Token+constraint systems for tangible interaction with digital information

BRYGG ULLMER
Zuse Institute Berlin (ZIB)
and
HIROSHI ISHII
MIT Media Laboratory
and
ROBERT J. K. JACOB
Tufts University

We identify and present a major interaction approach for tangible user interfaces based upon systems of tokens and constraints. In these interfaces, tokens are discrete physical objects which represent digital information. Constraints are confining regions that are mapped to digital operations. These are frequently embodied as structures that mechanically channel how tokens can be manipulated, often limiting their movement to a single degree of freedom. Placing and manipulating tokens within systems of constraints can be used to invoke and control a variety of computational interpretations.

We discuss the properties of the token+constraint approach; consider strengths that distinguish them from other interface approaches; and illustrate the concept with eleven past and recent supporting systems. We present some of the conceptual background supporting these interfaces, and consider them in terms of Bellotti et al.'s five questions for sensing-based interaction. We believe this discussion supports token+constraint systems as a powerful and promising approach for sensing-based interaction.

Categories and Subject Descriptors: H.5.1 [**Multimedia Information Systems**] Artificial, augmented, and virtual realities; H.5.2 [**User Interfaces**] Input devices and strategies

Additional Key Words and Phrases: tangible interfaces, token+constraint interfaces

---

The research underlying this paper was conducted as Ph.D. work within the MIT Media Laboratory.
This work was supported in part by IBM, Steelcase, Intel, and other sponsors of the MIT Media Laboratory's Things That Think and Digital Life consortiums. The paper was also supported by Hans-Christian Hege (Zuse Institute Berlin / ZIB) and the EC "GridLab" project, IST-2001-32133.

Authors' addresses: Brygg Ullmer, Visualization Department, Zuse Institute Berlin, Takustrasse 7, Berlin, 14195, Germany, ullmer@zib.de; Hiroshi Ishii, Tangible Media Group, MIT Media Laboratory, 20 Ames St., E15, Cambridge, MA, 02141, USA, ishii@media.mit.edu; Robert J. K. Jacob, Department of Computer Science, Tufts University, Halligan Hall, 161 College Avenue, Medford, MA, 02155, USA, jacob@cs.tufts.edu.

1. INTRODUCTION

Tangible user interfaces (TUIs) are one of several genres of sensing-based interaction that has attracted significant attention during recent years. Broadly viewed, tangible interfaces give physical form to digital information. The approach has two basic components. 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 transformed into several questions. First, what kinds of information and operations might one wish to represent and manipulate with physical objects? And second, what kinds of physical / digital systems might be used to mediate these interactions?

Several major approaches have evolved that illustrate possible answers to these questions [Ullmer and Ishii 2001]. Likely the most popular application of tangible interfaces has been using 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, buildings [Underkoffler et al. 1999], and furniture [Fjeld et al. 1998]. These particular instances illustrate an *interactive surfaces* approach, with users manipulating 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 approach that has also been used for the physical modeling of physical systems draws inspiration from building blocks and LEGO™. Such *constructive assemblies* of modular, interconnecting elements have been used for modeling buildings [Aish 1984; Frazer 1994; Anderson et al. 2000], fluid flow [Anagnostou et al. 1989], and other geometrical forms [Anderson et al. 2000].

These examples provide several possible answers to our leading questions. While interactive surfaces and constructive assemblies have broader applications, they have most often been 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. For example, the ability to save and retrieve digital state is important across the full spectrum of computational systems, but this capability has no intrinsic physical representation.

We present a third approach for physically interacting with digital information which, while illustrated by a number of past and present systems, has not been articulated in previous publications. This approach combines two kinds of physical/digital artifacts: *tokens* and *constraints*. In these interfaces, physical tokens are used to reference digital information. Physical constraints are used to map structured compositions of these tokens onto a variety of computational interpretations. This is loosely illustrated in Figure 1.

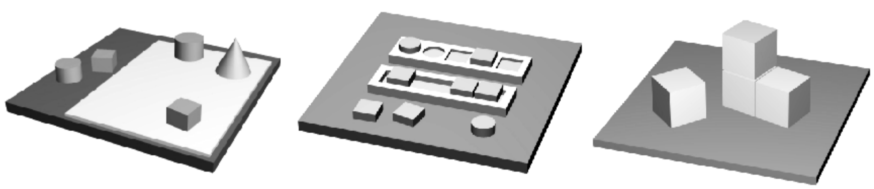

Figure 1a,b,c: Loose illustrations of interactive surface, token+constraint, and constructive assembly approaches

Token+constraint systems are most often used to interact with abstract digital information that has inherent physical representation, nor any intrinsic physical language for its manipulation. Token+constraint systems both extend the space of tasks for which tangible interfaces may productively be used, and complement other computational interfaces (whether tangible or otherwise) that can benefit from these tasks. While systems employing the interactive surface and constructive assembly approaches have also begun to see use for manipulating abstract information, token + constraint systems offer a number of additional, complementary benefits that support them as a powerful approach for tangible interface design.

In the following pages, we will discuss some of the properties of token+constraint interfaces. We continue with a discussion of conceptual background, and concretely illustrate the token + constraint approach with a number of example interfaces. We then consider token+constraint systems from the perspective of the five questions for sensing-based interaction articulated within [Bellotti et al. 2002], and conclude with a discussion.

2. TOKEN + CONSTRAINT INTERFACES

Human interaction with physical artifacts frequently involves the manipulation of objects that are subject to some form of mechanical constraint. This relationship between objects and constraints is usually observable with both visual and haptic modalities, and draws on some of humans' most basic knowledge about the behavior of the physical world. The interaction between objects and constraints also has important implications for human performance. Writing on the topic of two-handed interaction, 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 1998]

Token+constraint interfaces are a class of tangible interfaces that build upon relationships between systems of physical tokens and constraints (Figure 2). In the context of this paper, tokens are discrete, spatially reconfigurable physical objects that typically represent digital information. Constraints are confining regions within which tokens can be placed. These regions are generally mapped to digital operations, which are applied to tokens located within the constraint's perimeter.

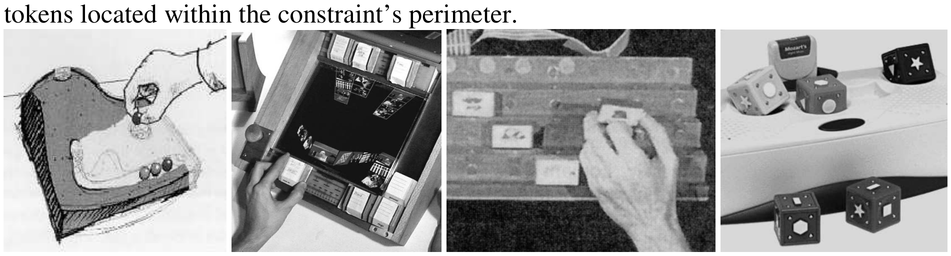

Figure 2: Examples of token+constraint approach: Marble Answering Machine [Polynor 1995], mediaBlocks [Ullmer et al. 1998], LogJam [Cohen et al. 1999], Music Blocks [Neurosmith 1999]

We use the phrase "token+constraint" to express the close interdependency between these two elements. Just as computational expressions typically require both operators and operands, tokens and constraints must be combined together to compose fully formed computational expressions. Even when tokens and constraints are physically separated, their physical "complementarities" to each other enable them to passively express allowable combinations and alternative usage scenarios.

In this paper, constraints are embodied as physical structures that mechanically channel how "child" tokens can be manipulated, each limiting the movement of individual child tokens to (at most) a single physical degree of freedom. Other variations on this approach are possible. For example, constraints may be expressed as visual regions that are not mechanically confining. Conversely, mechanical constraints may be used to "confine" graphical elements which are not themselves physically embodied. While we will consider these variations in the discussion, this paper focuses upon interactions between mechanical constraints and embodied physical tokens.

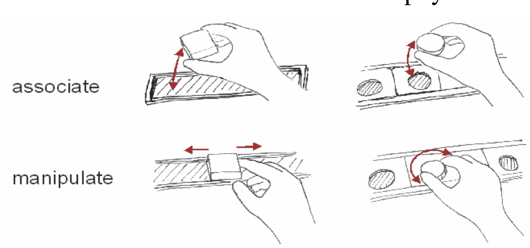

Figure 3: Illustration of token+constraint interfaces' two phases of interaction.

Token+constraint interfaces have two phases of interaction: *associate* and *manipulate*. These are illustrated in Figure 3. 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 usually 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 corresponding digital bindings and interpretations.

Some token+constraint interfaces support only the "associate" phase of interaction. However, many token+constraint interfaces also support a second "manipulate" phase, where tokens may be manipulated within the confines of this constraint. In this case, when placed within a constraint, tokens are usually constrained mechanically to move with a single degree of freedom. Specifically, the token may be translated along a linear axis, or turned about a rotational axis. These relationships are illustrated in Figure 4.

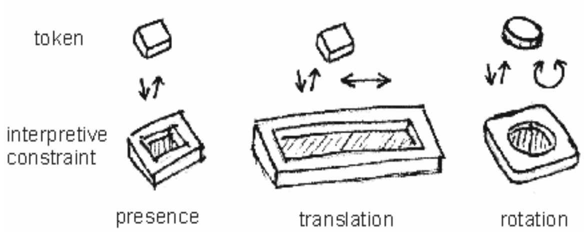

Figure 4a,b,c: Basic token/constraint combinations: presence; presence+translation; and presence+rotation.

Several additional examples are illustrated in Figure 5. First, tokens can be transferred between different constraints to apply different digital operations. Second, some constraints can contain multiple physical tokens, whether of one kind or multiple different kinds. In these cases, the relative and absolute positions of tokens, both with respect to each other and to the constraint, can all potentially map to different interpretations. The token+constraint relationship can also be nested. A physical artifact can serve both as a parent constraint for one or more child tokens, and simultaneously as a child token within a larger frame of reference. The game of "Trivial Pursuit™" provides a familiar example in its "pie" tokens, which each have receptacles for six child "wedges."

