machine tools provide some ability to leverage the powers of software for iterative modification, but the time-overhead of jigging, fixturing, etc. is frequently significant.

A relatively new class of machine tools such as laser cutters and water jet cutters offer powerful resources for TUI design. For example, with $CO_2$ laser cutters sold by Universal Laser and other vendors, shapes drawn in consumer drawing software like CorelDRAW™ can be cut in seconds from wood, acrylic, and many other materials placed upon the cutter's bed. Early prototypes can be rapidly iterated in light materials like corrugated cardboard and foamcore, with the flexibility of software control allowing for repeatable fabrication and easy modification. Alternately, water jet cutters are frequently larger, messier, and more expensive, and generally require more fixturing, but allow fabrication in aluminum, stone, glass, and other materials that cannot be cut with $CO_2$ laser cutters.

Finally, a wide variety of 3D printing technologies are commercially marketed, including laser-cured stereo lithography, automated stacking of successive laminates, and selective binding of powder, among others. At present, these technologies often have very slow fabrication times (build times of >20 hours for fist-sized objects are common) and a limited range of materials. Nonetheless, these technologies have been used successfully for tangible interface designs, and will likely grow increasingly attractive with continuing improvements in speed and range of materials. Especially exciting is the prospect of incorporating electronically functional materials, as suggested by early results such as inkjetted displays [Comiskey et al. 1998] and printed transistors [Ridley et al. 1999].

## A.4 Interface electronics

The physical artifacts of tangible interfaces must generally be instrumented electronically to realize their coupling with the digital world. The "input" and "output" elements of sensors, displays, and actuators have been briefly reviewed above. Having determined which of these are to be used, these elements must be linked to one or more microcontrollers or microprocessors from which the firmware and software can coordinate the tangible interfaces' high level behavior.

This kind of electronics interfacing can take place at a number of different levels. At the highest level, if (e.g.) a commercial projector or high-resolution data display are to be "interfaced," often all that is necessary is to power the device through a wall AC connection or batteries, and connect the device to (e.g.) a controlling computer through a standard video cable.

However, frequently a tighter level of integration is necessary. Several intermediate levels of integration exist. First, a toolkit of component elements may be available for use, such as the Phidgets of Greenberg and Fitchett [2001]. Second, several modular system of sensors and actuators are currently marketed, such as the iCube system from Infusion [2002]. Finally, a wide range of data acquisition and control products are marketed by companies such as National Instruments.

At a still lower level of integration, a number of "single-board computers" are suitable for embedded use. Examples that have been popular for TUI development include the iRX [Poor 1999]; the Basic Stamp [Parallax 2002]; TINI [iButton 2002]; and SaJe + JStamp [Systronix 2002].

Technologies that have been embraced by the wearable computer community, such as the PC104 line of single-board computers, also have special TUI relevance.

At a lower level still, several families of microcontrollers provide an excellent means for implementing tangible interfaces. For example, Microchip's PIC family of microcontrollers have proven popular for TUI design [Microchip 2002]. As one example, the PIC 12CXXX family of microcontrollers are available in pea-sized, easy-to-solder (DIP) form. Costing as little as US$1 in single-unit quantities, they include an analog-to-digital converter; require no external components other than a power source to function; and can be programmed with the C language (e.g., from CCS) and a <$200 programmer (e.g., Microchip's PICstart);. Larger PIC chips, such as the 16F84 and 16F876, have also frequently been used in TUI designs (including the systems of this thesis).

A great many other electronics-related issues ranging from power technologies to PC board fabrication are important for tangible interfaces, but these are beyond the scope of this appendix.

## A.5 Communications

At present, tangible interfaces frequently link one or more "traditional" computers with one or more "embedded" microcontroller devices. (This can be expected to change as computing systems continue to become available in ever smaller forms.) Correspondingly, determining how these processors will communicate is an important design decision.

Perhaps the most common approach is the use of RS232 or RS422 serial communications, generally at communication rates ranging between 9600 and 115K bits per second. The greatest strength of this approach is that it is relatively simple to implement and widely supported. However, this approach also has several limitations. First, RS232 generally requires close proximity between the two communicating devices. Second, while personal computers have traditionally included one or two RS232 serial ports, many tangible interfaces require more than this number of serial ports (e.g., the mediaBlocks and tangible query interface systems used seven and four serial ports, respectively). Third, RS232 communication speeds are limiting for some kinds of tangible interfaces (e.g., those employing haptic feedback). Fourth, increasing numbers of personal computers have no RS232 serial ports.

As a means for providing additional serial port connections, a number of multi-serial PC cards and USB- and Ethernet-based serial hubs are sold commercially. Alternately, an increasingly plausible option is for embedded controllers to directly integrate USB or Ethernet support. For example, the TINI [iButton 2002] and Saje [Systronix 2002] single-board computers both provide internal Ethernet support, and PIC processors are now available with internal USB support (e.g., see [Maclean et al. 2002]). Both USB and Ethernet have the advantage of higher communication speeds, and support multiplexing through inexpensive hubs. Moreover, Ethernet offers the powerful advantage of TCP/IP network connectivity, allowing tangible interfaces to communicate with both proximal and remote computers and networked devices.

In addition to serial, USB, and Ethernet connections, a number of other wired communications mediums such as $I^2C$, SPI, GPIB, OneWire, and custom protocols have been used within tangible interfaces. Alternately, wireless communication is an increasingly attractive prospect. Infrared communications have been used within tangible interfaces dating back to the 1980s (e.g. [Anagnostou et al. 1989]), using protocols such as Sony remote control code and IRDA.

In addition, radio communications have become an increasingly plausible alternative. Easily integrated RF communications modules have been available for years from companies like Linx Technologies and RF Monolithics. Early generation devices made it difficult for multiple copresent devices to carry on simultaneous bidirectional communications. However, newer products offer multi-channel and collision detection/resolution support. As another variation, RFID protocols offer a kind of near-field RF communications medium. Dual-ported RFID technologies such as [IBM 2000, Atmel 2002] allow standard RFID protocols to be used to communicate between proximal active devices. Finally, wireless Ethernet, mobile telephone protocols, and bidirectional pager protocols all offer additional communication prospects for future TUI use.

## A.6 Software

The topic of software implementation for tangible interfaces may also be considered at a number of different levels. One perspective relates to software for simulating tangible interfaces, sometimes with triggers from sensing systems. Macromedia's Director product has been a popular authoring environment both for expressing scripted TUI scenarios (as with Bishop's original marble answering machine concept [Polynor 1995]), as well as for making interactive pieces that react to real sensor systems via "Xtra" plugins. OpCode's "Max" program, a visual programming environment for coordinating MIDI devices such as the iCube [Infusion 2002]), is another popular choice for scripting TUI concept implementations. At a somewhat related level of description, 3D skeching and animation packages like Caligari TrueSpace™ have been used for illustrating both stills and animations of prospective TUI designs.

Traditional PC-based programming and scripting languages likely remain the most common software mediums for realizing TUI concepts. Popular examples include high-level languages like C, C++, and Java, as well as scripting languages such as Tcl and Python. These languages are frequently used in combination with graphics libraries such as OpenGL, Open Inventor, Tk, or Java2D. In addition, support for serial and network communications is another important factor.

Another level of software design relates to the programming of embedded processors. C and Assembler cross-compilers are among the oldest and most common tools for microcontroller programming. However, microcontroller- and single board computer-based implementations of BASIC [Parallax 2002], Logo [Resnick et al. 1993], and Java [iButton 2002, Systronix 2002] offer higher-level approaches for controlling embedded electronics.

Another equally important perspective relates to prospects for TUI software libraries, event models, and architectures. Sensor-based events and mappings, dynamic binding, network- and serial-linked communication, database support, etc. are all relevant aspects. Underkoffler has discussed design principles relating to the role of graphical dynamism within tangible interfaces, including "apparent life;" "disambiguation of the real and the virtual;" "increased resolution;" and "aesthetics" [Underkoffler et al. 1999]. Other concepts relating to TUI software architectures are discussed in [Ullmer 1997].

