DataTiles: A Modular Platform for Mixed Physical and Graphical Interactions

Jun Rekimoto **Brygg Ullmer*** **Haruo Oba**

Interaction Laboratory
Sony Computer Science Laboratories, Inc.
3-14-13 Higashigotanda
Shinagawa-ku, Tokyo 141-0022 JAPAN
Phone: +81 3 5448 4380
Fax +81 3 5448 4273
{rekimoto,oba}@csl.sony.co.jp
http://www.csl.sony.co.jp/person/rekimoto.html

* Tangible Media Group
MIT Media Laboratory
20 Ames Street
Cambridge, MA 02139 USA
ullmer@media.mit.edu

ABSTRACT
The DataTiles system integrates the benefits of two major interaction paradigms: graphical and physical user interfaces. Tagged transparent tiles are used as modular construction units. These tiles are augmented by dynamic graphical information when they are placed on a sensor-enhanced flat panel display. They can be used independently or can be combined into more complex configurations, similar to the way language can express complex concepts through a sequence of simple words. In this paper, we discuss our design principles for mixing physical and graphical interface techniques, and describe the system architecture and example applications of the DataTiles system.

Keywords
Interaction techniques, tangible user interfaces, graphical user interfaces, visual language, radio-frequency identification tags

INTRODUCTION
In recent years, there have been many efforts towards designing user interfaces incorporating specialized physical objects, as alternatives to onscreen graphics and general-purpose input devices such as the mouse [21, 16, 3, 7]. We can expect several potential advantages from this approach. Physical objects offer stronger affordances than purely visual ones. People can use their sophisticated skills for manipulating objects – not just pointing and clicking, but also rotating, grasping, attaching, etc. Interactions may also involve two hands, or allow many people to interact cooperatively in the same physical interaction space. Also, unlike the graphical objects of most GUIs, physical objects do not suddenly disappear or reappear when the system changes modes. This potentially leaves users with more confidence and a stronger sense of control over the status of the interaction.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGCHI'01, March 31-April 4, 2001, Seattle, WA, USA.
Copyright 2001 ACM 1-58113-327-8/01/0003...$5.00.

However, most prior work in this area has pursued special purpose systems that do not scale to support many different applications, as is common with graphical user interfaces. This may reflect the fundamental nature (or perhaps the limitations) of these systems. In many cases, physical objects may have less flexible behavior than their graphical counterparts, and objects designed for one task may be difficult to repurpose. This increased potential for task specificity can support a simplified interface, and is probably desirable for particular application domains (e.g., digitally-enhanced toys). However, we believe this scalability challenge is one reason these systems have not yet seriously competed with mainstream graphical user interfaces. Our goal in this paper is to design a system that utilizes the strengths of both graphical and physical user interfaces.

Another motivation behind this work is the increasing complexity of orchestrating digital devices. Computers are shifting from the PC era to a new generation that is often called *ubiquitous computing* [20]. In this new generation of computing, it is suggested that users will be able to focus more on the task itself rather than on the underlying computer, and that an abundance of task-specific devices ("information appliances") will be major interfaces to the digital world. Moreover, in theory, these devices should communicate with each other to support our daily lives.

However, in practice, the increasing complexity of many digital devices makes our life complicated. For example, if a person wishes to send a snapshot of a TV screen in his living room to a friend by email, the necessary operations are currently rather complicated. Even if each individual device has a simple, well-implemented user interface, these kinds of inter-appliance operations are still potentially quite complex. Our design goal is to facilitate this kind of usage context by providing a sort of "universal control" interface that integrates the function of many different digital appliances.

This paper addresses these issues by introducing the use of

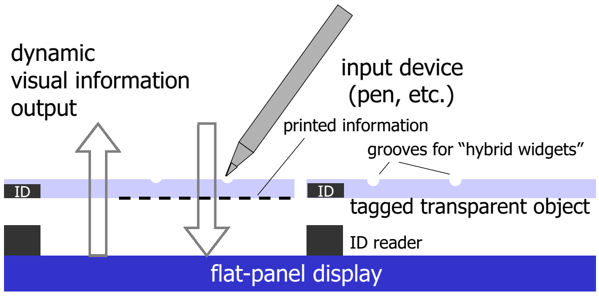

Figure 1: Concept of graphically augmented physical objects. It offers tight coupling of input and output through tagged, transparent objects.