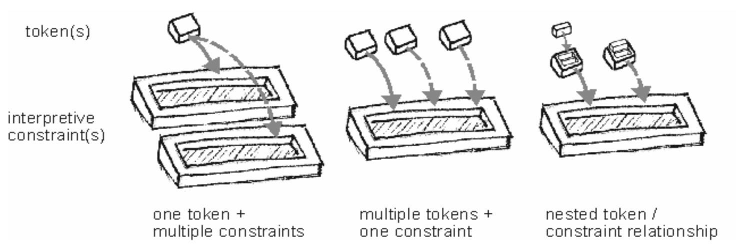

Figure 5: More complex combinations of tokens and constraints: one token + multiple separate constraints; multiple tokens + a single constraint; nested token/constraint relationships

Another important aspect of the associate and manipulate phases of interaction is that they often correspond with discrete and continuous modalities of interaction. This observation has also been discussed in related terms within [MacLean et al. 2000]. 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 spatially continuous interactions with tokens within the confines of a parent constraint. Token+constraint interfaces thus support the benefits of both discrete expressions (e.g., commands and discrete relationships), as well as continuous ones (e.g., manipulating continuous scalar values and indices within information aggregates).

In some respects, token+constraint interfaces realize a kind of simple physical/digital "language," allowing open-ended combinations of physically-embodied operations and operands. While several tangible interfaces have explicitly pursued the idea of a tangible programming language [Perlman 1976; Suzuki and Kato 1993; McNerney 2000], most token+constraint interfaces do not share this orientation. Instead of the deliberate, cumulative expressions of most programming languages, token+constraint interfaces are generally used to embody interactive workspaces where physical actions bring an immediate interpretation and response by the system. In this respect, the approach closely follows the principles of "direct manipulation" articulated in [Shneiderman 1983].

2.1 Physical expressions of digital syntax

A key property of token+constraint interfaces is that they give physical form not only to digital information itself, but also to aspects of the syntax for manipulating this information. Syntax is defined by the Oxford English Dictionary 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 both by users and the underlying computational system.

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 GUIs. Software and graphics alone cannot physically enforce consistency in configurations of discrete physical objects. By mechanically structuring and limiting which tokens can be accommodated and what configurations these can assume, constraints can express and partially enforce the syntax of their associated digital operations.

The token+constraint approach can be seen as developing a hierarchical syntax, with "child" tokens placed within or removed from compatible "parent" constraints. Compatibility and complementarity are often expressed with the physical shape of the tokens and constraints, with incompatible elements rendered incapable of mechanically engaging with each other.

When viewed from the perspective of computer science and "object-oriented" programming, the token+constraint approach illustrates a kind of "multiple inheritance." When placed within a constraint, tokens are often used to simultaneously represent both the *container* for a chunk of digital information, and also the *control* for acting upon this 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.

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. Constraints also help to support consistency by mechanically restricting the physical relationships that objects can express. However, constraints do not fully express the syntax of physical / digital expressions, or eliminate the possibility of invalid expressions. Speaking broadly of this issue, 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]

2.2 Examples of token+constraint mappings

One recurring example of constraints is the use of "racks" that structure the manipulation of physical tokens within a linear constraint [Ullmer et al. 1998; Cohen et al. 1999; Singer et al. 1999; Ullmer et al. 2003]. Several example configurations of racks and tokens are illustrated in Figure 2b,c. These configurations are the product of combining several basic physical properties. Specifically, these configurations can be described in terms of the relative and absolute positions of tokens, both with respect to the constraint and to each other. This observation builds on ideas about spatial prepositions from disciplines including linguistics, psychology, and artificial intelligence, which discuss related ideas in terms of "primary objects," "reference objects," and "reference frames" [Retz-Schmidt 1988]. More carefully stated, the physical relationships between tokens and constraints can be understood in terms of four basic relationships:

a) Absolute configuration of token(s) with respect to constraint
b) Relative configuration of token(s) with respect to constraint
c) Absolute configuration of tokens with respect to each other
d) Relative configuration of tokens with respect to each other

Table 1: Different physical relationships between tokens and constraints

These abstract physical relationships can be mapped onto a number of specific digital interpretations. Several of these are summarized in Table 2: Grammars for mapping physical relationships to digital interpretations. Many of these particular mappings will be illustrated concretely in the example systems of §4 and §5.

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

Table 2: Grammars for mapping physical relationships to digital interpretations

2.3 Strengths of token+constraint approach

It is useful to summarize some of the strengths of the token+constraint approach. In some cases, our points should be considered as potential benefits or goals that may not always be present, and may benefit from empirical validation. It is also important to note that the physical relationships and physical/digital grammars of Tables 1 and 2 are not limited to token+constraint approaches. For example, the same relationships can also be expressed within interactive surface interfaces, which usually possess a superset of the physical degrees of freedom of token+constraint approaches. Nonetheless, when compared with interactive surfaces, the use of physical constraints offers a number of benefits, including:

1) increased passive haptic feedback;
2) increased prospects for active force feedback;
3) decreased demands for visual attention;
4) increased kinesthetic awareness;
5) increased prospects for embedded uses; and
6) flexible, widely accessible sensing technologies.

Many of these benefits draw from the styles of physical embodiment employed by the token+constraint approach. Specifically, the use of physically embodied, mechanically confining constraints helps to express:

- *the set of physical tokens that can take part within a given constraint:* The mechanical structure of constraints can help express physical/digital compatibilities with subsets of tokens, as encoded in physical properties such as size and shape.
- *the set of physical configurations these physical tokens can take on:* Tokens are often mechanically restricted to configurations that have well-defined computational interpretations
- *the demarcation between interaction regions with different computational interpretations:* The well defined boundaries of constraints are an aid to combining

and integrating multiple constraints, each potentially with different behaviors. These boundaries also aid the integration of constraints into self-contained devices.

Viewed from a somewhat different perspective, the use of physical constraints has other positive ramifications from both usage and implementational standpoints. These include:

- *human perception* – constraints use physical properties to perceptually encode digital syntax. Among other things, they shift cognitive load to external representations (see §3.2.1), and support perceptual chunking of object aggregates.
- *human manipulation* – constraints provide users with an increased sense of kinesthetic feedback, stemming from the passive haptic feedback provided by token/constraint ensembles. 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 of tokens), or through actions like "sweeping" a series of multiple tokens which are jointly (e.g., by a rack).
- *machine sensing* – constraints can significantly simplify the sensing of a tangible interface's physical state. This can ease implementation, increase scalability, and increase flexibility in the physical forms that tangible interfaces can assume.
- *machine interpretation* – constraints can simplify the underlying computational interpretation of the physical objects composing a tangible interface, by virtue of limiting them to a smaller space of relatively well-defined states. This is both an implementational aid, and can help to minimize error conditions.

3. CONCEPTUAL BACKGROUND

Humans are clearly no newcomers to interaction with the physical world, or to the process of associating symbolic functions and relationships with physical artifacts. In this section we consider some of the conceptual background underlying token+constraint systems. We begin by considering several historical examples – the abacus and board games – which are both inspirations for the token+constraint approach, and suggestive of potential interaction genres [Bellotti et al. 2002]. Next, we overview several closely related areas of study from psychology and cognitive science. Finally, we briefly review work in the discipline of human-computer interaction, reviewing several principles and models in the context of tokens and constraints.

3.1 Motivating examples

3.1.1 The abacus

The abacus and board games offer classes of physical artifacts that are inspirational to the token+constraint interface approach. Both are believed to date back on the order of 5000 years to Mesopotamian origins among the earliest civilizations of recorded history [Ifrah 2001; Bell 1979; Masters 2002]. The earliest versions of the abacus are believed to have Sumerian origins on the order of 2700 BC [Ifrah 2001], and may in turn have roots in clay accounting tokens dating back to 8000 BC [Schmandt-Besserat 1997] (thus predating written language and even the wheel).

The abacus is believed to have originated with the use of tokens upon marked or grooved boards or tables (tabula). In some instances, deeply grooved lines served as constraints for spherical tokens (Figure 6a). The use of rods and beads within the abacus appeared in ca. 1200 AD in China as the "suan pan," and was adopted in Japan as the "soroban" ca. 1600 AD (Figure 6b). Interestingly, a related abacus form of Aztec origins (the "nepohualtzitzin"), composed of kernels of maize threaded through strings mounted upon a wooden frame, may also have been used ca. 900-1000 AD [Fernandes 2001; Lütjens 2002; Tomoe 2002; Durham 2002a,b].

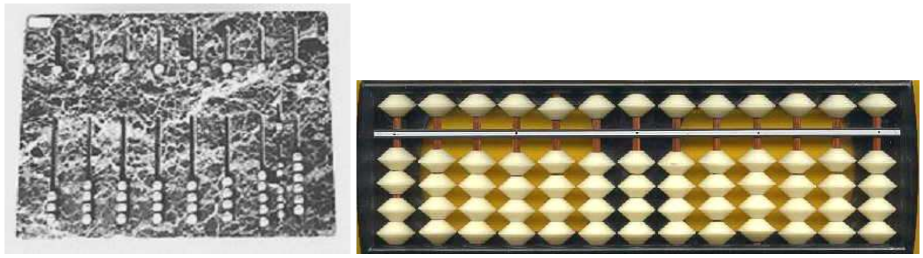

Figure 6a,b: Roman tabula, pebbles constrained within grooves [Tomoe 2002]; Japanese soroban [Lutjens 2002]

The abacus represents information not just as discrete physical beads, but also through the spatial structuring and configuration of these elements within the "constraints" of the counting board and rods. While the pragmatics of mobility and managing numerous physical elements eventually pushed the abacus to a system of captive beads, abacus tokens remained removable and spatially reconfigurable for much of the device's history. As evidenced by the deeply grooved counting board of Figure 6a, some abacus devices closely approximated the token+constraint approach.

The abacus remains in use by some adults in East Asia, and in the West "counting boards" are commonly used in elementary education. However, the abacus has passed out of active "professional" use in the West for on the order of 500 years. Still, shadows of the abacus can be found in many token+constraint interfaces, with tokens representing abstractions like "images" or "people" rather than digits, and projected graphics or other displays used to bring alive computational mediations within their physical frames.

3.1.2 Board games

Board, card, and tile games present another richly populated class of physical artifacts 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 poker 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 7, imagining the physical tokens as digitally representing people, places, devices, data structures, and software, with the board "constraints" embodying the "syntax" used to compose mixed physical and computational expressions, provides a stimulating point of departure for envisioning potential token+constraint TUIs.

Board games offer compelling examples for how abstract rules and relationships can be encoded within systems of physical objects. For example, Monopoly™ utilizes distinctive physical tokens as representations of people (player tokens), physical entities (house & hotel tokens), money, actions (through several kinds of cards), and elements of chance (the dice). The Monopoly™ board expresses the framing syntax for composing and interpreting these tokens within the visual "constraints" printed upon its surface. These artifacts also express a range of physical properties governing their 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. Games also afford interaction not only between people and information, but also between multiple people, in a compelling and engaging fashion.