## appendix b

# user study

### Questionnaire

*Note: formatting is adjusted for inclusion in appendix.*

Please circle/fill in your answers to the following questions.

**Biographical information**

**Age:** ______

**Sex:** M F

**Are you primarily left- or right-handed?** Right Left

**Expertise with using computers:** [complete novice] 1 2 3 4 5 6 7 [seasoned expert]

## About the "graphical interface" (on the screen, with the mouse and sliders):

Expressing queries with the graphical interface was generally [difficult/easy]:

(please indicate your rating by circling a point on the scales below)

[very difficult] 1 2 3 4 5 6 7 [very easy]

I felt the graphical interface was generally ...

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| easy to learn: | [strongly disagree] | 1 | 2 | 3 | 4 | 5 | 6 | 7 | [strongly agree] |
| easy to use: | [strongly disagree] | 1 | 2 | 3 | 4 | 5 | 6 | 7 | [strongly agree] |

I felt this kind of interface would allow me to effectively query real databases:

[strongly disagree] 1 2 3 4 5 6 7 [strongly agree]

## About the "tangible interface" (projected, with the physical wheels):

Expressing queries with the tangible interface was generally [difficult/easy]:

[very difficult] 1 2 3 4 5 6 7 [very easy]

I felt the tangible interface was generally ...

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| easy to learn: | [strongly disagree] | 1 | 2 | 3 | 4 | 5 | 6 | 7 | [strongly agree] |
| easy to use: | [strongly disagree] | 1 | 2 | 3 | 4 | 5 | 6 | 7 | [strongly agree] |

I felt this kind of interface would allow me to effectively query real databases:

[strongly disagree] 1 2 3 4 5 6 7 [strongly agree]

In comparing the performance of these two interfaces, I felt that:

[graphical interface is much faster] 1 2 3 4 5 6 7 [tangible interface is much faster]

In general, I prefered using :

[strongly preferred graphical interface] 1 2 3 4 5 6 7 [strongly preferred tangible interface]

*[Subjects were provided with more space to respond to the following questions]*

In general, what were the strong points of the graphical interface (on the screen)?

In general, what were the weak points of the graphical interface?

In general, what were the strong points of the tangible interface (with the "wheels")?

In general, what were the weak points of the tangible interface?

Please give us any comments or suggestions you have on this experiment.

## appendix c

# Properties of Sumerian tokens

Quoted from [Schmandt-Besserat 1996, p. 97], which in turn draws from [Hockett 1960]:

1. *Semanticity:* Each token was meaningful and communicated information
2. *Discreteness:* The information conveyed was specific. Each token shape, like each pictograph, was bestowed a unique meaning. The incised ovoid, for example, like the sign ATU 733, stood for a unit of oil.
3. *Systematization:* Each of the token shapes was systematically repeated in order to carry the same meaning. An incised ovoid, for example, always signified the same measure of oil.
4. *Codification:* The token system consisted of a multiplicity of interrelated elements. Besides the cone, which stood for a small measure of grain, the sphere represented a larger measure of grain, the ovoid meant a jar of oil, the cylinder an animal, and so on. Consequently, the token system made it feasible, for the first time, to deal simultaneously with information concerning different items.
5. *Openness:* The repertory of tokens could be expanded at will by creating further shapes representing new concepts. The tokens could also be combined to form any possible set. This made it feasible to store an unlimited quantity of information concerning an unlimited number of items.
6. *Arbitrariness:* Many of the token forms were abstract; for example, the cylinder and lenticular disk stood respectively for one and ten *(?)* animals. Others were arbitrary representations; for instance, the head of an animal bearing a collar symbolized the dog.
7. *Discontinuity:* Tokens of closely related shapes could refer to unrelated concepts. For example, the lenticular disk stood for ten *(?)* animals, whereas the flat disk referred to a large measure of grain.
8. *Independence of phonetics:* The tokens were concept signs standing for units of goods. They were independent of spoken language and phonetics and thus could be understood by people speaking different tongues.
9. *Syntax:* The tokens were organized according to set rules. There is evidence, for example, that tokens were arranged in lines of counters of the same kind, with the largest units placed at the right.
10. *Economic content:* The tokens, like the earliest written texts, were limited to handling information concerning real goods. It is only centuries later, about 2900 B.C., that writing began to record historical events and religious texts.

# References

ABOWD, G., AND MYNATT, E. (2000). Charting Past, Present, and Future Research in Ubiquitous Computing. In *ACM Transactions on Human-Computer Interaction*, 7(1), pp. 29-58.

ABRAMS, R. (1999). Adventures in Tangible Computing: The Work of Interaction Designer, Durrell Bishop, In Context. MA thesis, Royal College of Arts, 1999.

AHLBERG, C. AND WISTRAND, E. (1995). IVEE: An Information Visualization & Exploration Environment. In *Proceedings of IEEE Visualization 95.*

AHLBERG, C. AND SHNEIDERMAN, B. (1994). Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. In *Proceedings of CHI 94*, pp. 313-317.

AISH, R., AND NOAKES, P. (1984). Architecture without numbers – CAAD based on a 3D modelling system. In *Computer-Aided Design*, v16n6, November 1984, pp. 321-328.

AISH, R. (1979). Three-dimensional Input for CAAD Systems. In *Computer-Aided Design*, v11n2, March 1979, pp. 66-70.

ANAGNOSTOU, G., DEWEY, D., AND PATERA, A. (1989). Geometry-defining processors for engineering design and analysis. In *The Visual Computer*, 5:304-315, 1989.

ANANNY, M. (2001). Telling Tales: A new toy for encouraging written literacy through oral storytelling. In *Proceedings of Society for Research in Child Development.*

ANDERSEN, P. (1997). *A theory of computer semiotics: semiotic approaches to construction and assessment of computer systems.* Cambridge: Cambridge University Press, 1997.

ANDERSON, D., FRANKEL, J., MARKS, J., AGARWALA, A., BEARDSLEY, P., HODGINS, J., LEIGH, D., RYALL, K., SULLIVAN, E., AND YEDIDIA, J. (2000). Tangible Interaction + Graphical Interpretation: A New Approach to 3D Modelling. In *Computer Graphics (Proceedings of SIGGRAPH'00.*

ANICK, P., BRENNAN, J., FLYNN, R., HANSSEN, D., ALVEY, B., AND ROBBINS, J. (1989). A Direct Manipulation Interface for Boolean Information Retrieval via Natural Language Query. In *Proceedings of Research and Development in Information Retrieval'89*, pp. 135-150.

ARAI, T., MACHII, K, AND KUZUNUKI, S. (1995). Retrieving Electronic Documents with Real-World Objects on the InteractiveDESK. In *Proceedings of UIST'95*, pp. 37-38.

ARAYA, A. (1995). Questioning Ubiquitous Computing. In *ACM Conference on Computer Science*, pp. 230-237.

ARIAS, E., EDEN, H., AND FISHER, G. (1997). Enhancing communication, facilitating shared understanding, and creating better artifacts by integrating physical and computational media for design. In *Proceedings of DIS'97*, pp. 1-12.

ARNOLD, K., WOLLRATH, A., O'SULLIVAN, B., SCHEIFLER, R., AND WALDO, J. (1999). *The Jini Specification.* Addison-Wesley, 1999.

ATMEL. (2002). AT88RF001: RFID External EEPROM Interface IC. `http://eu.atmel.com/atmel/acrobat/doc2004.pdf` [Visited July 14, 2002]

AZUMA, R. (1997). A Survey of Augmented Reality. In *Presence*, 6(4), August 1997, pp. 355-385.

BAEZA-YATES, R. (1998). *Visual Programming.* `http://www-lsi.upc.es/~rbaeza/sem/vp/vp.html` (Visited 8/19/1998.)