Figure 2: DataTiles system.

graphically augmented physical objects (Figure 1). Our prototype system, DataTiles (Figure 2), uses tagged transparent tiles as a modular unit of interaction. By combining a sensor-enhanced flat panel display with these tiles, the system enables users to manipulate digital data as physical DataTiles, while retaining the flexibility and power of graphical user interfaces.

This system integrates three significant interaction ideas:

1. *Tagged transparent objects as graspable interaction modules.* It uses transparent acrylic tiles to serve both as physical windows for digital information and to trigger specific actions (e.g., launch an application, or submit a query) when placed on a sensor-enhanced display surface.

2. *Mixed visual and physical interactions.* Users can interact with the information displayed by DataTiles using a pen or mouse. In addition to normal GUI interactions (e.g., mouse operations and widget operations), several combinations of physical and graphical interfaces are possible. For example, a printed high-resolution image (on a tile) can be visually fused with dynamic displayed graphics. Grooves engraved upon the tile surfaces also act as passive haptic guides of pen operations.

3. *Physical language for combining multiple tiles.* Following a simple "tile language," the placement of several tiles can compose a kind of "sentence." In this way, relatively simple combinations of tiles can express complex computational behaviors and functions.

RELATED WORK

Our inspiration for using transparent tiles as interface devices came from the film *"2001: A Space Odyssey"*, where the memory of the computer "HAL" was stored in transparent rectangular slabs. The idea of interacting *through* a transparent object is partly inspired by the Magic Lens and ToolGlass [2] a purely graphical interaction technique, and NaviCam [12], which is an augmented reality system. Graphically augmented transparent panes have been used in the metaDESK's "passive lens" [7] and the Virtual Table's "transparent props" [14]. However, we believe the integrated use of multiple transparent panes within a coordinated system is original to this work.

A number of efforts have used physical objects as a means of computer interaction [21, 16, 3, 7]. Most of these have produced special-purpose systems with fixed application domains. In contrast, there has been little research on systems that make strong use of physical artifacts, but retain the flexibility of the graphical user interface. This observation motivated us to design the system described in this paper.

It is interesting to note that computer systems allowing users to express information through systems of physical objects have existed for many years. For instance, Japan National Railways has been using a computerized ticket reservation system consisting of metal flippable plates and pegs for at least thirty years. The system recognizes reservation requests based upon the positions of pegs within holes in the plates.

IntelligentPad [17] is a visual environment that allows users to construct programs by combining sets of graphical "pads." When one pad is placed upon another active pad in the workspace, communication between the two is activated. For example, a user can put a bar graph pad onto a map pad to inspect parameters in a map (e.g., the population of each area). IntelligentPad was another source of inspiration for the use of a system of modular, intercommunicating interface elements.

Several interfaces have used systems of modular physical elements as a kind of programming interface [16, 4, 5, 1, 9]. A common theme is to use modular physical objects (such as cubes or tiles) as a kind of programming element. These systems have generally relied upon external computer monitors to display their digital state, introducing a significant physical and conceptual distance between the system's input and output pathways. Unlike these systems, DataTiles use graphically-augmented transparent objects to provide a tight connection between the system's input and output spaces.

Most systems providing a physical/digital workspace use overhead video projection to graphically augment physical objects [21, 15, 8, 19, 13]. The hardware underlying these systems tends to be complicated and expensive, because a computer projector (and also a camera) must be mounted above the desk. Our sensor-augmented flat panel display approach is much more compact and mobile, with all of the system's hardware integrated into a thin flat display.

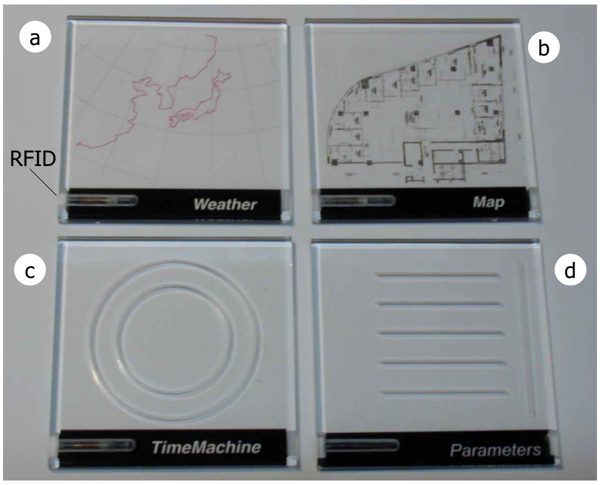

Figure 3: Tile examples. (a) and (b): partially printed tiles, (c) and (d) tiles with "grooves".