Board games can suggest specific physical elements and actions that can be employed within tangible interfaces. For example, the "rack" structure's use within the mediaBlocks system [Ullmer et al. 1998] was partly inspired by two such examples: "word blocks" and the Scrabble™ game's tile rack. In both instances, a series of physical tokens are constrained within a linear constraint to facilitate the composition of words or sentences. While the object configurations of board games are interpreted only within the mind of the human user, they broadly lend themselves to the variety of computational interpretations and mediations discussed within this paper.

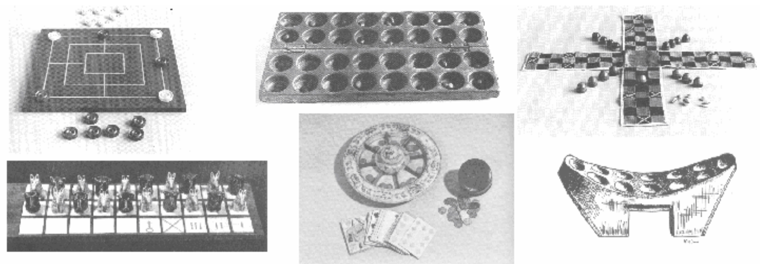

Figure 7: Example board games (Nine Men Morris; Mancala; Parcheesi; Game of Thirty; Pope Joan; Awari)

3.2 Perspectives from psychology and cognitive science

Psychology and cognitive science offer one of the broadest areas of scientific study related to tangible interfaces. This is partially 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 (as illustrated by the abacus and board games) and broader range of modalities for engagement between people and computation than GUIs. These factors contribute to the relevance of an even broader range of subdisciplines. In this section, we discuss the representational aspects of token+constraint interfaces from the perspectives of external representation, distributed cognition, and affordances.

3.2.1 External representations and distributed cognition

Cognitive scientists are approaching 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].

Drawing from a series of cognitive studies, Zhang and Norman assert 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 concludes that "external representations are neither mere inputs and stimuli 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]. Elaborating on this, 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" [personal communications, 1999].

A related topic is the distinction between people's use of their hands for physical performance versus exploration. Human manipulation of objects can be divided into of "exploratory" and "performatory" actions [Gibson 1979], or alternately "epistemic" and "pragmatic" actions [Kirsh 1995]. Exploratory/epistemic actions are performed to uncover information that is hidden or hard to compute mentally. This perspective relates to the distinction of "in-band" vs. "out-of-band" interactions with TUI elements. In-band manipulations of tokens are sensed and interpreted by the computational system. In contrast, out-of-band manipulations may or may not be sensed or computationally mediated, but are not interpreted by the TUI as expressing specific actionable commands.

Out-of-band manipulations can be seen as serving important exploratory, epistemic roles. Out-of-band manipulations are far more easily employed within tangible interfaces than GUIs, given the porous boundaries between tangible interfaces and the surrounding physical world. The token+constraint approach facilitates the delineation between in-band and out-of-band, in that tokens outside of constraints are usually out-of-band. Token manipulation within constraints can be either in-band or out-of-band, depending upon the interface's specific semantics. The corresponding interpretation should generally be clarified by computational mediation, as we will discuss in §6.2.1.

3.2.2 Affordances

Ideas about "affordances" by Gibson, Norman, and others have long been of interest to the HCI community, and hold special relevance for TUI design. Affordances are the physical traits of an artifact that suggest how a person (or animal) can engage with the object. 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]

From the perspective of constraints, Norman goes on to add:

> Physical constraints are closely related to real affordances: For example, it is not possible to move the cursor outside the screen [though Rekimoto has shown compelling realizations of this]…. Physical constraints make some activities impossible: there is no way to ignore them. [Norman 1999]

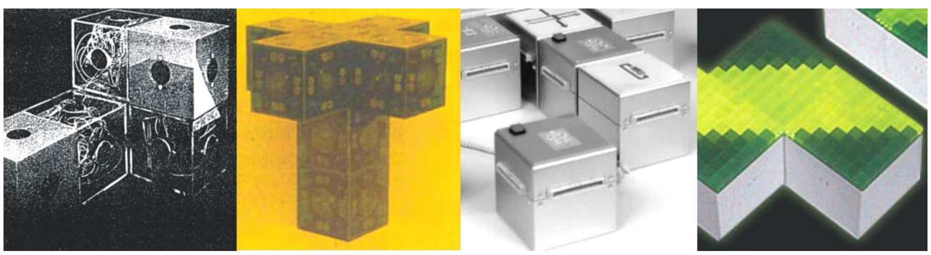

Figure 8: Cubes of Frazer [1982], Anagnostou et al. [1989], Suzuki and Kato [1993], Shießl [2001]

These observations have a number of implications. For example, 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 [1982], Anagnostou et al. [1989], Suzuki and Kato [1993], and Shießl [2001] all independently converged upon cubes of roughly 10cm/side (Figure 8) – not far from the "five-inch cube" referred to by Gibson.

Similarly, a number of token+constraint systems (e.g., mediaBlocks) have converged on tokens of roughly 5cm/side. These sizes seem to reflect the anatomy of the human hand. In classifications of hand postures by Cutkosky and Howe [1990], the 10cm cube corresponds to a power grasp, while the 5cm sizes corresponds to a precision grasp.

3.3 Models for human-computer interaction

A number of models and perspectives from HCI hold relevance to the study of tangible interfaces, and are surveyed in [Ullmer 2002]. Perhaps the most relevant to the token + constraint approach is Shneiderman's articulation of "direct manipulation" [1983]. While posed in the context of graphical interfaces, the direct manipulation concept is also directly applicable to tangible interfaces, arguably to an even greater than within GUIs. Shneiderman's direct manipulation 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 closely with the persistent nature of TUI tangibles. The second principle has special resonance with the token+constraint approach. Constraints serve as an embodiment of computational syntax, and transform physical actions within their perimeter (the constrained placement and manipulation of tokens) into the execution of computational operations. Constraints can also be seen to facilitate incremental and reversible operations; e.g., the placement of tokens is limited, and changes in computational context generally require the explicit movement of tokens to different constraints.

3.4 Models for tangible interfaces

3.4.1 MCRit

Several models have been proposed for tangible interfaces. Drawing from the MVC (model-view-control) model of GUI-based interaction, we have previously suggested an interaction model for tangible interfaces called MCRit[1], an abbreviation for "model-control-representation (intangible and tangible)" (Figure 9b) [Ullmer and Ishii 2001].

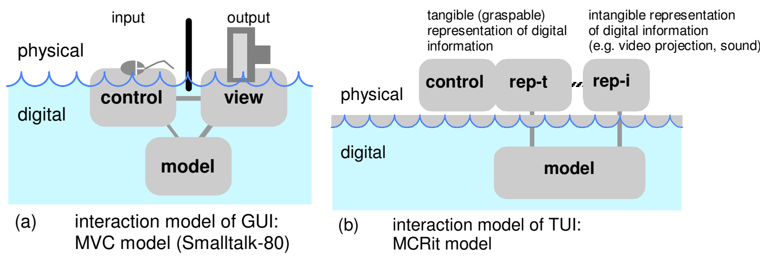

Figure 9: MVC and MCRit interaction models

MCRit highlights two conceptual aspects of tangible interfaces. First, the "view" concept from graphical interfaces is replaced by an interdependency between tangible representations (the interface's graspable, physically manipulable elements) and intangible representations (mediations such as dynamic graphics and sound). Second, TUIs utilize these physical representations as the interface's primary (and often sole) means for control, thus realizing a conceptual union in a key facet where graphical interfaces exhibit a fundamental divide.

We believe the MCRit model holds for token+constraint systems. The capacity for control can be seen as distributed between both tokens and constraints. For example, in the mediaBlocks system [Ullmer et al. 1998], mediaBlocks serve as both containers and controls (hence the "multiple inheritance" reference of §2.1). However, the specific nature of control is determined by the constraint within which the mediaBlock is placed. When placed within the "position rack" constraint, a mediaBlock serves as an "indexing" control for navigating its list of media contents. However, when placed within the "sequence rack" constraint, the mediaBlock expresses the logical sequence of its contents with respect to those of other mediaBlocks upon the rack. In this way, mediaBlock tokens and constraints contribute equally to the realization of the interface's "control" functionality. This will be discussed further in §4.1.

3.4.2 Terminology for styles of mapping vs. structural approaches

In another model, we have discussed TUIs within this paper and [Ullmer 2002] in terms of the "interactive surface," "token+constraint," and "constructive assembly" approaches. In previous writings, we have also described tangible interfaces in terms of "spatial,"

---

[1] Our original abbreviation for this model was "MCRpd" for "model, control, representation (physical and digital)." As discussed in [Ullmer 2002], we have revised the terms "physical and digital" to "tangible and intangible" for improved clarity.

"relational," and "constructive" mappings [Ullmer and Ishii 2001]. These terminologies are partially overlapping and worthy of clarification.

We see the "spatial," "relational," etc. terms as describing *styles of mapping* between the physical configuration of objects and the computational interpretations projected upon them. In contrast, Hornecker has noted that the "interactive surface" and "token + constraint" terms can be seen as describing broad *structural approaches* through which tangible interfaces are commonly embodied [personal communications, 2003].

There are frequently relationships between styles of mapping and structural approaches (Table 2). We believe the token+constraint approach has been the most common method for realizing relational mappings. However, the relationship between mappings and structural approaches is not one-to-one. Systems such as the Senseboard [Jacob et al. 2001] and Sensetable [Patten et al. 2001] have demonstrated relational mappings on interactive surfaces. AlgoBlocks [Suzuki and Kato 1993] and tangible programming bricks of [McNerney 2001] employ relational mappings within constructive assemblies. Also, later generations of the Urp urban planning system have used the token+constraint approach to express spatial mappings (e.g., the orientation of wind) [Ishii et al. 2002]. Just as graphical interfaces combine multiple styles of interaction (e.g., menus, spatial pointing, and command dialogs), we believe mature tangible interfaces may often employ multiple styles of mapping and structural approaches.

Style of mapping	Associated structural approach(es)
Spatial	Interactive surface, but also token+constraint
Relational	Token+constraint, but also interactive surface and constructive assembly
Constructive	Constructive assembly

Table 3: Styles of mapping and associated TUI architectures

3.4.3 Containers, tools, and tokens

In an influential model for tangible interfaces, Holmquist, Redström, and Ljungstrand suggested the terms "containers," "tools," and "tokens" as classifications for the roles served by physical/digital objects [1999]. While we see significant value in this classification, we have long used the "token" term in its more general sense, which is also consistent with the term's traditional meaning in computer science. More verbosely, Holmquist et al.'s "tokens" can be seen as iconic tokens with permanent bindings; "containers" are symbolic tokens with dynamic bindings; and "tools" are tokens that are bound to operations [Ullmer and Ishii 2001].