BAJURA, M., FUCHS, H., AND OHBUCHI, R. (1992). Merging Virtual Objects with the Real World: Seeing Ultrasound Imagery within the Patient. In *Proceedings of SIGGRAPH'92*, pp. 203-210.

BAYLEY, O. AND WHITE, S. (1997). Candle Altar: The Creation ofan Artifact to Facilitate Spirituality. Interval Technical Report IRC#1997-099. November 1997. `http://www.whitenoise.org/projects/candle`

BELL, R. (1979). *Board and Table Games from Many Civilizations.* New York: Dover Publications.

BOROVOY, R., MARTIN, F., VEMURI, S., RESNICK, M., SILVERMAN, B., AND HANCOCK, C. (1998). Meme Tags and Community Mirrors: Moving from Conferences to Collaboration. In *Proceedings of CSCW'98.*

BOROVOY, R. (1995). Genuine Object-Oriented Programming. MS Thesis, MIT Media Laboratory, 1995.

BRAVE, S., ISHII, H. AND DAHLEY, A. (1998). Tangible Interfaces for Remote Collaboration and Communication. In *Proceedings of CSCW '98*, pp.169-178.

BRITANNICA ONLINE (ENCYCLOPÆDIA). (2002a). "abacus." `http://members.eb.com/bol/topic?eu=3239&sctn=1` [Accessed 10 February 2002].

BRITANNICA ONLINE (ENCYCLOPÆDIA). (2002b). "city." `http://www.britannica.com/eb/article?eu=109103` [Accessed August 3, 2002].

BRITANNICA ONLINE (ENCYCLOPÆDIA). (2002c). "wheel." `http://www.britannica.com/eb/article?eu=78784` [Accessed August 3, 2002].

BRITANNICA ONLINE (ENCYCLOPÆDIA). (2002d). "writing" `http://www.britannica.com/eb/article?eu=119409` [Accessed August 3, 2002].

BRITANNICA ONLINE (ENCYCLOPÆDIA). (2000). "architecture." `http://www.britannica.com/eb/article?eu=119277` [Accessed July 1, 2000].

BROCK, D. (2001). The Electronic Product Code (EPC): A Naming Scheme for Physical Objects. MIT Auto ID center: `http://www.autoidcenter.org/research/MIT-AUTOID-WH-002.pdf`.

BROLL, W., MEIER, E. AND SCHARDT, T. (2000). The Virtual Round Table: a Collaborative Augmented Multi-User Environment. In *Proceedings of ACM Collaborative Virtual Environments*, pp. 39-46.

BURBECK, S. (1987). Applications Programming in Smalltalk-80: How to use Model-View-Controller. http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html

BUSH, V. (1945). As We May Think. In *Atlantic Monthly*, July 1945.

CAMERATA, K., DO., E., JOHNSON, B., AND GROSS, M. (2002). Navigational Blocks: Navigating Information Space with Tangible Media. In *Proceedings of IUI'02.*

CARD, S., MACKINLAY, J., AND SHNEIDERMAN, B. (1999). *Readings in Information Visualization: Using Vision to Think.* New York: Morgan Kaufmann, 1999.

CATARCI, T., COSTABILE, M., LEVIALDI, S., AND BATINI, C. (1997). Visual Query Systems for Databases: A Survey. In *Journal of Visual Languages and Computing*, 8(2), 1997, pp. 215-260.

CHAN, D., MINER, C., AND CAMPBELL, C. (2001). Digital Jewelry: Wearable Technology for Everyday Life. In *Extended Abstracts of CHI'01.*

CHIZLETT, C. (1999). The Token Woman. In *Proceedings of 2nd National Conference on Iconic Communication.*

COHEN, J., WITHGOTT, M., PIERNOT, P. (1999). Logjam: A Tangible Multi-Person Interface for Video Logging. In *Proceedings of CHI'99*, pp. 128-135.

COMISKEY, B., ALBERT, J., YOSHIZAWA, H., AND JACOBSON, J. (1998). An Electrophoretic Ink for All-Printed Reflective Electronic Displays. In *Nature*, 394(6690), pp. 253-255.

COOPERSTOCK, J., TANIKOSHI, K., BEIRNE, G., NARINE, T., AND BUXTON, W. (1995). Evolution of a Reactive Environment. In *Proceedings of CHI'95.*

COUTAZ, J. (1987). PAC, an Object Oriented Model for Dialog Design. In *Proceedings of Interact'87*, pp. 431-436.

CSIKSZENTMIHALYI, M., AND ROCHBERG-HALTON, E. (1981). *The meaning of things: domestic symbols and the self.* Cambridge, UK: Cambridge University Press, 1981.

DAHLEY, A., WISNESKI, C. AND ISHII, H. (1998). Water Lamp and Pinwheels: Ambient Projection of Digital Information into Architectural Space. In *Proceedings of CHI'98 Companion*, pp. 269-270.

DAVIS, M., AND ELLIS, T. (1964). The RAND Tablet: A Man-Machine Graphical Communication Device. In *AFIPS Conference Proceedings (FJCC)*, v26(1), pp.325-331.

DE VOOGT, A. (1998). Editorial. In *International Journal for the Study of Board Games.* `http://www.iias.nl/host/boardgames/editorial.html` [Accessed 23 February 2002].

DEAI (MINI ENCYCLOPEDIA). (2002). Genkan: Entrance. `http://www.tjf.or.jp/deai/contents/teacher/mini_en/html/genkan.html`

DEFANTI, T. AND BROWN, M. (1991). Visualization in Scientific Computing. In *Advances in Computers*, v.33. Academic Press, pp. 247-305.

DOURISH, P. (2001). *Where The Action Is: The Foundations of Embodied Interaction.* Cambridge, MA.: MIT Press, 2001.

DUNNE, A. AND RABY F. (1994). Fields and Thresholds. Presentation at the Doors of Perception 2, November 1994, http://www.mediamatic.nl/Doors/Doors2/DunRab/DunRab-Doors2-E.html

DUNSMOIR, B. (2002a). Course Syllabus: Introduction to Psychology. http://www.realtime.net/~dunjo/drbad/freud.html [Accessed 8 March 2002].

DUNSMOIR, B. (2002b). Personal communications. March 10, 2002.

DURHAM, J. (2002a). Abrasives, Trust, and How the Abacus Got Its Name. `http://bart.cba.nau.edu/~durham-j/newsite/id153.htm` [Accessed 10 February 2002].

DURHAM, J. (2002b). Personal communications. February 10, 2002.

EILAN, N., MCCARTHY, R. AND BREWER, B. (1993). *Spatial Representation.* Oxford, UK: Blackwell, 1993.

ELLIS, T., HEAFNER, J., AND SIBLEY, W. (1967). The "Grail" Project. In *Technical Digest, 8th Annual IEEE Symposium on Human Factors in Electronics*, p. 19.

ENGELBART, D. (1962). Augmenting Human Intellect: A Conceptual Framework. AFOSR-3233 summary report, October 1962. `http://www.bootstrap.org/augment/AUGMENT/133182-0.html`

FAMILANT, M., AND DETWEILER, M. (1993). Iconic reference: evolving perspectives and an organising framework. In *International Journal of Man-Machine Studies*, 39, pp. 705-728.

FEINER, S., MACINTYRE, B., AND SELIGMANN, D. (1993). Knowledge-based augmented reality. In *Comm. of the ACM*, 36(7), July 1993, 52-62.

FERNANDES, L. (2001). The abacus: the art of calculating with beads. `http://www.ee.ryerson.ca:8080/~elf/abacus/` [Accessed 10 February 2002].

FISHKIN, K., MORAN, T., AND HARRISON, B. (1998). Embodied User Interfaces: Towards Invisible User Interfaces. In *Proceedings of EHCI'98.*

FISHKIN, K. AND STONE, M. (1995). Enhanced Dynamic Queries via Movable Filters. In *Proceedings of CHI 95*, pp. 415-420.

FITZMAURICE, G. (1996). Graspable User Interfaces. Ph.D. Thesis, University of Toronto, 1996.