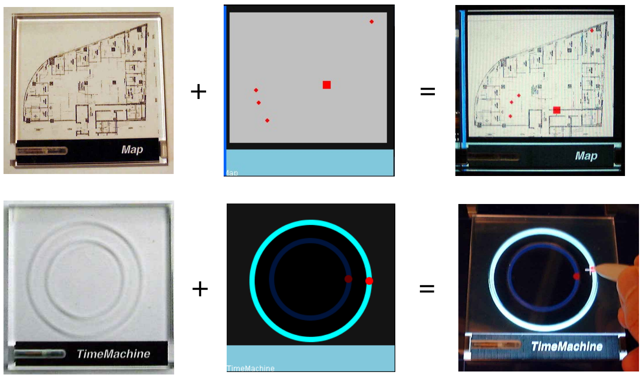

Figure 4: Combination of physical tiles and graphical information. Above: high-resolution printed information can be augmented by displayed graphics. Below: combination of physical grooves and graphical information creates a GUI widget with passive haptics.

Several interface systems, including Pick&Drop [10] and mediaBlocks [18], have used physical objects to contain and transfer digital information. For example, Pick&Drop uses a tagged pen to pick up and drop a digital object from one computer to another. MediaBlocks is based upon a series of tagged blocks that can be bound to digital content. When a user places a block in the interaction workspace, bound information appears on a physically contiguous display. DataTiles extends these ideas further, with graphical information displayed "within" its transparent physical tiles. In particular, when combined with DataTiles' stylus input mechanism, this tighter coupling between input and output space makes possible a number of new interaction techniques.

DATATILES SYSTEM DESIGN

The DataTiles system is designed and implemented to utilize the benefits of both physical and graphical user interfaces. Figure 2 shows an overview of the DataTiles system, which consists of the following elements:

- Acrylic transparent tiles with embedded radio frequency identification (RFID) tags.
- A flat-panel liquid crystal display.
- An electromagnetic pen tablet behind the display.
- RFID readers (sensor coils) mounted behind the display's cover glass.
- A custom electronic circuit for sensing multiple sensor coils using a single RFID reader.

Tray

A flat panel screen is placed horizontally to serve as both an information display and a support surface for the tiles. We call this unit the DataTiles "tray". The tray also integrates a digitizer so users can manipulate information displayed on the tray with an electronic pen. The tray's screen area is separated into square regions. RFID Sensor coils are enclosed in small boxes that are mounted on the screen surface, at the left-bottom corner of each square region.

Tiles

A DataTiles "tile" is a transparent acrylic square object with an embedded RFID tag (Figure 3). In principle, a tile acts as a graspable window for digital information. When a tile is placed on the tray, its associated function is automatically triggered. For example, placing a weather tile onto the tray automatically retrieves the current weather forecast information from the Internet and displays the processed results on the region of the screen under the tile.

Partial Printing and Graphical Augmentation To clearly express these pre-defined functions, some tiles are *partially printed* (Figure 3 (a,b)). That is, the fixed part of the information is printed or etched on a tile, and graphics from the tile tray dynamically augments this static printing (Figure 4). By integrating both static printing and dynamic graphics display, it is possible to combine high-resolution information (such as a map) with dynamic information. This printing also enables users to easily perceive the role of the tile, and better anticipate its behavior when placed on the tray.

Grooved Widgets As in a normal GUI environment, a user can also manipulate a tile's dynamically displayed objects with an electronic stylus. Moreover, some tiles are engraved with grooves to physically guide the pen. For example, the combination of a graphical scroll bar (displayed on the LCD) and a physical straight-line groove (engraved on the tile) provide a stronger impression and affordance than normal purely graphical widgets. We call this combination a *grooved widget*.