From the standpoint of this paper, it is useful to consider Holmquist et al.'s terminology in the context of token+constraint systems. Our "tokens" are most commonly used as "containers" (e.g., in the Marble Answering Machine [Polynor 1995], mediaBlocks [Ullmer et al. 1998], LogJam [Cohen et al. 1999], and Music Blocks [Neurosmith 1999]). However, the cartoon character objects of ToonTown [Singer et al. 1999] use iconic forms of physical representation, thus serving as "tokens" by Holmquist et al.'s terms. Similarly, several tiles of DataTiles [Rekimoto et al. 2001] serve as "tools." We suspect future systems will continue to see tokens serve a variety of roles.

We find Holmquist et al.'s categories to be valuable for compactly identifying some of the key functional roles that TUI "tangibles" serve in practice. Regarding the "dual use" of the "tokens" term, our earlier term "phicons" [Ishii and Ullmer 1997] might serve as a substitute label for iconic, statically bound tokens. Holmquist et al. noted our earlier description of mediaBlocks (symbolically, dynamically bound objects) as "phicons" in [Ullmer et al. 1998] as one rational for a substitute term. In retrospect, we agree that the "phicon" term is perhaps better limited to the description of iconic, statically bound tokens. Nonetheless, as we discuss in [Ullmer and Ishii 2001], a highly analogous debate over nuances of the GUI "icon" term continued for at least a decade. In practice, we suspect similarly diverse usage of terminology will continue to be common for TUIs.

Holmquist et al.'s terminology seems less suited to the characterization of constraints. Constraints could be considered as "tools," insofar as they are usually used to represent computational operations. However, constraints are also used as kinds of syntactic framing or structured workspaces that are not well-captured by the "tool" term. Holmquist et al. also propose the term "faucets" for locales where "tokens" can be accessed. For the present, we feel the "constraint" term is valuable in identifying the more specialized role served by these elements.

3.4.4 Factors and effects relating to cooperative uses

As observed in work such as [Cohen et al. 1999, Ishii et al. 2002, and Hornecker 2002], tangible interfaces' support for group communications appears to be one of their clearest and most compelling virtues. Hornecker has identified some of the enabling factors and positive effects relating to cooperative uses of tangible interfaces [2002]. These are summarized in Table 3. The token+constraint approach can be seen as having special implications for several of these, especially in comparison with interactive surfaces.

Enabling factors
constant visibility
bodily shared space
haptic direct manipulation
parallel access

Positive effects	
externalisation	active participation
intuitive use	gestural communication
awareness	provide focus
performative meaning of actions	

Table 4: Factors and effects for cooperative use of TUIs (adapted from [Hornecker 2002]). Facets with special ties to the token+constraint approach are shown in bold text.

For example, while most tangible interfaces make use of physical objects to represent digital information, interactive surface systems typically represent operations in dynamic, transient graphical form. In contrast, token+constraint interfaces typically use physical constraints as the embodiments of operations. Correspondingly, the passive haptic feedback, physical persistence, and other aspects of constraints can be argued to have positive consequences for group interactions. Specifically, in Hornecker's language, the constant visibility and haptic direct manipulation associated with constraints have benefits including externalization, intuitive use, awareness, and the performative meaning of actions. In fairness, as we will consider in §7.2, these advantages likely come at the expense of somewhat reduced flexibility and increased requirements for physical things.

3.5 Discussion

In this section we have presented some of the conceptual background underlying the token+constraint approach. With the abacus and board games, we find inspirations for the token+constraint approach, as well as examples of specific physical representations which might be thus employed. The abacus and board games also suggest possible "system genres" for token+constraint interfaces, as discussed by Bellotti et al [2002].[2]

In our discussion of external representations, distributed cognition, and affordances, we have attempted to situate the token+constraint approach within several specific subdisciplines of cognitive science. In addition to serving as general background material, we have attempted to highlight a number of issues from these areas with specific design implications for token+constraint systems. A number of other psychological subdisciplines are also of relevance, including diagrammatic representation [Larkin and Simon 1987; Petre 1995; Ullmer 2002] and motor psychology [Guiard 1987; Hinckley 1998]. Relevant ties from perspectives including semiotics and anthropology are considered in [Ullmer and Ishii 2001, 2002]. We also believe that numerous other

[2] System genres are "a set of design conventions anticipating particular usage contexts," such as media appliances or games [2002].

areas of study and practice, including product design, museum installation design, installation art, and sculpture, have specific relevance to the token+constraint approach.

Finally, we have considered several models and perspectives from the discipline of human-computer interaction. These include both classic instances such as direct manipulation, as well as a growing body of discussion specific to tangible interfaces.

4. EXAMPLE SYSTEMS

In the past pages, we have introduced the concept of token+constraint interfaces and considered some of their conceptual background. While the token+constraint concept is original to this paper (in parallel with [Ullmer 2002] and [Ullmer et al. 2003]), a number of past and recent interfaces employ the token+constraint approach.

In this section we briefly present and illustrate eleven such examples. Our interest is not in providing a literature survey, but instead in concretely illustrating ways the token+constraint approach has been employed in practice. We address this in part by describing the elements of each interface with the language introduced by this paper. Also, given the highly visual (and physical) nature of these interfaces, we accompany each description with figures illustrating their appearance and use. We hope this will be a resource for researchers who are developing new applications and variations of the token+constraint approach. We begin with two systems we have developed – media-Blocks and tangible query interfaces – and continue with systems by other researchers.

4.1 mediaBlocks

MediaBlocks is a system for physically capturing, retrieving, and manipulating digital media such as images and video [Ullmer et al. 1998]. MediaBlocks are small wooden blocks, which serve as tokens for the containment, transport, and control of online media. As with all of the other token+constraint examples we will present, these block-tokens do not actually store their "contents" internally. Instead, mediaBlocks are embedded with digital ID tags that allow them to function as "containers" for online content, while technically serving as a kind of physically embodied URL.

Figure 10a,b: mediaBlocks sequencer, printer slot

The mediaBlocks system was built around two types of devices, each making different uses of the token+constraint approach. First, "slots" – simple constraints supporting only the associate phase of interaction – were attached to or associated with a series of media input and output devices including a printer, wall display, overhead video camera, digital whiteboard, and a computer monitor (Figure 10b). These slots were each bound to either the "play" or "record" action for their associated device. On insertion of a mediaBlock into a slot, the system would store a media element "into" the block, or retrieve media "from" the block. Secondly, the central interface of the mediaBlocks system was the media sequencer (Figure 10a). This device integrated four different rack and "pad" constraints, each associated with different digital semantics. The sequencer supported the browsing and manipulation of media sequences.

4.2 Tangible query interfaces

The tangible query interfaces project developed several tangible interfaces for physically expressing and manipulating parameterized database queries [Ullmer 2002, Ullmer et al.

2003]. These interfaces use several kinds of physical tokens to 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. Examples of these interfaces are illustrated in Figure 11.

Figure 11a,b,c: Parameter wheels on query rack, in system overview; parameter bars on query rack

Figure 11a,b illustrates the “parameter wheel” approach for expressing queries. Here, round disks called “parameter wheels” are bound to database parameters, which can be placed within round “pad” constraints that are embedded within a “query rack.” Placement of these wheels within the query rack (the associate phase) expresses active parameters and the axes of data visualizations. Wheel rotation (the manipulate phase) allows physical manipulation of the wheels’ associated parameter values.

Figure 11c illustrates a second variation of the query interfaces employing “parameter bars.” These bars integrate active displays and mechanical levers that build upon the graphical “dynamic queries” technique of [Ahlberg and Shneiderman 1994]. The bar-tiles are again primarily used within a “query rack” constraint, although their embedded displays and controls also support uses outside of the query rack. Bar placement (the associate phase) again expresses active parameters. Manipulation of the sequence and adjacency of bars within the rack (the manipulate phase) drives the expression of 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.

4.3 Slot machine

Perhaps the earliest example of the token+constraint approach, and one of the earliest known tangible interfaces, is the Slot Machine of Perlman [1976]. It was co-developed along with a second closely-related interface, the “Button Box,” which is cited as one of the inspirations for the GUI “icon” concept [Smith 1975].

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 (tokens) were configured within horizontal slots (constraints) to construct Logo programs. Multiple card-tokens 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 12a). The Slot Machine provided a fairly sophisticated level of programmatic control, and supported concepts such as recursion that have not been repeated in other known tangible interfaces to date.

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, the slot machine makes strong use of the associate phase, but does not support a manipulate phase. Alternately stated, a card may enter or exit a slot, but no further physical manipulation of the card is supported once it is within the slot.

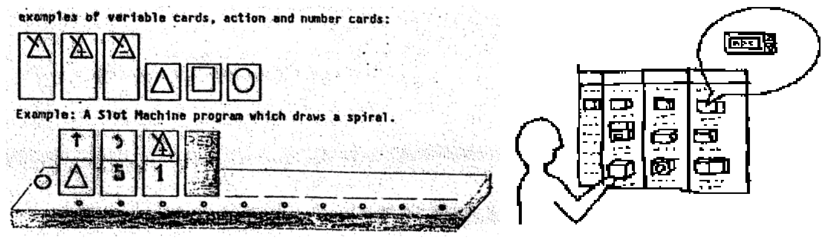

Figure 12a,b: Slot machine, recursive programming example [Perlman 1976]; LegoWall (described in [Fitzmaurice 1995])

4.4 LegoWall

Another "early" token+constraint system – perhaps the second-oldest known example, albeit nearly twenty years older than the slot machine – was the LegoWall interface of Molenbach (as described in [Fitzmaurice 1995]). The LegoWall system implemented a wall-based matrix of electronically sensed LEGO bricks that was employed for a ship scheduling application (Figure 12b). 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 associated information.

As illustrated in Figure 12b, the different port columns appear to have served as kinds of "constraints," with vertical movement of ship tokens within these constraints mapped to scheduling within time. The token+constraint mapping employed has no "manipulate" phase, and shares a similar "language" to other common uses of magnetic tokens upon whiteboards (e.g., for planning and scheduling).