FITZMAURICE, G., ISHII, H., AND BUXTON, W. (1995). Bricks: Laying the Foundations for Graspable User Interfaces. In *Proceedings of CHI'95*, pp. 442-449.

FJELD, M., LAUCHE, K. , BICHSEL, M., VOORHORST, F., KRUEGER, H., AND RAUTERBERG, M. (2002). Physical and Virtual Tools: Activity Theory Applied to the Design of Groupware. In *Computer Supported Cooperative Work*, special issue on Activity Theory and the Practice of Design, Nardi and Redmiles, eds. 11(1-2), pp. 153-180.

FJELD, M., BICHSEL, M., AND RAUTERBERG, M. (1998). BUILD-IT: An Intuitive Design Tool Based on Direct Object Manipulation. In *Gesture and Sign Language in Human-Computer Interaction, Lecture Notes in Artificial Intelligence,* v.1371, Wachsmut and Fröhlich, eds. Berlin: Springer-Verlag, pp. 297-308.

FLETCHER, R. (2002). Chipless Tags: Design and Implementation. Ph.D. thesis, MIT Media Laboratory.

FOXLIN, E., HARRINGTON, M., AND PFEIFFER, G. (1998). Constellation: A Wide-Range Wireless Motion-Tracking System for Augmented Reality and Virtual Set Applications. In *Computer Graphics (Proceedings of SIGGRAPH'98)*, pp. 371-378.

FRADEN, J. (1997). *Handbook of modern sensors: physics, designs, and applications, 2e.* Woodbury, NY: American Institute of Physics, 1997.

FRANCETIC, A., AND SHWE, H. (2000). Smart Toys: Benefits of Technology-Enhanced Play. In *2000 Game Design Proceedings*, http://www.gdconf.com/archives/proceedings/2000/shwe.doc

FRAZER, J. (1994). *An Evolutionary Architecture.* Architectural Association: London, 1994.

FRAZER, J. (1982). Three-Dimensional Data Input Devices. In *Computers/Graphics in the Building Process*, Washington, 1982.

FREI, P., SU, V., MIKHAK, B., AND ISHII, H. (2000). curlybot: Designing a New Class of Computational Toys. In *Proceedings of CHI'00*, pp. 129-136.

FROHLICH, D. (1997). Direct Manipulation and Other Lessons. In *Handbook of Human-Computer Interaction*, 2e, Ch. 21. Amsterdam: Elsevier, 1997.

FUKUZAKI, Y. (1993). Electronic pen according to the BTRON guideline and its background. In *Tronware*, v4, Personal Media publishers, Japan, 49-62. (cited in [Fitzmaurice 1996, §3.3.10])

GALLOWAY, K., AND RABINOWITZ, S. (1980). Hole in Space.

GENTNER, D., AND NIELSEN, J. (1996). The Anti-Mac Interface. In *Communications of the ACM*, August 1996.

GERSHENFELD, N. (1999). *When Things Begin to Think.* New York: Henry Holt.

GIBSON, J. (1979). *The Ecological Approach to Visual Perception.* New York: Erlbaum Associates, 1979.

GLOS, J. (1997a). Digital Augmentation of Keepsake Objects: A Place for Interaction of Memory, Story, and Self. MS thesis, MIT Media Laboratory.

GLOS, J. (1997b). Rosebud: Technological Toys for Storytelling. In *CHI '97 Extended Abstracts.*

GLOS, J. & UMASCHI, M. (1997). Once Upon an Object: Computationally-augmented Toys for Storytelling. In *ICCIMA '97 Conference Proceedings.* pp. 245-249.

GONZÁLEZ, J. (1995). Autotopographies. In *Prosthetic Territories: Politics and Hypertechnologies.* Westview Press: Boulder, pp. 133-150.

GOODMAN, C. (1987). *Digital Visions: Computers and Art.* New York: Harry Abrams, 1987.

GORBET, M. (1998). Beyond Input Devices: A New Conceptual Framework for the Design of Physical-Digital Objects. MS Thesis, MIT Media Laboratory.

GORBET, M., ORTH, M., AND ISHII, H. (1998). Triangles: Tangible Interface for Manipulation and Exploration of Digital Information Topography. In *Proceedings of CHI'98,* pp. 49-56.

GREEN, M., AND JACOB, R. (1991). Software Architectures and Metaphors for Non-WIMP User Interfaces. In *Computer Graphics,* 25(3), pp. 229-235, July 1991.

GREENBERG, S., AND FITCHETT, C. (2001). Phidgets: Easy Development of Physical Interfaces through Physical Widgets. In *Proceedings of UIST 2001,* pp. 209-218.

GUIARD, Y. (1987). Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. In *The Journal of Motor Behavior,* 19(4), pp. 486-517.

GUIMBRETIERE, F., STONE, M., AND WINOGRAD, T. (2001). Fluid Interaction with High-resolution Wall-size Displays. In *Proceedings of UIST 2001.*

GUTWIN, , C., AND GREENBERG, S. (1998). Design for Individuals, Design for Groups: Tradeoffs Between Power and Workspace Awareness. In *Proceedings of CSCW'98.*

HADIS, M. (1997). Magic & Tangible Bits. Unpublished paper, MIT Media Laboratory.

HALASZ, F. (1991). "Seven Issues" Revisited. Keynote address, Hypertext'91 conference.
http://www.parc.xerox.com/spl/projects/halasz-keynote/ +
http://www.csdl.tamu.edu/~leggett/halasz.html (video)

HALASZ, F. (1988). Reflections on Notecards: Seven Issues for the Next Generation of Hy4permedia Systems. In *Communications of the ACM,* 31(7),. July 1988, pp. 836-852.

HARRIS, R. (1986). *The Origin of Writing.* Chicago: Open Court Publishing.

HARRISON, B., FISHKIN, K., GUJAR, A., MOCHON, C., AND WANT, R. (1998). Squeeze me, hold me, tilt me! An Exploration of Manipulative User Interfaces. In *Proceedings of CHI'98,* pp. 17-24.

HAYWARD, P. (1993). Situating Cyberspace: The Popularisation of Virtual Reality. In *Future Visions: New Technologies of the Screen,* Hayward and Wollen, eds. London: British Film Institute, pp. 180-204. Online at `http://www.rochester.edu/College/FS/Publications/HaywardCyberspace.html`

HEARST, M. (1995). Tilebars: Visualization of term distribution information in full text access. In *Proceedings of CHI '95,* pp. 59-66.

HEATON, K. (2000). Physical Pixels. MS Thesis, MIT Media Laboratory.

HEATON, K., POOR, R., AND WHEELER, A. (1999). Nami. In *Conference Abstracts and Applications of SIGGRAPH'99, Emerging Technologies,* p.214.

HESCHONG, L. (1979). *Thermal Delight in Architecture.* Cambridge: MIT Press.

HINCKLEY, K., PIERCE, J., SINCLAIR, M., AND HORVITZ, E. (2000). Sensing Techniques for Mobile Interaction. In *Proceedings of UIST '00*, pp. 91-100.

HINCKLEY, K., PAUSCH, R., PROFFITT, D., AND KASSELL, N. (1998). Two-Handed Virtual Manipulation. In *ACM TOCHI'98.*

HINCKLEY, K., PAUSCH, R., GOBLE, J., AND KASSEL, N. (1994). Passive Real-World Interface Props for Neurosurgical Visualization. In *Proceedings of CHI'94*, pp. 452-458.

HLUBINKA, M., BEAUDIN, J., TAPIA, E., AND AN, J. (2002). AltarNation: Interface Design for Meditative Communities. In *Extended Abstracts of CHI 2002.*

HOCKETT, C. (1960). The Origins of Speech. In *Scientific American*, 203(3), pp. 89-96.

HOLLAN, J., AND STORNETTA, S. (1992). Beyond Being There. In *Proceedings of CHI'92*, pp. 119-125.