Figure 3(c) and (d) show two examples of grooved widgets. Tile (c) has circular grooves to guide circular motion, in the context of a time navigation interaction. This kind of circular movement might be difficult for users of purely graphical interfaces (given the difficulty of controlling a pointing device in this way), but users of DataTiles can easily move the pen tip through this circular groove, and have their movements correspondingly guided. We have used this ring-like widget as a kind of jog dial, suitable for manipulating continuous media.

It is also possible to add different kinds of engraved textures. For example, a small physical depression along a grooved track can give a passive haptic "clicking" feedback. Similar-

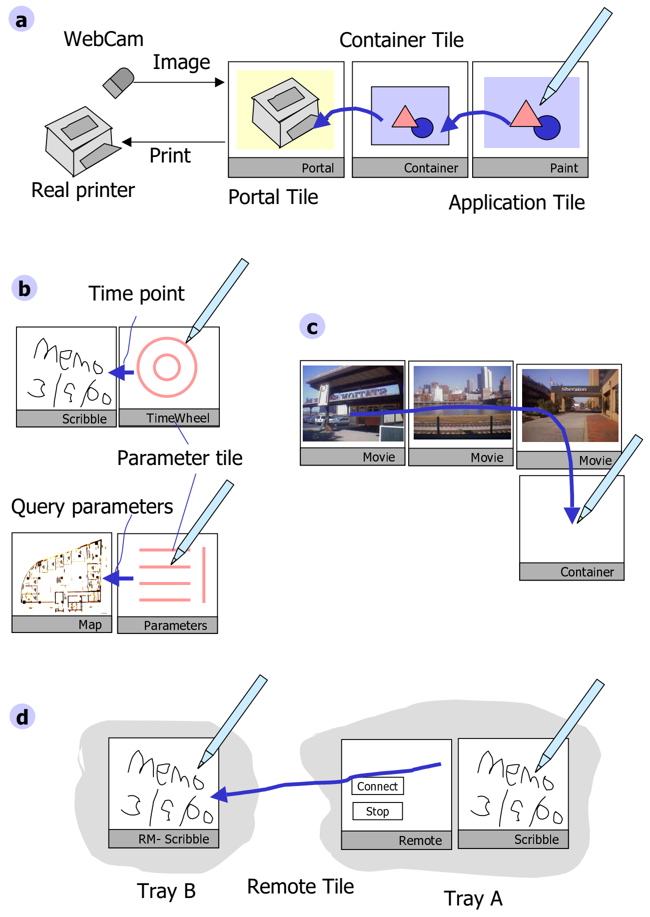

Figure 5: Examples of tiles and tile combinations. (a) An image from an application tile (right) is stored in a container tile (middle), and then transmitted to the portal tile. The portal tile represents a real world object (a printer in this example). (b) Parameter tiles can be used to specify various types of parameters. (c) Concatenates three video clips and stores item in a container tile. (c) Remote tiles are used to connect distributed tile trays. In this example, a shared drawing environment has been constructed.

ly, different physical textures can provide different impressions (e.g., rough or smoothed) when manipulating grooved widgets.

Tile Categories

Building upon these basic features, our system supports several different kinds of tiles, which can be classified into the following five categories (Figure 5):

Application Tiles: Tiles that are bound to specific applications or functions. When an application tile is placed on the tray, its pre-defined function is automatically activated. The tile will be graphically augmented as the result of this action.

Various kinds of information services – e.g., a weather forecast or live sports broadcast – can be provided as tiles. Users can regard them as physical representations of digital information. Paint or scribble tools, along with other software