4.5 Bricks "Tray" and "Inkwells"

Another relatively early use of the token+constraint approach was the "tray" and "inkwell" devices of Fitzmaurice, Ishii, and Buxton's Bricks system [1995]. Bricks was one of the earliest systems developing the "interactive surface" TUI approach. A central example of the broader "graspable user interface" approach, the Bricks system used the placement of one or more bricks – abstract, sensor-tracked physical blocks – onto various screen-based virtual objects, b-spline control points, etc. Bricks could then be used to physically rotate, translate, or (with multiple bricks) scale and deform the attached virtual entities by manipulating the proxying brick devices (Figure 13a).

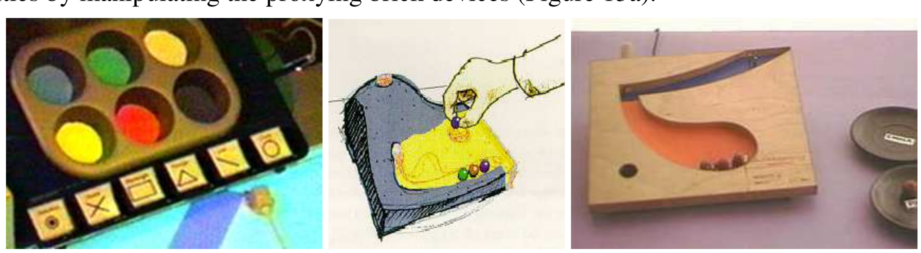

Figure 13a,b,c: Bricks – GraspDraw prototype and tray+inkwell close-up [Fitzmaurice et al. 1995]; Marble answering machine, animation and physical prototype [Polynor 1995, Abrams 1999]

The Bricks "GraspDraw" application used physical "tray" and "inkwell" devices (Figure 13a) to bind tools and attributes (colors) to Bricks. These bindings persist until Bricks are explicitly rebound. However, bindings are not active on the workbench unless a button upon the Brick is pressed; normal Brick behavior is as a handle for graphical objects. Fitzmaurice et al. did not elaborate upon the tray and inkwell devices; the above Brick behaviors were described as different styles of binding (transitory and persistent). The persistent bindings to the brick "token" approximate a kind of "container" functionality. The tray and inkwell each illustrate kinds of constraints, albeit without a "manipulate" phase of interaction.

4.6 Marble answering machine

Bishop's influential Marble Answering Machine concept sketch illustrated the use of physical marbles as containers and controls for manipulating voice messages [Polynor 1995] (Figure 13b,c). The marbles are moved between different depressions or "wells" to replay marble contents, redial a marble message's caller, or store the message for future reference. Bishop also developed a broader series of designs exploring the manipulation of physically-instantiated digital media, providing one of the earliest illustrations for interlinking systems of physical products through a shared physical/digital "language."

Bishop's designs illustrated a number of important functions that were further developed in the mediaBlocks system. These included the concept of physical objects as "containers" for digital media, and their use for transporting digital media between a family of multiple devices that share a common "constraint language." Bishop also made compelling use of "out-of-band" manipulations of physical / digital tokens, with marble-messages passively stored in labeled dishes and racks for reference by other answering machine recipients (Figure 13b). The marble answering machine and its accompanying devices support an associate phase of interaction, but no manipulate phase.

4.7 LogJam

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 LogJam and ToonTown were based upon the configuration of physical tokens upon a multi-tier rack (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 to and removed from the racks to annotate video footage by a group of video loggers (Figure 14a). 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.

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.

Figure 14a,b: LogJam system in use [Cohen et al. 1999]; ToonTown prototype with tokens [Singer et al. 1999]

For example, LogJam's users frequently made "out-of-band" configuration of their category blocks, organizing these blocks in front of them with individualized layouts and groupings. Users also 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 seemed to strongly distinguish its use from that of GUI alternatives.

4.8 ToonTown

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 14b). Manipulation of these tokens upon an array of racks allows the addition+removal of users; audio localization of users; assignment of users to tokens; and the display of information relating to participants.

The ToonTown system includes a number of interesting and provocative components. One of these is the physical representation of people, which we believe has powerful potential in future communication systems. Also, together with mediaBlocks, we believe ToonTown's mapping of linear position to left/right fade is one of the first published uses of the "manipulate" phase of token+constraint interaction.

4.9 Music Blocks

Another TUI for manipulating audio content is the "Music Blocks" system, one of the first tangible interfaces to be marketed commercially [Neurosmith 1999]. This system binds different musical fragments to the faces of physical cubes (tokens) (Figure 2d). Blocks can be sequenced within several constraint-receptacles, and new music mappings can be exchanged with desktop computers via a "Cyber Cartridge" memory module. The system supports an associative phase of interaction, but no manipulate phase.

4.10 Tagged handles

Likely the first token+constraint system to utilize force feedback is the "tagged handles" research of MacLean et al. [2000]. Here, RFID-tagged tokens represent digital contents such as video sequences, and mate with force feedback docks to provide haptic cues. These docks function as constraints, but mechanically constrain tokens "from within" (mating to cavities within the tokens), rather than constraining tokens' outside perimeters (Figure 15a). The haptic feedback introduced by tagged handles is an important development for the token+constraint approach, especially in eyes-busy contexts. These include systems where the eyes may be focused on separate graphical representations produced by token+constraint interfaces. MacLean et al. also make important theoretical contributions in discussing the combination of discrete and continuous modes of interaction, providing an earlier consideration for some of the analysis within this paper.

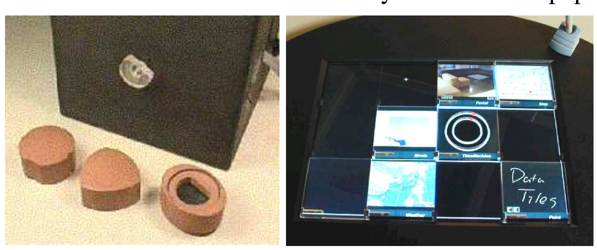

Figure 15a,b: Tagged handle concept (one example) and prototype [MacLean et al. 2000]; DataTiles system, combination of physical + digital elements [Rekimoto et al. 2001]

4.11 DataTiles

A final example related to the token+constraint approach is the DataTiles system of Rekimoto, Ullmer, and Oba [2001]. DataTiles used transparent plastic tiles (tokens) to represent modular software elements that could be composed on a graphically augmented 2D grid (constraint). These tiles were faced with partially transparent printed matter and pen-constraining grooves that allowed tiles to be persistently associated with different classes of information and functionality. Augmenting information and interactive manipulations were then mediated with dynamic computer graphics (Figure 15b).

DataTiles is a hybrid interface that integrates a number of tangible and graphical interface techniques. The system employs constraints in at least two different fashions. First, the workspace utilizes a two-dimensional array of pad constraints that limits the placement of tile-tokens to specific cells. Second, the grooves engraved into individual tiles are used to physically constrain the stylus, and in a sense also “constrain” dynamic graphical elements (e.g., selection points) that are mediated underneath these grooves.

DataTiles also heavily employs pen-based interaction with GUI applets displayed beneath the tiles. This hybrid approach draws strength from both physical and graphical interaction techniques, and seems a promising direction for continuing research.

4.12 Discussion

A number of observations can be made from these examples and the discussion of §2 and §3. First, a number of token+constraint systems have been developed and applied to a wide variety of applications. These systems have all relied upon a simple “language” employing a few recurring styles of constraints and tokens (Table 4 and Table 5).

Linear constraints (racks): mediaBlocks, tangible query interfaces, LogJam, ToonTown
Rotary constraints: Tangible query interfaces, tagged handles
Point constraints (pads, slots, wells): mediaBlocks, slot machine, LegoWall, Bricks tray, marble answering machine, ToonTown, Music Blocks, DataTiles

Table 5: Styles of constraints employed within example token+constraint systems

Table 4 summarizes the three basic styles of constraints that are used within the eleven example systems. These are the same basic constraints referenced in §2.1 and Figure 4. Figure 5 presented a summary of more complex combinations of tokens and constraints. All eleven example systems employed the movement of individual tokens between multiple constraints (Figure 5a). This associate phase can be seen as one the most fundamental “grammatical” compositions of token+constraint systems. Five examples employ the use of multiple tokens within a single constraint (Figure 5b) – mediaBlocks, the query interfaces, the slot machine, LogJam, and ToonTown. The query interfaces explored nested constraint relationships (Figure 5c), and this topic is the subject of ongoing work, but the use of nested relationships remains in an early stage.

Table 5 summarizes the four basic physical forms of tokens employed by the example systems. Each of these token forms is characterized by physical affordances that are mechanically complementary to their associated constraints. All of the tokens of the example systems are also of a size and mass affording manipulation with a precision hand posture (§3.2.2), with the exception of the query interfaces’ parameter bars and possibly LegoWall’s blocks, which are manipulated with a power posture.

Cubical or rectangular: mediaBlocks, tangible query interfaces, LegoWall, Bricks, LogJam, ToonTown, Music Blocks
Cylindrical: tangible query interfaces, tagged handles
Cards or tiles: Slot Machine, DataTiles
Physically representational: ToonTown

Table 6: Styles of tokens employed within example token+constraint systems

As discussed in §2.0 and summarized in Table 7, some token+constraint systems employ only the “associate” phase of interaction, while others employ both the associate and manipulate phases. This table indicates that the manipulate phase has emerged in relatively recent systems, beginning with the mediaBlocks and ToonTown. Finally, the example systems map constraints to several recurring functional interpretations. These are summarized in Table 8.

Only associate phase: Slot machine, LegoWall, Bricks tray, marble answering machine, music blocks, mediaBlocks (1/2)

Associate and manipulate phase: mediaBlocks (1/2), tangible query interfaces, ToonTown, Tagged Handles	
Hybrid approach: DataTiles (uses stylus-mediated manipulate phase)	

Table 7: Use of associate and manipulate phases within example token+constraint systems

Dynamic binding: mediaBlocks, Bricks tray, LogJam, ToonTown	
Manipulation of continuous parameter: mediaBlocks, tangible query interfaces, ToonTown, tagged handles, DataTiles	
Playback of digital media: mediaBlocks, marble answering machine, music blocks, DataTiles	
Storage and retrieval of digital state: mediaBlocks, DataTiles	

Table 8: Recurring functional interpretations of constraints in example token+constraint systems

A number of other observations and generalizations can be drawn from the example systems we have presented. Also, the example constraint behaviors we have identified in Table 7 are not exhaustive. Nonetheless, we believe the examples of this section should suggest generalizations and design patterns that are likely to hold for many future interfaces employing the token+constraint approach.