HOLMQUIST, L., REDSTRÖM, J., AND LJUNGSTRAND, P. (1999). Token-Based Access to Digital Information. In *Proceedings of Handheld and Ubiquitous Computing (HUC) 1999*, pp. 234-245.

HOLMQUIST, L., WIGSTRÖM, J. AND FALK, J. (1998). The Hummingbird: Mobile Support for Group Awareness. In *Proceedings of CSCW '98.*

HOUDE, S., AND SALOMON, G. (1993). Working Towards Rich & Flexible File Representations. In *Proceedings of INTERCHI'93*, Adjunct Proceedings, pp. 9-10.

HOURCADE, J., BEDERSON, B. (1999). Architecture and Implementation of a Java Package for Multiple Input Devices (MID). University of Maryland CS-TR-4018, UMIACS-TR-99-26.

HUTCHINS, E. (1995). *Cognition in the Wild.* Cambridge, MA: MIT Press, 1995.

HUTCHINS, E., HOLLAN, J., AND NORMAN, D. (1986). Direct Manipulation Interfaces. In *User Centered System Design: New Perspectives on Human-computer Interaction*, Norman and Draper, eds. Lawrence Erlbaum: Hillsdale, N.J., 1986, pp. 87-124.

IB TECHNOLOGY. (2002). Single chip RFID readers. `http://www.ibtechnology.co.uk/hybrid.htm` [Visited August 4, 2002]

IBM. (2000). Asset ID Information Brief. `http://www5.pc.ibm.com/ww/me.nsf/WW-webdocs/Information+Brief:+Asset+ID` [Visited July 14, 2002]

IBUTTON. (2002). Introducing TINI: Tiny InterNet Interface. `http://www.ibutton.com/TINI/`

INFUSION. (2002). I-Cube: the ultimate MIDI controller. `http://www.infusionsystems.com/products/` [Visited August 5, 2002]

IFRAH, G. (2001). *The Universal History of Computing: From the Abacus to the Quantum Computer.* New York: John Wiley & Sons.

ISHII, H., MAZALEK, A., AND LEE, J. (2001). Bottles as a Minimal Interface to Access Digital Information. In *Extended Abstracts CHI'01*, pp.187-188.

ISHII, H., FLETCHER, R., LEE, J., CHOO, S., BERZOWSKA, J., WISNESKI, C., CANO, C., HERNANDEZ, A., AND BULTHAUP, C. (1999). MusicBottles. In *Abstracts and Applications of SIGGRAPH'99, Emerging Technologies*, p. 174.

ISHII, H., WISNESKI, C., ORBANES, J., CHUN, B., AND PARADISO, J. (1999). PingPongPlus: Design of an Athletic-Tangible Interface for Computer-Supported Cooperative Play. In *Proceedings of CHI'99*, pp. 394-401.

ISHII, H., AND ULLMER, B. (1997). Tangible Bits: Towards Seamless Interfaces between People, Bits, and Atoms. In *Proceedings of CHI'97*, pp. 234-241.

ISHII, H., KOBAYASHI, M., AND ARITA, K. (1994). Iterative Design of Seamless Collaboration Media. In *Communications of the ACM*, 37(8), pp. 83-97.

IWAI, T. (1999). Composition on the Table. In *Visual Proceedings of SIGGRAPH'99.*

JACOB, R., ISHII, H., PANGARO, G., AND PATTEN, J. (2002). A Tangible Interface for Organizing Information Using a Grid. In *Proceedings of CHI 02.*

JACOB, R., LEGGETT, J., MYERS, B. AND PAUSCH, R. (1993). Interaction styles and input/output devices. In *Behaviour & Information Technology*, 12(2), pp. 69-79.

JACOBSON, J., COMISKEY, B., TURNER, C., ALBERT, J., AND TSAO, P. (1997). The Last Book. In *IBM Systems Journal*. 36(3), pp. 457-63.

JFN (JAPAN FORUM NEWSLETTER). (1996). A day in the life: ofuro/bath. `http://www.tjf.or.jp/eng/ge/ge04ofuro.htm`

JEREMIJENKO, N. (1996). Voice Boxes: Memesis. *Ars Electronica'96 Festival Catalog*. New York: Springer, p. 403.

JOHNSON, J., ROBERTS, T., VERPLANK, W., SMITH, D., IRBY, C., BEARD, M., AND MACKEY, K. (1989). The Xerox Star: A Retrospective. In *IEEE Computer*, 22(9), September 1989, pp. 11-29.

JOHNSON, J., AND ROBERTS, T. (1989). Direct Manipulation. In *IEEE Computer*, 22(9), September 1989, pp. 14-15.

JONES, S. (1998). Graphical Query Specification and Dynamic Results Preview for a Digital Library." In *Proceedings of UIST 98*, pp. 143-151.

JONES, W., AND DUMAIS, S. (1986). The Spatial Metaphor for User Interfaces: Experimental Tests of Reference by Location versus Name. In *ACM Transactions on Office Information Systems*, v4(1), January 1986, pp. 42-63.

KATO, H., BILLINGHURST, M., POUPYREV, I., IMAMOTO, K., AND TACHIBANA, K. (2000). Virtual Object Manipulation on a Table-Top AR Environment. In *Proceedings of the International Symposium on Augmented Reality* (ISAR 2000).

KAYE, J. (1999). Symbolic Olfactory Display. MS Thesis, MIT Media Laboratory.

KIRSH, D. (1995). The Intelligent Use of Space. In *Artificial Intelligence*, 1995.

KLEMMER, S., NEWMAN, M., FARRELL, R., BILEZIKJIAN, M., AND LANDAY, J. (2001). The Designers' Outpost: A Tangible Interface for Collaborative Web Site Design. In *Proceedings of UIST'01*, pp. 1-10.

KOWALSKI, J. (2002). Tabula. `http://www.personal.psu.edu/users/w/x/wxk116/roma/tabula.html` [Visited August 5, 2002]

KRAMER, K. (1998). Moveable Objects, Mobile Code. MS Thesis, MIT Media Laboratory.

KRAMER, K., AND MINAR, N. (1997). Stackables: Manipulable Distributed Displays. http://el.www.media.mit.edu/projects/stackables/.

KRAMPEN, M. (1986). "Objects," in *Encyclopedic Dictionary of Semiotics*, T. Sebeok, ed. New York: Mouton de Gruyter, 1986, v1, pp 635-639.

KRUEGER, M. (1983). *Artificial Reality II.* Reading, MA: Addison-Wesley, 1983.

KRÜGER, W., AND FRÖHLICH, B. (1994). The responsive workbench. In *IEEE Computer Graphics and Applications*, May 1994, pp. 12-15.

KUZUOKA, H., AND GREENBERG, S. (1999). Mediating Awareness and Communication through Digital but Physical Surrogates. In *Proceedings of CHI'99 Companion*, pp. 11-12.

LARKIN, J., AND SIMON, H. (1987). Why a Diagram is (Sometimes) Worth Ten Thousand Words. In *Cognitive Science*, v11, 1987, pp. 65-99.

LJUNGSTRAND, P. AND HOLMQUIST, L. (1999a). WebStickers: Using Physical Objects as WWW Bookmarks. In *Extended Abstracts of CHI '99*.

LJUNGSTRAND, P., BJORK, S., AND FALK, J. (1999b). The Wearboy: A Platform for Low-cost Public Wearable Devices. In *Proceedings of ISWC'99*, pp. 195-196.

LÜTJENS, J. (2002). Abacus Online Museum. `http://www.joernluetjens.de/sammlungen/abakus/abakus-en.htm` [Accessed 10 February 2002].

LYNCH, KEVIN. (1960). *The Image of the City*. Technology Press: Cambridge, 1960.

MACKAY, W. (1998). Augmented Reality: linking real and virtual and virtual worlds. In *Proceedings of ACM AVI'98* (published keynote).

MACKAY, W., AND PAGANI, D. (1994). Video mosaic: laying out time in a physical space. In *Proceedings of Multimedia'94*, pp. 165-172.