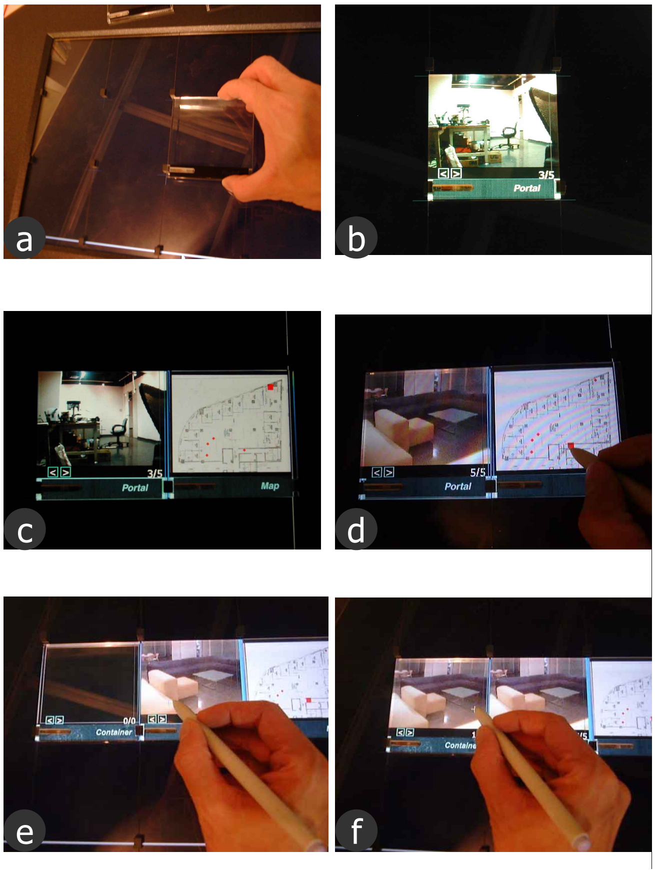

Figure 6: Examples of tile combination: (a) When a user places a portal tile on the tray, (b) an associated webcam image appears on the tile. (c) Then the user places a map tile, and the map displays locations of webcams. (d) The user clicks on a spot on the map to select another webcam. (e, f) Then the user makes an inter-tile gesture (from portal tile to the container tile) to store a snapshot image in the container tile.

tools, are additional examples of application tiles. These tiles can be used independent applications, as well as in combination with other tiles.

Portal Tiles: Tiles that represent real-world things (e.g., physical objects and places) or other digital entities. For example, a printer tile may be used to represent and manipulate printer devices. Such tiles can both display printer status, as well as serve as a destination for data transfer from other DataTiles. As another example, portal tiles may be associated with other networked appliances (e.g., televisions or stereos); webcams (Internet-linked cameras); or even people (e.g., mediated via e-mail or cell phone).

Portal tiles can also be bi-directional. For example, a whiteboard tile can both display the image of the corresponding whiteboard, while transforming drawings on the tile causes information to be sent back to the (real) whiteboard surface.

Parameter Tiles: Tiles that are used for controlling other tiles. For example, a "time wheel" tile (Figure 3(c)), with its engraved circular grooves, is used to control other tiles that represent time-based media. When placed next to another tile, such as a video tile, the current time point appears as a light spot on an ring groove. By manipulating this spot, the user can control the time point of the displayed video. Another example is a linear parameter tile (Figure 3(d)). This contains a set of linear slider grooves, where the slider's labels and values are dynamically bound to parameters of its target tile.

Container Tiles: Tiles for recording the contents of other tiles. They play a similar role to the Pick&Drop and media-Blocks systems. A user can store data in this tile and carry it between different trays or other supporting devices. The container tile can also store multiple contents.

A variation of the container tile is a "macro" tile, which records user operations applied to the connected tile. This recording of user interactions can then be attached to a target tile and replayed.

Remote Tiles: A pair of tiles for connecting different DataTiles trays. When a remote tile is placed next to any other tile, the contents of this adjacent tile are continuously transmitted to one or more "twin" remote tiles that may be placed on a different DataTiles tray. These remote tiles can be used to control a distributed information environment through the physical configuration of tiles.

Composing Tiles

As described in the previous section, each tile has an associated function that is automatically activated when a tile is placed on the tray. In addition to this simple activation, the combination of multiple tiles can express more complicated semantics. When two tiles are placed in adjacent cells on a tray, they begin to communicate with each other, with their functions affected by this digital connection. For example, when a user places a map tile next to a portal tile, the map tile shows the current location of the corresponding real-world object (Figure 6). The tile tray visually indicates this connection by illuminating the boundary between the communicating tiles. The map tile also shows other webcam locations, allowing the user to switch the currently active camera by selecting one of these points.

The user can also place a time-wheel tile next to the portal tile, which causes the camera's corresponding timestamp to appear on the time-wheel tile. The user can then inspect previous images from the webcam by manipulating the time-wheel tile. Finally, a user can place a container tile next to a portal tile and make a pen-based "copy" gesture to copy an image from the camera tile to the container tile.