5. FIVE QUESTIONS FOR SENSING SYSTEMS

Bellotti et al. have recently proposed five questions for framing the discussion of sensing-based interaction, highlighted by the terms “address,” “attention,” “action,” “alignment,” and “accident” [Bellotti et al. 2002]. We believe that tangible interfaces in general, and token+constraint interfaces in particular, hold advantages for addressing these questions over sensing interfaces with more ambiguous methods for expressing engagement.

Specifically, tangible interfaces center around the explicit manipulation of special physical objects. This directed engagement with special artifacts expresses intentionality to engage with the system, thus clearly distinguishing people’s interactions with TUIs from that of other physical-world activities. In contrast, many other styles of sensing-based interaction are forced to contend with ambiguous distinctions between in-band interactions that should be interpreted and acted upon by the interface, and out-of-band interactions that should not be interpreted as “actionable” (e.g., coincidental movement in the proximity of the interface). Even humans sometimes have difficulty with such determinations, making this an especially difficult challenge for computational systems.

Nonetheless, we believe that considering token+constraint interfaces from the perspective of Bellotti et al.’s five questions is a valuable exercise. We frame our discussion within two broad perspectives: from a conceptual and perceptual standpoint, and in terms of the technological mechanisms through which these issues can be addressed.

5.1 Address

How does a system know the user is addressing it but not other systems?

5.1.1 *Conceptual and perceptual*

Constraints serve as well defined sensing zones that respond in clearly defined ways to the arrival, departure, presence, and absence of tokens within their perimeters. Constraint perimeters are clearly expressed through mechanically confining structures, visual markings, or both, reducing the potential for ambiguous configurations. When tokens are present within these perimeters, the system knows it is being addressed. If a mechanically enforcing constraint allows the movement of tokens, this movement offers another means for address. When no tokens are present within its constraints, the underlying system generally can assume it is not being addressed by its user(s).

5.1.2 *Technological*

Token+constraint systems detect that users are addressing them by sensing the presence, identity, and configuration of tokens within their constraints. The systems introduced in

§4 accomplish this through embedding tokens with some form of electronic tag, and embedding sensing electronics within the constraints. Such tags are discussed in more detail within [Want and Russell 2002] and [Ullmer 2002].

Of the examples in §4, six employ electrical contact between constraints and tags, while four employ wireless communications using RFID or light. Most of the systems using electrical contact suffered reliability problems; RFID and other wireless approaches seem preferable for future systems. Some systems from §4 tag objects with "analog" elements (e.g., with resistors of varying values), but most employ some form of digital ID, which generally brings improved reliability and scalability. Several interfaces also employ tag reader arrays, potentiometers, shaft encoders, etc. for sensing the configuration of tokens within constraints as another means of "address".

5.2 Attention

How does the user know the system is attending to her request?

5.2.1 Conceptual and perceptual

When tokens are placed within an interface's constraints, users expect the system to respond with some form of computational mediation. If a mechanically-enforcing constraint allows the movement of tokens, users generally expect mediation in response to this movement. This mediation should suggest whether the motion is interpreted or non-interpreted. If the motion is interpreted, the system should respond with additional mediation to indicate that this activity is being sensed and interpreted.

5.2.2 Technological

Token+constraint systems typically generate "events" corresponding to the entrance, exit, and motion of tokens with respect to constraints, which form the systems' internal representation of user "requests." These events generally should be accompanied by corresponding mediation. This mediation alerts the user that the system has sensed user activity, indicates the nature of event that was sensed, and provides computational products back to the user. The ten systems of §4 use diverse forms of mediation to let users know the system is attending to their requests. To illustrate the variety of mediation employed, we summarize the classes of technologies used by the systems of §4:

Visual mediation:

- Embedded high-resolution flat panel displays (mediaBlocks, DataTiles)
- Embedded low-resolution LCD displays (LegoWall, query interfaces)
- "Single-pixel" LED displays (Slot machine, LegoWall, query interfaces)
- High resolution projector (Query interfaces)
- Traditional desktop display screen (LogJam)

Sonic mediation:

- Audio-only systems (Marble answering machine, ToonTown, Music Blocks)
- Audio-augmented systems (mediaBlocks, Log Jam, query interfaces)

Mechanically actuated mediation:

- Physical motion (Slot machine, tagged handles)
- Force feedback (tagged handles)

5.3 Action

How does the system know what object the user's command (e.g., save) relates?

5.3.1 Conceptual and perceptual

In most systems within §4, tokens represent elements or aggregates of data, and constraints represent operations that may be applied to this data. In this fashion, users may express both the action itself and the object of this action through physically composing different combinations of tokens and constraints. For example, in Bellotti et al.'s "save" example, a constraint might represent the "save" operation, with a token representing the container into which content is to be saved. (This particular example was illustrated by the mediaBlocks system.) The data to be saved might have been invoked by another token+constraint ensemble within the interface – e.g., a token containing source data, placed within a constraint bound to a "show" operation.

In several systems, tokens have represented both data and operations, with constraints used more as a compositional tool. For example, in the slot machine, data and operations are both represented with card-tokens of different heights. These are grouped together in single slots to express both the "subject" and "verb" of a command. A row of multiple slots represents the ordered sequence of a chain of commands. The DataTiles system also represents both data and operations as tiles. Here, the "subject" and "verb" are combined by placing them in adjacent cells within the grid of the DataTiles workspace.

5.3.2 Technological

Most commonly, token+constraint systems technologically "know" the mapping between physical tokens and their corresponding digital information through tags embedded within tokens. Often, these tags are encoded with a unique digital serial number, somewhat resembling a credit card number or the library catalog number of a book. This digital ID can then be mapped to corresponding digital information through some form of database, with the ID serving as a key.

In cases where a unique digital ID is not present – e.g., with the use of resistors as forms of analog ID – systems generally attempt to resolve some form of digital ID through whatever sensing modality they employ, and then proceed in a similar fashion. Constraints are frequently physically fixed within token+constraint systems, making their "identification" a relatively straightforward process. However, constraints themselves are sometimes physically reconfigurable. Especially in these cases, constraints may also be embedded with ID tags.

5.4 Alignment

How does the user know the system understands and is correctly executing users' goals?

5.4.1 Conceptual and perceptual

As with Bellotti et al.'s second question ("attention"), the process of alignment is closely linked to the system's approach for mediating responses to user interaction. In some token + constraint systems, the concepts and mechanisms for expressing attention and alignment are very similar. For example, with the mediaBlocks sequencer and the DataTiles workspace, the graphical mediations used for expressing attention and alignment are roughly colocated. In mediaBlocks, the consequences of physical manipulations are mediated from a graphical surface adjacent to the constraint workspaces; while in DataTiles, the mediation is displayed directly underneath the manipulated tiles.

In other systems, there is a gap between the mediations expressing attention and alignment. For example, in the parameter wheels prototype of tangible query interfaces, the identity and values of parameters are projected contiguous to the parameter tokens, but the actual query result is displayed on a separate display surface. It could be argued that alignment is born out by the mediations adjacent the parameter tokens. Nonetheless, there remains a gap between the locus of user interaction and the locale where the consequence of these interactions are ultimately displayed. For example, with the mediaBlocks sequencer, we have discussed the struggle to integrate graphical mediations with the system's physical elements in [Ullmer et al. 1998].

Approaches for tightly integrating "control" and "display" aspects of interaction are a common and consequential challenge for tangible interfaces in general, and the token + constraint approach in particular. This issue seems partly a function of the application domain, and partly a product of design. The integration of physical and graphical spaces is clearly easier in domains that offer intrinsic geometrical mappings, but this is generally not the case for the kinds of information token+constraint interfaces are used to represent.

5.4.2 *Technological*

The mechanisms for mediating a sense of alignment are similar to those for communicating attention, and we have discussed in §6.2.2. Given the potential for a perceptual gap between tokens and associated graphical mediations, audio and mechanical feedback channels can also play a strong role for expressing alignment, even in systems that rely primarily on graphical mediation. Audio has been used for feedback by the mediaBlocks system, likely among others. Similarly, physical movement and force feedback have been used in the tagged handles work of MacLean et al. [2000]. More recent work such as the Actuated Workbench of Pangaro et al. [2002] also has strong potential for combination with token+constraint interfaces.

5.5 Accident

How do the user and the system resolve misunderstandings?

5.5.1 *Conceptual and perceptual*

Token+constraint systems discourage erroneous combinations of tokens and constraints through the kinds of mechanical complementarities and compatibilities between tokens and constraints discussed in §2.1 and §6.1.1. However, these compatibilities express syntactic, not semantic, relationships. Per the quote of Ten Hagen in §2.1, "[syntax] will allow many [digital expressions] that don't make sense" [1981]. In these cases, expression of the erroneous combination is left to computational mediation.

In actual practice, as Bellotti et al. have noted for sensor-based interfaces at large [2002], insufficient work has been done regarding error expression and resolution in token+constraint systems. As with the "Listen Reader" example cited by Bellotti et al., some token+constraint systems are sufficiently simple that "error" conditions can be "assumed away." In other examples from §4, many prototype systems have not developed to the point where error conditions are cleanly expressed and resolved.

Token+constraint systems have often mediated error conditions with visual or audio feedback. However, with the increasing development of actuation technologies (e.g., [MacLean et al. 2000; Pangaro et al. 2002]), new paths are being opened for tangible interfaces to respond to erroneous or ambiguous configurations. Moreover, while prototypes such as [Pangaro et al. 2002] support continuous actuation on a 2D workspace, these technologies can be especially well-suited for token+constraint systems. Among other reasons, this is because mechanical constraints can enable actuation with many fewer active elements, leading to more economical prospects for applied use.

5.5.2 *Technological*

From a sensing perspective, technological "misunderstandings" can be reduced by employing robust technologies. Wireless sensing approaches – especially RFID – often performs well in this respect. However, even relatively robust techniques like RFID have numerous failure modes. For example, many RFID readers are unable to sense multiple tags within a given read-volume (i.e., they lack "anti-collision" technology). In such systems, the presence of multiple colocated tags may lead either to an error condition or (perhaps worse) to an absence of detected tags. If the error condition can be sensed, mediations can be used to communicate this to users. Otherwise, the error hopefully can be detected by users through the absence of corresponding mediations.

6. DISCUSSION

A major goal of this paper is to support the token+constraint approach as a viable and promising style of sensing-based interaction worthy of more widespread research, development, and deployment. Toward this, we see several paths forward. Building on the themes and examples identified within this paper, a first path might be to refine and distill these techniques; to employ them as primary and supplementary interfaces within both new and existing systems; and to deploy these systems into use with real users.