MACKINLAY, J., ROBERTSON, G., AND CARD, S. (1991). The Perspective Wall: Detail and context smoothly integrated. In *Proceedings of CHI'91*, pp. 173-179.

MACLEAN, K., SHAVER, M. , AND PAI, D. (2002). Handheld Haptics: A USB Media Controller with Force Sensing. In *Proceedings of IEEE HAPTICS 2002*.

MACLEAN, K., SNIBBE, S., AND LEVIN, G. (2000). Tagged Handles: Merging Discrete and Continuous Manual Control. In *Proceedings of CHI'00*, pp. 225-232.

MAEDA, J. (2002). Personal communications.

MAEDA, J., AND MCGEE, K. (1993). *Dynamic Form.* Tokyo: International Media Research Foundation, 1993.

MAES, P., DARRELL, T., BLUMBERG, T., AND PENTLAND, A. (1996). The ALIVE System: Wireless, Full-Body Interaction with Autonomous Agents. In *ACM Special Issue on Multimedia and Multisensory Virtual Worlds*, Spring 1996.

MAIER, D., ULLMAN, J., AND VARDI, M. (1984). On the Foundations of the Universal Relation. In *Transactions On Database Systems*, 9(2), 1984, pp 283-308.

MALONE, T. (1983). How Do People Organize Their Desks? Implications for the Design of Office Information Systems. In *ACM Transactions on Office Information Systems*, 1(1), 1983, pp. 99-112.

MASTERS, J. (2002). The Royal Game of Ur and Tau. `http://www.tradgames.org.uk/games/Royal-Game-Ur.htm` [Visited August 5, 2002]

MATSUSHITA, N., AND REKIMOTO, J. (1997). HoloWall: Designing a Finger, Hand, Body, and Object Sensitive Wall. In *Proceedings of UIST'97*.

MAZALEK, A. (2001). Tangible Interfaces for Interactive Point-of-view Narratives. MS thesis, MIT Media Laboratory.

MCCARTHY, J. (1990). *An Introduction to Theoretical Kinematics.* MIT Press: Cambridge, 1990.

McCloud, S. (1993). *Understanding Comics: The Invisible Art.* London: Rutgers University Press, 1993.

McLuhan, M. (1964). *Understanding Media: The Extensions of Man.*

McCullough, D. (1977). *The path between the seas: the creation of the Panama Canal, 1870-1914.* New York: Simon and Schuster, 1977.

McGee, D., and Cohen, P. (2001). Creating tangible interfaces by augmenting physical objects with multimodal language. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI 2001).*

McNerney, T. (2000). Tangible Programming Bricks: An Approach to Making Programming Accessible to Everyone. MS Thesis, MIT Media Laboratory.

Meikle, J. (1995). *American Plastic: A Cultural History.* Piscataway: Rutgers University Press.

Microchip. (2002). PICmicro® Mid-Range MCU Family Reference Manual.
`http://www.microchip.com/1010/suppdoc/refernce/midrange/`
[Visited August 4, 2002]

Milgram, P., and Kishino, F. (1994). A Taxonomy of Mixed Reality Displays. In *IEICE Transactions on Information Systems,* E77-D(12), December 1994.

Minar, N., Gray, M., Roup, O, Krikorian, R., and Maes, P. (1998). Hive: Distributed Agents for Networking Things. In *Proceedings of the First International Symposium on Agent Systems and Applications.*

Moore, D., Want, R., Harrison, B., Gujar, A., and Fishkin, K. (1999). Implementing phicons, combining computer vision with infrared technology for interactive physical icons" In *Proceedings of UIST'99,* pp. 67-68.

Moore, J., Davis, C., and Coplan, M. (1989). *Building Scientific Apparatus: A Practical Guide to Design and Construction, 2e.* Reading, MA: Addison-Wesley, 1989.

Moran, T., Saund, E., van Melle, W., Gujar, A., Fishkin, K., and Harrison, B. (1999). Design and Technology for Collaborage: Collaborative Collages of Information on Physical Walls. In *Proceedings of UIST'99,* pp. 197-206.

Muller, M., Christiansen, E., Nardi, B., and Dray, S. (2001). Spiritual life and information technology. In *Communications of the ACM,* 44(3), pp. 82-83.

Mynatt, E., Igarashi, T., Edwards, W., LaMarca, A. (1999). Flatland: New Dimensions in Office Whiteboards. In *Proceedings of CHI'99,* pp. 346-353.

Negroponte, N., Ishii, H., and Ullmer, B. (1997). Tangible Bits. In *Wired,* 5(5).
`http://web.media.mit.edu/~nicholas/Wired/WIRED5-05.html`

Negroponte, N. (1995). *Being Digital.* New York: Alfred A. Knopf.

Negroponte, N. (1979). Books Without Pages. In *International Conference on Communications IV.* IEEE 156.56(1): 1-8, 1979.

Nelson, L., Ichimura, S., Pedersen, E., and Adams, L. (1999). Palette: a paper interface for giving presentations. In *Proceedings of CHI'99,* pp. 354-361.

Neurosmith. (1999). MusicBlocks product. `http://www.neurosmith.com/`

Norman, D. (1998). *The Invisible Computer : Why Good Products Can Fail, the Personal Computer Is So Complex, and Information Appliances Are the Solution.* Cambridge: MIT Press.

Norman, D. (1993). *Things that make us smart.* Reading, MA: Addison-Wesley, 1993.

NOVA. (1974). *A Man, A Plan, A Canal: Panama.* Video: WGBH Boston Video, 1974, 1999.

OBA, H. (1990). Environment Audio System for the Future. Sony concept video.

OLSTON, C., WOODRUFF, A., AIKEN, A., CHU, M., ERCEGOVAC, V., LIN, M., SPALDING, M., AND STONEBRAKER, M. (1998). DataSplash. In *Proceedings of SIGMOD 98*, pp. 550-552.

ORTH, M. (2001). Sculpted Computational Objects, with Smart and Active Computing Materials. Ph.D. thesis, MIT Media Laboratory.

PARALLAX. (2002). Basic Stamp: Getting Started. `http://www.parallaxinc.com/html_files/getting_started/gettingstarted.htm`

PARLETT, D. (1999). *Oxford History of Board Games*. Oxford: Oxford University Press.

PATTEN, J., RECHT, B., AND ISHII, H. (2002). AudioPad: A Tag-Based Interface for Musical Performance. In *Proceedings of the International Conference on New Interfaces For Musical Expression.*

PATTEN, J., ISHII, H., HINES, J., AND PANGARO, G. (2001). Sensetable: A Wireless Object Tracking Platform for Tangible User Interfaces. In *Proceedings of CHI 01*, pp.253-260.

PEDERSEN, E., AND SOKOLER, T. (1997). AROMA: abstract representation of presence supporting mutual awareness. In *Proceedings of CHI'97*, pp. 51-58.

PEDERSEN, E., MCCALL, K., MORAN, T., AND HALASZ, F. (1993). Tivoli: an electronic whiteboard for informal workgroup meetings. In *Proceedings of CHI'93*, pp. 391-398.

PERLMAN, RADIA. (1976). Using Computer Technology to Provide a Creative Learning Environment for Preschool Children. MIT Logo Memo #24, 1976.

PETRE, M. (1995). Why Looking Isn't Always Seeing: Readership Skills and Graphical Programming. In *Communications of the ACM*, v38, June 1995, pp. 33-44.

PIPER, B., RATTI, C., AND ISHII, H. (2002). Illuminating Clay: A 3-D Tangible Interface for Landscape Analysis. In *Proceedings of CHI'02*, pp. 355-362.

PLESNIAK, W., AND PAPPU, R. (1998). Coincident Display Using Haptics and Holographic Video. In *Proceedings of CHI'98*, pp. 304-311.

POLYNOR, R. (1995). The Hand That Rocks the Cradle. *I.D.*, May/June 1995, pp. 60-65.