The same tile can be used within several different contexts. For example a "container tile" can be placed besides any other tile to record a snapshot. As another example, when a time-wheel tile is placed next to the movie tile, it acts as a movie jog dial. If the same time-wheel tile is placed next to a portal (webcam) tile, it becomes a time-machine navigation dial [11] and can be used to retrieve past webcam images at a designated time point.

As we have mentioned, one of the motivations for our work has been the analogy between language composition and tile

Figure 7: The same time wheel tile is used in different contexts. (Left: as a jog dial for movie playback. Right: retrieve an archived data and image at a specific time point).

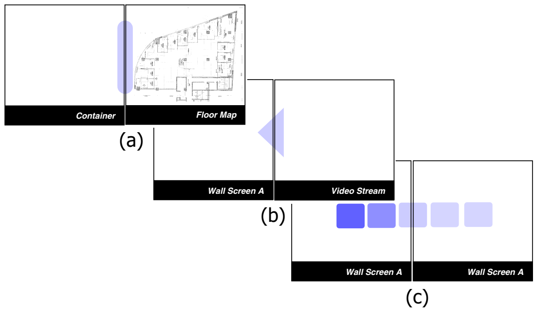

Figure 8: Several visual feedback approaches for indicating connection types. (a) one-way discrete data transmission from right to left, (b) one-way continuous data transmission, and (c) bi-directional continuous connection using animations.

composition. Figures 5 and 6 illustrate how tiles can act as modular building blocks ("words") to compose complex expressions ("sentences"). Moreover, the same tile can be used in various different contexts (Figure 7), and a user can explore and learn new tile usages on the basis of similar examples.

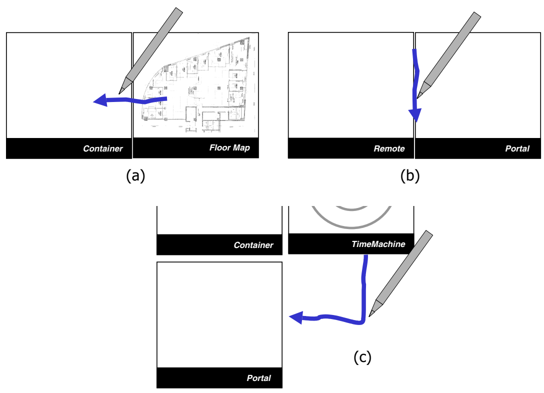

Figure 9: Inter-tile gestures by a pen to control a data connection between two adjacent tiles. (a) triggers a discrete data transmission, (b) suspends a continuous data transmission, and (c) connects two disjoint tiles. (Note: During these operations, the pen tip must be sufficiently close to the tile surfaces to be sensed, but need not touch them.)

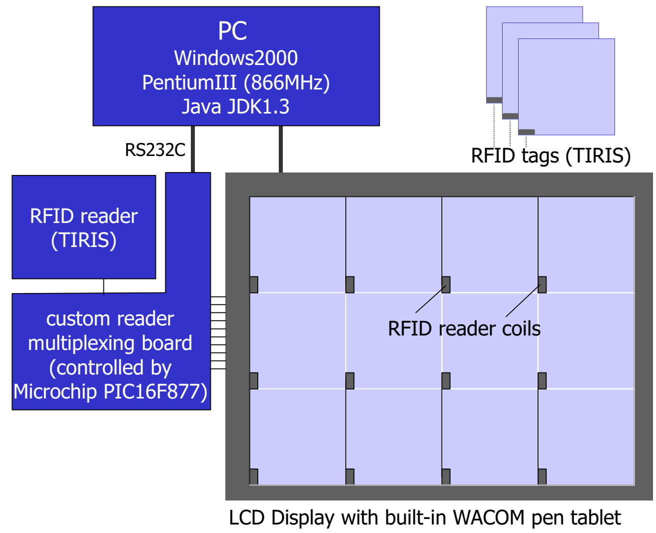

Figure 10: DataTiles system architecture

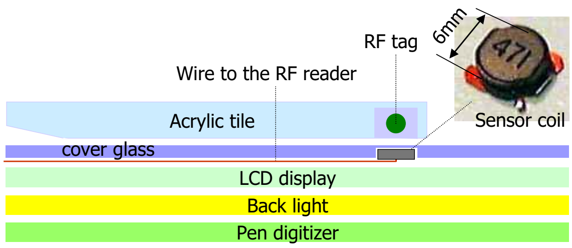

Figure 11: Cross section view of the tile tray.