Aside from Music Blocks and perhaps DataTiles, we suspect that none of the token + constraint systems we have discussed has reached a level of maturity (especially robustness) that supports serious use. This partly reflects the research challenges of simultaneous developments in electronics, mechanics, product design, and software, and has limited both the evaluation of existing systems and the proliferation of new systems.

Nonetheless, we are convinced that these challenges are increasingly manageable by both small teams and individuals. Building on advances in RFID, embedded computing, networking, and rapid prototyping technologies, we believe the token+constraint approach is amenable to robust, inexpensive, widespread deployment. A number of hardware/software toolkits have begun to appear to support such efforts; e.g., [Ballagas et al. 2003, Gellersen et al. 2002, Klemmer 2003]. In a related path, Calvillo-Gámez et al. have proposed the "TAC" paradigm as a generalization of the token+constraint concept [2003]. Among other goals, TAC seeks to provide a set of abstractions that can serve as the basis for software toolkits.

Perhaps as with early comparisons between GUIs and character-based interfaces, we believe the strength of token+constraint interfaces lies not in quantitative performance, but with qualitative factors, especially regarding colocated collaboration. However, to the extent this is true, confirmation of these factors is unlikely to fully emerge until robust systems are deployed in real usage contexts.

Another possible path forward is to consider variations on the token+constraint approach that expose new design spaces. We consider several such variations in the next section. In the final section, we discuss some of the limitations of the token+constraint approach, as well as prospects that might mitigate and potentially transform these issues.

6.1 Variations on token+constraint approach

This paper has described tokens and constraints as exhibiting the following properties:

tokens: physically embodied, discrete, rigid elements, each representing data

constraints: physically embodied, mechanically confining, rigid elements; each representing operations, and each allowing token movement with one or zero continuous degrees of freedom.

We believe these properties are an accurate reflection of the token+constraint systems that have been developed to date, and that this combination brings about a number of benefits (discussed in §2.3). However, a number of possibilities are exposed by relaxing or reversing these attributes.

6.1.1 Visual and graphical constraints for physical elements

This paper has focused upon constraints with "hard" (mechanically confining) perimeters. However, constraints with "soft" perimeters are also possible. These may be expressed in static visual form, as with the printed cells found in many board games (e.g., the square "property" cells ringing the perimeter of the Monopoly™ board). They may also be expressed in dynamic graphical form, especially in the context of TUIs employing interactive surfaces. This approach has seen early development within the Sensetable and Audio Pad systems of Patten et al. [2001, 2002].

Removing the mechanically confining perimeter of constraints sacrifices some of the benefits discussed in §2.3. Nonetheless, "soft" constraints may still employ many aspects of the token+constraint approach, and also offer other benefits. For example, passive visual constraints may be realized at reduced cost, with precedent in the different mechanical forms of some "economy" vs. "deluxe" board games (e.g., Scrabble™).

When realized in graphical form upon interactive surface systems, constraints can also draw upon the malleability and other benefits of graphical interfaces.

6.1.2 Physical constraints for graphical elements

Conversely, mechanical constraints may be used to "confine" graphical elements. Here, graphical "tokens" might be manipulated with the finger, a stylus, or other physical tools, with the mechanical constraint serving as a kind of "jig" for providing passive haptic feedback. The DataTiles system's stylus+constraint interaction illustrates one such use [Rekimoto et al. 2001]. As with DataTiles, such variations might yield benefits including passive haptic feedback and new interaction paradigms for stylus-based systems.

6.1.3 Physical constraints for non-discrete physical materials

In another variation, one can imagine using physical constraints in conjunction with more "continuous" physical mediums such as liquids, granular materials (e.g., sand), and "phase change" materials (e.g., ice). For example, we have considered heated pad "constraints" into which fluids "embodying" various media might be poured. Poured contents might activate playback; when the fluid evaporates, playback might cease. Several related ideas have been developed in [Mazalek and Jehan 2000].

6.1.4 Tokens and constraints of varying size

Tokens and constraints might also have adjustable size. For instance, Fitzmaurice et al. experimented with a "stretchable square" [Fitzmaurice et al. 1995], and the metaDESK employed a "scaling-constraint instrument" [Ullmer and Ishii 1997]. These and similar objects might be candidates for resizable tokens. Additionally, variations on the stretchable square might be employed as a resizable constraint (e.g., a resizable rack). Resizable constraints might be useful for sharing physical real estate between multiple differently-purposed racks; for loosening, relaxing, or "unlocking" certain constraint relationships; or for other styles of constraint morphing. Resizable constraints might also have special potential in combination with actuation and/or force feedback.

6.1.5 Alternative digital associations and semantics

The above examples have illustrated alternate physical and graphical representations which might be employed within token+constraint interfaces and their descendants. In addition, it is possible to develop major variations in digital semantics. For example, while we have described tokens as usually representing digital information or "operands," the DataTile system offers clear examples of tile-tokens that also serve as operators or controls (e.g., the "time machine" and "parameter" tiles). The 2D array of tile pads within the DataTiles workspace also illustrates "constraints" that do not represent "operations" per se, so much as a structured workspace for tile composition. We believe the DataTiles system is a hybrid of several different interaction approaches (including "constructive assemblies" as well as graphical pen-based systems) which defies simple classification. In practice, we expect many mature interfaces will employ combinations of different interaction approaches. We will elaborate upon this in the next section.

6.2 Limitations of token+constraint approach

Where this paper has concentrated on the potential strengths of token+constraint interfaces, it is also important to consider some of their limitations. Perhaps most obviously, for applications requiring spatial interaction with geometrical content, the constraint interfaces we have discussed do not support the continuous two-dimensional positioning common to graphical interfaces and interactive surface TUIs, or the higher-dimensional positioning afforded by other sensing approaches. We believe that token+constraint systems can strongly complement systems providing more freeform input, but they clearly do not fully substitute for these systems.

More broadly viewed, by the very act of imposing physical structure onto the interface, token+constraint systems are more limited in "malleability" than traditional GUIs and TUIs that employ interactive surfaces. While we have argued how this can sometimes be a strength, in other cases it clearly presents limitations. At the same time, the modular, recomposable, and (sometimes) open-ended aspect of many token+constraint interfaces can act as a counterbalancing force. Also, the DataTiles system illustrates the potential for combining token+constraint interfaces with graphical interface techniques, yielding a significant increase in interface malleability.

6.2.1 Stand-alone vs. integrated uses

We suspect that mature tangible interfaces will often employ combinations of different interaction styles (e.g., combining both interactive surface and token+constraint approaches). Moreover, we suspect that in the long term, the token+constraint approach may be used more frequently in conjunction with other user interface techniques – including virtual reality, augmented reality, presentation, and conferencing systems – than as a fully independent technique. By way of analogy, some early GUI applications channeled the bulk of user interaction through menus. Today, most GUIs utilize menus, but few depend upon menus as the sole modality of interaction. While we hope the range of applications for token+constraints may be broader than that of menus, a related analogy may apply.

As specific examples of possible integrations, tangible query interfaces might be combined with systems such as TouchCounters [Yarin and Ishii 1999] or the Urp urban planning simulator [Underkoffler and Ishii 1999]. The TouchCounters system used arrays of display-augmented containers to provide a "distributed visualization" for the containers' usage history [Yarin and Ishii 1999]. Tangible query interfaces could provide a strong approach for querying such a system, with results displayed directly onto the containers. In another variation, tangible query interfaces could query (e.g.) census information within Urp, with query results integrated directly within Urp's graphical work surface. As a further example, mediaBlocks offer a general means for saving and retrieving digital state. This functionality could hold value in combination with many TUIs, VR systems, and other interfaces, which frequently lack such support.

In these examples, token+constraint elements might serve as kinds of TUI "widgets" integrated within more complex interfaces. These examples also suggest paths for improving the integration of control and display, another shortcoming of some token + constraint interfaces.

6.2.2 Scalability

As noted in [Ullmer et al. 2001], some variation on the "Deutsch limit" (suggesting the implausibility of more than 50 visual primitives in simultaneous use within screen-based visual languages) may apply to individual token+constraint systems, as with other tangible interfaces. At the same time, the style of mechanical structuring provided by token + constraint interfaces may help manage physical clutter and structure interaction better than more freeform interactive surface systems. Also, we believe the combination of dynamic binding and new display technologies (e.g., "electronic ink") will have major implications for the scalability of token+constraint systems, in that token's visual labelings might be quickly alterable to reflect evolving digital associations.

6.2.3 User feedback and testing

A relatively small percentage of the systems discussed in §4 have reported real user testing. Of these, two systems – Logjam and tangible query interfaces – preliminarily tested hypotheses of improved performance with respect to graphical interfaces, and neither was able to confirm this hypothesis [Cohen et al. 1999; Ullmer 2002].

On the other hand, both of these systems reported strong positive user feedback. More concretely, one example of token+constraint systems (Music Blocks, §4.9) has been a critical and commercial success, while another (the Marble Answering Machine, §4.6) has been held up as a highly successful illustration of interaction design [Preece et al. 2002].

While we believe there are likely tasks in which token+constraint systems hold quantitative performance advantages, we believe these are not the primary benefits of token+constraint systems. Rather, we believe their benefits are more qualitative in nature, as elaborated in §2.3. Many of these benefits are likely to be in the service of interpersonal communications and colocated collaboration. We suspect these may be better evaluated through experiences with real-world use rather than controlled experiments. Here, we share some of the goals of Aish and Frazer, who developed some of the earliest tangible interfaces for architectural uses beginning in the late 1970s [Aish and Noakes 1984, Frazer 1982]. Aish was optimistic that physical/digital tools might help lay people and designers to communicate, negotiate, and explore alternatives in face-to-face contexts [1984]. We share this optimism, and believe that token+constraint systems extend these prospects to interaction with abstract digital information.

It is also worth mentioning that twenty years ago, several studies tried to demonstrate that graphical interfaces were "faster" than text-based interfaces, but found this hypothesis difficult to confirm (e.g., [Jones and Dumais 1986]). While text-based interfaces remain in widespread use, many people prefer GUIs. We believe that token + constraint systems may also come to hold broad relevance, including for tasks where quantitative performance gains are not confirmed.

7. CONCLUSION

This paper has worked to articulate and explore the implications of a relatively simple set of ideas. We have identified a subset of tangible interfaces that center around relationships between two kinds of physical objects: tokens and constraints. We have discussed and illustrated ways by which tokens can represent digital information (or "operands"), and constraints can represent computational operations. Finally, we have shown how a simple set of physical and digital relationships between these elements can be used to express and manipulate a wide variety of open-ended computational expressions.