POOR, R. (2001). Embedded networks: pervasive, low-power, wireless connectivity. Ph.D. thesis, MIT Media Laboratory.

POOR, R. (1999). The iRX 2.2 ...where atoms meet bits. `http://web.media.mit.edu/~ayb/irx/irx2/`

POST, E., ORTH, M., RUSSO, P., AND GERSHENFELD, N. (2000). E-broidery: Design and Fabrication of Textile-based Computing. In *IBM Systems Journal*, 39(3&4).

PULLAN, J. (1968). *The History of the Abacus.* New York: Frederick A. Praeger.

REAS, C. (2001). Behavioral Kinetic Sculptures. MIT MS Thesis.

REDSTRÖM, J. (2001). Designing Everyday Computational Things. Göteberg University Ph.D. thesis.

REKIMOTO, J., ULLMER, B., AND OBA, H. (2001). DataTiles: A Modular Platform for Mixed Physical and Graphical Interactions. In *Proceedings of CHI'01*, pp. 269-276.

REKIMOTO, J., AND SCIAMMARELLA, E. (2000). ToolStone: Effective Use of the Physical Manipulation Vocabularies of Input Devices. In *Proceedings of UIST'00*, pp. 109-117.

REKIMOTO, J., AND SAITOH, M. (1999). Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments. In *Proceedings of CHI'99*, pp. 378-385.

REKIMOTO, J. (1997). Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments. In *Proceedings of UIST'97*, pp. 31-39.

RESNICK, M., BERG, R., AND EISENBERG, M. (2000). Beyond Black Boxes: Bringing Transparency and Aesthetics Back to Scientific Investigation. In *Journal of the Learning Sciences*, 9(1). pp. 7-30.

RESNICK, M., MARTIN, F., BERG, R., BOROVOY, R., COLELLA, V., KRAMER, K., AND SILVERMAN, B. (1998). Digital Manipulatives: New Toys to Think With. In *Proceedings of CHI'98*, pp. 281-287.

RESNICK, M. (1993). Behavior Construction Kits. In *Communications of the ACM*, 36(7), July 1993, pp. 64-71.

RETZ-SCHMIDT, G. (1988). Various Views on Spatial Prepositions. In *AI Magazine*, 9(2), pp. 95-105.

REZNIK, D., AND CANNY, J. (2001). C'mon Part, do the Local Motion. In *IEEE International Conference on Robotics & Automation*, pp. 2235-2242.

REZNIK, D., MOSHKOVICH, E., AND CANNY, J. (1999). Building a Universal Part Manipulator. In *Distributed Manipulation*, Bohringer and Choset, ed. Kluwer Academic Press, 1999.

RIDLEY, B., NIVI, B., AND JACOBSON, J. (1999). All-Inorganic Field Effect Transistors Fabricated by Printing." *Science*, 286(5440), pp. 746-749.

ROBERTS, A., ed. (1994). *Staffs of life: rods, staffs, scepters, and wands from the Coudron collection of African art.* Iowa: University of Iowa Museum of Art, 1994.

ROBERTS, L. (1965). Machine perception of three-dimensional solids. In *Optical and electro-optical information processing*, J. Tippett et al., ed. Cambridge, Masachussets, MIT Press, pp. 159-97.

ROLAND. (2002). Modela mini-mill. `http://www.rolanddga.com/products/3D/combo/mdx15/`

SATYANARAYANAN, M. (2001). Pervasive Computing: Vision and Challenges. In *IEEE Personal Communications*, 2001.

SAWHNEY, N.,DODGE, C. (1997). Sandscapes: Expressing Emergent Temporal Patterns. Unpublished paper, MIT Media Laboratory. `http://www.media.mit.edu/~nitin/papers/sand.html`

SCAIFE, M., AND ROGERS, Y. (1996). External Cognition: How Do Graphical Representations Work? In *International Journal of Human-Computer Studies*, 45(2), pp. 185-213.

SCHÄFER, K., BRAUER, V., AND BRUNS, W. (1997). A new approach to human-computer interaction—synchronous modelling in real and virtual spaces. In *Proceedings of DIS'97*, pp.335-344

SCHKOLNE, S., PRUETT, M., AND SCHRÖDER, P. (2001). Surface Drawing: Creating Organice 3D Shapes with the Hand and Tangible Tools. In *Proceedings of CHI'01*, pp. 261-268.

SCHMANDT-BESSERAT, D. (1997). *How Writing Came About.* Austin: University of Texas Press.

SHNEIDERMAN, B. (1983). Direct manipulation: A step beyond programming languages. In *IEEE Computer*, 16(8), pp. 57-69.

SIIO, I. (1995). InfoBinder: A Pointing Device for a Virtual Desktop System. In *Proceedings of the IHCI'95.*

SIMPSON, R., RENEAR, A., MYLONAS, E., VAN DAM, A. (1996). 50 Years After "As We May Think": The Brown/MIT Vannevar Bush Symposium. In *Interactions*, March 1996, pp. 47-67.

SINGER, A., HINDUS, D., STIFELMAN, L., AND WHITE, S. (1999). Tangible Progress: Less is More in Somewire Audio Spaces. In *Proceedings of CHI'99*, pp. 104-111.

SMALL, D. (1999). Rethinking the Book. Ph.D. Thesis, MIT Media Laboratory.

SMALL, D. (1996). Navigating Large Bodies of Text. In *IBM Systems Journal*, 35(3-4), pp. 515-525.

SMITH, D. (1975). Pygmalion: A Creative Programming Environment. Ph.D. Thesis, Stanford University.

SMITH, R. (1987). Experiences with the alternate reality kit: an example of the tension between literalism and magic. In *Proceedings of CHI'87*, pp. 61-67.

SNIBBE, S., MACLEAN, K, SHAW, R., RODERICK, J., VERPLANK, W., AND SCHEEFF, M. (2001). Haptic Techniques for Media Control. In *Proceedings of UIST 01*.

SONY. (2002). What is Memory Stick? `http://www.memorystick.org/msinfo/eng/ms/ms.html` [Visited July 14, 2002]

STARNER, T., MANN, S., RHODES, B., LEVINE, J., HEALEY, J., KIRSCH, D., PICARD, R., AND PENTLAND, A. (1997). Augmented Reality through Wearable Computing. In *Presence*, 6(4), Winter 1997.

STEFIK, M., FOSTER, G., BOBROW, D., KAHN, K., LANNING, S., AND SUCHMAN, L. (1987). Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings. In *Communications of the ACM*, 30(1), pp. 32-47.

STIFELMAN, L. ARONS, B., AND SCHMANDT, C. (2001). The Audio Notebook: Paper and Pen Interaction with Structured Speech. In *Proceedings of CHI'01*, pp. 182-189.

STIFELMAN, L. (1996). Augmenting Real-World Objects: A Paper-Based Audio Notebook. In *Proceedings of CHI'96 Companion*, pp. 199-200.

STREECK, J., AND COLE, C. (2001). Evolving Media: Visualization; Writing: Tokens. `http://studio.cocomm.utexas.edu/Streeck_Air-UT/writing/tokens.html` [Accessed 10 February 2002].

STREITZ, N., TANDLER, P., MÜLLER-TOMFELDE, C., AND KONOMI, S. (2001). Roomware: Toward the Next Generation of Human-Computer Interaction Based on an Integrated Design of Real and Virtual Worlds. *Human-Computer Interaction in the New Millenium*, John M. Carroll, ed., pp. 553-678.

STREITZ, N. (2001). Mental vs. Physical Disappearance: The Challenge of Interacting with Disappearing Computers. Position paper in CHI 2001 workshop on Distributed and Disappearing UIs in Ubiquitous Computing. `http://www.teco.edu/chi2001ws/20_streitz.pdf`

STREITZ, N., GEIßLER, J., HOLMER, T., KONOMI, S., MÜLLER-TOMFELDE, C., REISCHI, W., REXROTH, P., SEITZ, P., AND STEINMETZ, R. (1999). i-LAND: an interactive landscape for creativity and innovation. In *Proceedings of CHI'99*, pp. 120-127.