There are several types of connections between adjacent tiles. One is a discrete connection, which requires an explicit operation (typically a pen gesture) to trigger actual data transmission. For example, the connection between a container tile and other tiles is discrete, so an explicit pen gesture is needed to store new data. Secondly, continuous connections support a continuous flow of control information or data between DataTiles. We provide visual feedback for distinguishing between these connection types (Figure 8), and use several "inter-tile" pen gestures to control these connections (Figure 9).

SYSTEM DESCRIPTION

Sensor Architecture

Figure 10 shows our current system configuration. It consists of a a liquid crystal display that integrates an internal electromagnetic pen tablet (based on a Sony VAIO PCV-LX80), as well as an array of coil antennas mounted on the display's surface. The pen tablet is capable of sensing pen positions when the pen is sufficiently close (within about 15 mm) to the display surface. This range is enough for manipulating displayed information through a tile placed upon the display's surface.

Tiles are 5mm-thick transparent acrylic squares with a size of 75 mm $\times$ 75 mm. Each tile has an embedded RFID tag made by Texas Instruments [6]. These tags are sensed by an RFID reader. We mounted an array of twelve inductor coils (each $47\mu H$) under the cover glass of the LCD. Each inductor coil

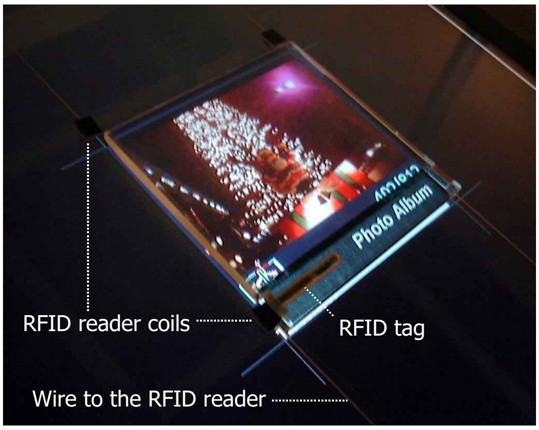

Figure 12: Close-up of the reader antenna and a DataTiles with an embedded RFID tag.

acts as a tag reader antenna, which is connected to the RFID reader by thin (0.05 mm) copper wire (Figure 11)

Custom electronics multiplex the tag reader across these twelve coils, similar to an approach described in [22]. A microcontroller manages this multiplexing circuit, as well as communications with the RFID reader. The microcontroller informs the DataTiles software about the presence/absence of RFID tags through an RS232 serial connection.

Figure 12 shows a close-up photograph of the installed antenna coils and a tile with an embedded RFID tag. Since the coils is very small (6 mm in diameter), the maximum recognition distance between the coil and the tag is also short (about 10 mm). This distance limitation is not a problem, however, because our interface design assumes that tiles will be recognized only when they are placed within the tray's grid cells. Another benefit of this limited range is that it allows simultaneous operation of the pen digitizer and RF readers, even though both technologies use similar frequency bands.

The per-coil recognition time is about 100 ms. Multiplexed over twelve coils, the current update rate is about 0.8 Hz, with an average recognition time of 0.6 s. While this speed is acceptable, it is slower than desirable, and does not scale to support larger numbers of sensing cells. We are currently implementing a second-generation sensor system that will incorporate two or more RFID readers operating in parallel.

Software Architecture

The DataTiles software is written in Java. When a user places a new tile on a tray cell, its ID is recognized, and its associated Java class is dynamically loaded (if necessary). The system then creates a corresponding Java tile object. Some tiles may also reload their contents (e.g., the persistent contents of a container tile).

Once the tile object is instantiated, it checks nearby tiles for the possibility of inter-tile communication. We have implemented several sets of inter-tile communication protocols, each defined as Java interface. For example, a tile class that supports image data creation and time-machine interaction declares itself such as:

```
public class SampleTile extends Tile implements
    ImageCreateInterface, TimeMachineInterface {
        ......
}
```

When two tiles are placed in adjacent cells, each checks the other for its available interfaces. In Java, this process can be easily implemented by using the `instanceof` operator:

```
Tile tile = getLeftTile();
if (tile instanceof ImageCreateInterface) {
    ....
} else if (tile instanceof ....) {
   ....
}
```

Using this simple method, tiles can communicate with each other even when they do not know the actual classes in advance.