While the articulation of this approach is original to this paper, it is supported by interfaces spanning nearly three decades. Further, the approach draws on techniques for physically representing and manipulating abstractions that are as old as civilization itself. Nonetheless, recent years have brought a combination of needs and enabling technologies that give the token+constraint approach heightened relevance and promise.

The proliferation of computer technology into new physical and social contexts is creating demands for interface techniques that are compatible with eyes-busy and fractional-attention use; that foster and facilitate colocated collaboration; and that fit into diverse usage contexts within the home and workplace. In parallel, many of the interface examples we have considered depend upon technologies that have recently made great progress in cost and pervasiveness – especially RFID tagging, embedded computing, and embedded networking. We believe these demands and opportunities bode well for broadening use of the token+constraint approach.

As we have discussed, token+constraint interfaces seem suitable both for stand-alone use, and as elements of more complex systems. Especially when used as interfaces to simple devices, the token+constraint approach can stand by itself, as demonstrated perhaps most compellingly by Bishop's marble answering machine. We also believe the token+constraint approach has equal promise for use as elements of more complex interfaces. These include combinations with TUIs employing interactive surfaces and constructive assemblies; with virtual and augmented reality systems; with physical devices and appliances; and even with traditional graphical interfaces. An early illustration was provided by the Bricks "tray" and "inkwells" (§4.5), and more sophisticated uses have been developed within new systems that have yet to be published.

Our hope is that the token+constraint approach can play a role in shaping 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]. We believe that the marble answering machine, mediaBlocks, ToonTown, and DataTiles each speak to ways that token+constraint 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, understanding, and discussing the implications of these advances hold 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]. We hope that token+constraint systems such as mediaBlocks, DataTiles, and tangible query interfaces can play a positive role in supporting these critical dialogues and decisions.

8. ACKNOWLEDGEMENTS

We would like to thank Miriam Konkel, Eva Hornecker, Lars Erik Holmquist, Johan Redström and the anonymous reviewers for valuable feedback and suggestions on the paper draft. We also thank John Maeda, James Patten, Dan Maynes-Aminzade, and Gian Pangaro for discussions of many of the ideas in this paper. The research underlying this paper was conducted as Ph.D. work within the MIT Media Laboratory. The research was supported in part by IBM, Steelcase, Intel, and other sponsors of the MIT Media Lab's Things That Think and Digital Life consortiums. The paper was also supported by Hans-Christian Hege (Zuse Institute Berlin/ZIB) and the EC GridLab project, IST-2001-32133.

9. REFERENCES

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.

ANAGNOSTOU, G., DEWEY, D., AND PATERA, A. 1989. Geometry-defining processors for engineering design and analysis. In *The Visual Computer*, 5:304-315, 1989.

ANDERSON, D., FRANKEL, J., MARKS, J., et al. 2000. Tangible Interaction + Graphical Interpretation: A New Approach to 3D Modelling. In *Computer Graphics Proceedings* (SIGGRAPH'00), pp. 393-402.

BALLAGAS, R., RINGEL, M., STONE, M., AND BORCHERS, J. 2003. iStuff: a physical user interface toolkit for ubiquitous computing environments. In *Proceedings of CHI'03*, pp. 537-544.

BELL, R. 1979. *Board and Table Games from Many Civilizations*. New York: Dover Publications.

BELLOTTI, V., BACK, M., EDWARDS, et al. 2002. Making Sense of Sensing Systems: Five Questions for Designers and Researchers. In *Proceedings. of CHI'02,* pp. 415-422.

CALVILLO-GÁMEZ, E., LELAND, N., SHAER, O., AND JACOB, R. 2003. The TAC paradigm: unified conceptual framework to represent Tangible User Interfaces. In *Proceedings of LAHCI*, pp. 9-15.

PIERNOT, P. 1999. Logjam: A Tangible Multi-Person Interface for Video Logging. In *Proceedings of CHI'99*, pp. 128-135.

COHEN, J., WITHGOTT, M., PIERNOT, P. 1999. Logjam: A Tangible Multi-Person Interface for Video Logging. In *Proceedings of CHI'99*, pp. 128-135.

CUTKOSKY, M., AND HOWE, R. 1990. Human Grasp Choice and Robotic Grasp Analysis. In *Dextrous Robot Hands,* pp. 5-31, Springer Verlag.

DEAI Mini Encyclopedia). 2002. Genkan: Entrance. http://www.tjf.or.jp/deai/contents/teacher/mini_en/-html/genkan.html

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.

FERNANDES, L. 2001. The abacus: the art of calculating with beads. http://www.ee.ryerson.ca:8080/-~elf/abacus/ [Accessed 10 February 2002].

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., 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*, v.1371, Wachsmut and Fröhlich, eds. Berlin: Springer-Verlag, pp. 297-308.

FRAZER, J. 1994. *An Evolutionary Architecture*. Architectural Association: London, 1994.

GELLERSEN, H., SCHMIDT, A., AND BEIGL, M. 2002. Multi-sensor context-awareness in mobile devices and smart artifacts. In *Mobile Networks and Application,* v7n5, pp. 341-351.

GIBSON, J. 1979. *The Ecological Approach to Visual Perception*. New York: Erlbaum Associates, 1979.

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.

HINCKLEY, K., PAUSCH, R., PROFFITT, D., AND KASSELL, N. 1998. Two-Handed Virtual Manipulation. In *ACM Transactions on Computer-Human Interactions,* pp. 260-302.

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.

HORNECKER, E. 2002. Understanding the Benefits of Graspable Interfaces for Cooperative Use. In *Proceedings of Coop'2002*, pp. 71-87

IFRAH, G. 2001. *The Universal History of Computing: From the Abacus to the Quantum Computer*. New York: Wiley & Sons.

ISHII, H., UNDERKOFFLER, J., CHAK, D., PIPER, B., BEN-JOSEPH, E., YEUNG, L., KANJI, Z. 2002. Augmented Urban Planning Workbench: Overlaying Drawings, Physical Models and Digital Simulation. In *Proceedings of the International Symposium on Mixed and Augmented Reality* (ISMAR '02), pp. 203-214.

ISHII, H., AND ULLMER, B. 1997. Tangible Bits: Towards Seamless Interfaces between People, Bits, and Atoms. In *Proceedings of CHI'97*, pp. 234-241.

JACOB, R., ISHII, H., PANGARO, G., AND PATTEN, J. 2002. A Tangible Interface for Organizing Information Using a Grid. In *Proceedings of CHI 02*, pp. 339-346.

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.

KIRSH, D. 1995. The Intelligent Use of Space. In *Artificial Intelligence*, 1995.

KLEMMER, S. 2003. Papier-Mâché: Toolkit support for tangible interaction. In *Proceeidngs of UIST'03*, 1995.

LARKIN, J., AND SIMON, H. 1987. Why a Diagram is (Sometimes) Worth Ten Thousand Words. In *Cognitive Science*, v11, 1987, pp. 65-99.

LÜTJENS, J. 2002. Abacus Online Museum. http://www.joernluetjens.de/sammlungen/abakus/abakus-en.htm [Accessed 10 February 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.

MASTERS, J. 2002. The Royal Game of Ur and Tau. http://www.tradgames.org.uk/games/Royal-Game-Ur.htm [Visited August 5, 2002]

MAZALEK, A., AND JEHAN, T. 2000. Interacting with Music in a Social Setting. In *Extended Abstracts of CHI'00*, pp. 255-256.

MCNERNEY, T. 2000. Tangible Programming Bricks: An Approach to Making Programming Accessible to Everyone. MS Thesis, MIT Media Laboratory.

NELSON, L., ICHIMURA, S., PEDERSON, 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. 1999. Affordances, Conventions, and Design. In *Interactions*, v6 n3, pp. 38-43.

NORMAN, D. 1993. *Things that make us smart*. Reading, MA: Addison-Wesley, 1993.

OXFORD ENGLISH DICTIONARY (OED). 1989. "syntax." OED Online Oxford University Press.

PANGARO, G., MAYNES-AMINZADE, D., ISHII, H. 2002. The Actuated Workbench: Computer-Controlled Actuation in Tabletop Tangible Interfaces. In *Proceedings of UIST '02*, pp. 181–190.

PATTEN, J., RECHT, B., AND ISHII, H. 2002. AudioPad: A Tag-Based Interface for Musical Performance. In *Proceedings of the Int. 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.

PERLMAN, R. 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.

POLYNOR, R. 1995. The Hand That Rocks the Cradle. *I.D.*, May/June 1995, pp. 60-65.

PREECE, J., ROGERS, Y, AND SHARP, H. 2002. *Interaction Design*. New York: Wiley, 2002.

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.

RETZ-SCHMIDT, G. 1988. Various Views on Spatial Prepositions. In *AI Magazine*, 9(2), pp. 95-105.

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.

SCHIESSL, S. 2002. Digital Cubes. http://www.aec.at/festival2002/texte/schiessl_e.asp.

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.

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.

SMITH, D. 1975. Pygmalion: A Creative Programming Environment. Ph.D. Thesis, Stanford University.

SUZUKI, H., AND KATO, H. 1993. AlgoBlock: a Tangible Programming Language, a Tool for Collaborative Learning. In *Proceedings of 4th European Logo Conference*, pp. 297-303.

TEN HAGEN, P. 1981. Interaction and Syntax. In *International Journal of Man-Machine Studies*, v15.

TOMOE SOROBAN CO., LTD. 2002. Soroban museum: Roman Soroban. http://www.soroban.com/museum/-5s_eng.html [Accessed 10 February 2002].

ULLMER, B., ISHII, H., AND JACOB, R. 2003. Tangible Query Interfaces: Physically Constrained Tokens for Manipulating Database Queries. To appear in *Proceedings of INTERACT'03*.

ULLMER, B. 2002. Tangible Interfaces for Manipulating Aggregates of Digital Information. Ph.D. dissertation, MIT Media Laboratory, 2002.

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., AND ISHII, H. 1997. The metaDESK: Models and Prototypes for Tangible User Interfaces. In *Proceedings of UIST'97*, pp. 223-232.

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.

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.

WANT, R., AND RUSSELL, D. 2000. Ubiquitous Electronic Tagging. In *IEEE Distributed Systems Online*, September 2000, Vol. 1, No.2.

YARIN, P., AND ISHII, H. 1999. TouchCounters: Designing Interactive Electronic Labels for Physical Containers. In *Proceedings of CHI'99*, pp. 362-368.

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.

Received February 2003; revised August 2003; accepted February 2004.