STRICKLAND, R. (1998). Portable Effects: A Design Vocabulary of Everyday Life. `http://adaweb.walkerart.org/~dn/a/enfra/rstrickland.html` [Visited August 2002].

STRICKON, J., AND PARADISO, J. (1998). Tracking Hands Above Large Interactive Surfaces with a Low-Cost Scanning Laser Rangefinder. In *Proceedings of CHI '98*. pp. 231-232.

SUZUKI, H., KATO, H. (1993). AlgoBlock: a Tangible Programming Language, a Tool for Collaborative Learning. In *Proceedings of 4th European Logo Conference*, pp. 297-303.

SUTHERLAND, I. (1968). A Head-mounted Three Dimensional Display. In *Proceedings of the Fall Joint Computer Conference*, 1968, pp. 757-764.

SVANÆS, D., AND VERPLANK, W. (2000). In Search of Metaphors for Tangible User Intefaces. In *Proceedings of DARE'00*, pp. 121-129.

SWIFT, J. (1726). *Gulliver's Travels.*

SYSTRONIX. (2002). SaJe and JStamp. `http://saje.systronix.com/`

TEN HAGEN, P. (1981). Interaction and Syntax. In *International Journal of Man-Machine Studies*, v15.

THACKARA, J. (2001). The Design Challenge of Pervasive Computing. In *Interactions*, May-June 2001, pp. 46-52.

TOLLMAR, K., JUNESTRAND, S., AND TORGNY, O. (2000). Virtually Living Together: A Design Framework for New Communication Media. In *Proceedings of DIS'00*, pp. 83-91.

TOMOE SOROBAN CO., LTD. (2002). Soroban museum: Roman Soroban. `http://www.soroban.com/museum/5s_eng.html` [Accessed 10 February 2002].

TOWNSHEND, P. (1986). Games in culture: A contextual analysis of the Swahili board game and its relevance to variation in African mankala. Ph. D. thesis, Cambridge, Department of Anthropology.

ULLMER, B., AND ISHII, H. (2001). Emerging Frameworks for Tangible User Interfaces. In *HCI in the New Millenium*, John M. Carroll, ed., pp. 579-601.

ULLMER, B., KIM, E., KILIAN, A., GRAY, S., AND ISHII, H. (2001). Strata/ICC: Physical Models as Computational Interfaces. In *CHI'01 Extended Abstracts*, , pp. 373-374.

ULLMER, B., AND ISHII, H. (2000). Emerging Frameworks for Tangible User Interfaces. In *IBM Systems Journal*, v39, n3-4, pp. 915-931.

ULLMER, B. (2000). Cognitive Roles of Physical Artifacts in Non-Athletic Games. Unpublished manuscript.

ULLMER, B., AND ISHII, H. (1999). mediaBlocks: Tangible Interfaces for Online Media. In *CHI'99 Extended Abstracts* (video demonstration), pp. 31-32. `http://web.media.mit.edu/~ullmer/projects/mediablocks/`

ULLMER, B., AND ISHII, H. (1999b). musicbox: A Tangible Interfaces for Remote Presence and Browseable History. Submitted to UIST'99. `http://web.media.mit.edu/~ullmer/papers/musicbox.pdf`

ULLMER, B., ISHII, H., AND GLAS, D. (1998). mediaBlocks: Physical Containers, Transports, and Controls for Online Media. In *Computer Graphics Proceedings (SIGGRAPH'98)*, pp. 379-386.

ULLMER, B. (1997). Models and Mechanisms for Tangible User Interfaces. MS thesis, MIT Media Laboratory.

ULLMER, B., AND ISHII, H. (1997). The metaDESK: Models and Prototypes for Tangible User Interfaces. In *Proceedings of UIST'97*, pp. 223-232.

UMASCHI, M. (1997). Soft Toys with Computer Hearts: Building Personal Storytelling Environments. In *CHI'97 Extended Abstracts*, pp. 20-21.

UNDERKOFFLER, J., AND ISHII, H. (1999). Urp: A Luminous-Tangible Workbench for Urban Planning and Design. In *Proceedings of CHI'99*, pp. 386-393.

UNDERKOFFLER, J., ULLMER, B., AND ISHII, H. (1999). Emancipated Pixels: Real-World Graphics in the Luminous Room. In *Computer Graphics Proceedings (SIGGRAPH'99)*, pp. 385-392.

UNDERKOFFLER, J., AND ISHII, H. (1998). Illuminating Light: An Optical Design Tool with a Luminous-Tangible Interface. In *Proceedings of CHI'98*, pp. 542-549.

UNIVERSAL LASER SYSTEMS (ULS). (2002). `http://www.ulsinc.com/basics.html`

VERPLANK W., FULTON J., BLACK A., AND MOGGRIDGE, W. (1993). Observation and Invention: The Use of Scenarios in Interaction Design. Tutorial Notes from *Conference on Human Factors in Computing Systems: INTERCHI'93.*

VIHMA, S., ED. (1990). *Semantic Visions in Design.* Helsinki: UIAH, 1990.

WACHTEL, E. (1999). How Writing Came About (book review). In *Technology and Culture,* 40(1), pp. 128-129.

WANT, R., FISHKIN, K., GUJAR, A., AND HARRISON, B. (1999). Bridging Physical and Virtual Worlds with Electronic Tags. In *Proceedings of CHI'99*, pp. 370-377.

WATSON, J. (1968). *The Double Helix.* New York: Athenium.

WEINBERG, G., ORTH, M., AND RUSSO, P. (2000). The Embroidered Musical Ball: A Squeezable Instrument for Expressive Performance. In *Extended Abstracts of CHI 00.*

WEISER, M. AND BROWN, J. (1995). Designing Calm Technology. `http://www.ubiq.com/hypertext/weiser/-calmtech/calmtech.htm` December 1995.

WEISER, M. (1994). Creating the Invisible Interface (Invited talk). In *Proceedings of UIST'94*, p. 1.

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

WELLNER, P. (1993). Interacting with paper on the Digital Desk. In *Communications of the ACM.* July 1993, pp. 86-96.

WELLNER, P. (1991). The DigitalDesk Calculator: Tangible Manipulation on a Desk Top Display. In *Proceedings of UIST 1991.* pp. 27-33.

WILLIAMSON, C., AND SHNEIDERMAN, B. (1992). The Dynamic HomeFinder: Evaluating Dynamic Queries in a Real-Estate Information Exploration System. In *Proceedings of SIGIR'92*, pp. 339-346.

WINOGRAD, T., AND FLORES, F. (1986). *Understanding Computers and Cognition.* Reading, MA.: Addison-Wesley.

WISNESKI, C. (1999). The Design of Personal Ambient Displays. MS thesis, MIT Media Laboratory.

WISNESKI, C., ISHII, H., DAHLEY, A., GORBET, M., BRAVE, S., ULLMER, B. AND YARIN, P. (1998). Ambient Displays: Turning Architectual Space into an Interface between People and Digital Information. In *Proceedings of CoBuild'98*, pp. 22-32.

YARIN, P. (1999). Towards the Distributed Visualization of Usage History. MS thesis, MIT Media Laboratory.

YARIN, P., AND ISHII, H. (1999). TouchCounters: Designing Interactive Electronic Labels for Physical Containers. In *Proceedings of CHI'99*, pp. 362-368.

YIM, M., REICH, J., AND BERLIN, A. (2000). Two Approaches to Distributed Manipulation. In *Distributed Manipulation*, Choset and Bohringer, eds. Kluwer Academic Publishing, 2000.

ZHANG, J. (1999). Personal communications. August 30, 1999.

ZHANG, J. (1997). The nature of external representations in problem solving. In *Cognitive Science,* 21(2), pp. 179-217.

ZHANG, J. AND NORMAN, D. (1994). Representations in distributed cognitive tasks. In *Cognitive Science,* 18, pp. 87-122.