USER EXPERIENCE

We demonstrated the DataTiles system to visitors during a laboratory open house day. During this, we observed interesting user reactions and received many comments.

Physical Features The physical form factor of the DataTiles and tray made strong impressions on these early users. Many commented on the appropriateness of tile sizes and tactile feelings. The current DataTiles implementation uses small rectangular protrusions on the display surface as "guides" to correctly position and fixture tiles. However, some users commented that these protrusions interfered with their interaction, and should receive further design attention.

Parallax is another issue. While we printed information on the rear of tiles, grooves are engraved on the front. This causes parallax and registration issues between the dynamic graphics, printed graphics, and grooves, which is especially noticeable while using the grooved widgets.

Tile Composition Many visitors liked the idea of composing tiles, and tried to create new combinations that we had not anticipated. There was sometimes confusion when two or more possible tile connections existed – especially when many tiles were present on the tray. Even though the system provided visual feedback to indicate the types of tile connections, this was not well recognized by (untrained) users. We are currently re-designing more an improved set of visual feedbacks, with emphasis on the use of graphical animations.

While designing the DataTiles system, we had numerous discussions on introducing semantics to different spatial configurations of tiles. For example, a horizontal connection might represent data flows, while vertical connections might represent control flows. Howerever, it seemed that these semantics might be too complicated for novice users. As a result, we have not currently used spatial semantics, though believe this may hold promise for future work.

CONCLUSIONS AND FUTURE DIRECTIONS

This paper presented a system that integrates the benefits of both graphical and physical interfaces. Tagged transparent tiles can be used as modular building blocks for computational expressions, with compositions of multiple tiles expressing rich and open-ended configurations.

Our research is still at an early stage, and we are considering several possible future directions and research issues.

Application Domains

We consider one promising area of the DataTiles system to be controlling home and office appliances. Various kinds of information appliances, including computationally-enhanced TVs, VCRs, printers, phones, or cameras, might be integrated through corresponding portal tiles. The tile tray might serve as a kind of "digital dashboard" for these appliances, allowing people to simply express rich and open-ended interconnections of multiple networked devices through the tile layouts.

Among many other possibilities, we will consider two promising domains.

Media Editing Environment: Media editing environments show special promise for DataTiles usage. For example, rapidly-customizable video editing environments might be constructed by combining several tiles. Some tiles might contain media fragments, allowing users to organize and combine them on the tray. The time wheel tile might be used to define in/out points, with parameter tiles used to control digital effects. The results of this editing might be stored in a container tile, or transmitted to other devices through portal or remote tiles.

Education Platform: Education (edutainment) platforms offer another potential opportunity. Many different educational application modules might be represented in DataTile form, including sound, animation, and simulation tiles. These could be combined to construct and configure a variety of applications. As a simple example, one or more "Pong" tiles might be combined with one or more remote tile to construct a distributed game. DataTiles might also have special promise for the redesign of digital toys that presently use console-based visual programming languages (e.g., LEGO MindStorms).

Other types of Physical Objects

The shapes of tagged transparent objects are not limited to rectangular tiles, and their combination into more complex compositions is not limited to grid-based concatenation on simple horizontal trays. The vertical stacking of multiple tiles is one such promising extension.

The tray should not be limited to a 2D grid. For example, 1D trays might be promising for installation in a variety of places, such as a drawer for a living-room table or on a wall. Tile "racks" or "shelves" are another possibility. For example, each slot of a tile rack/shelf might have an RFID sensor and an LED. Upon receiving an e-mail from a person, the corresponding people tile might be illuminated in the rack.

It is also possible to integrate active electronics within DataTiles. Such tiles might integrate real dials, buttons, or other sensors; microprocessors; local storage; and perhaps even ancillary displays, cameras, etc. These enhanced tiles could still be used in combination with normal passive tiles. For example, a tile embedded with a small microphone might be used to store a voice memo into a container tile. Additionally, other digital devices with embedded RFID tags or RF transducers could also be a part of the DataTiles. For example, a (tagged) cellular phone might be placed onto a DataTiles tray, allowing the bidirectional exchange of data between the telephone and other tiles.

We are aware that among the many possibilities opened by DataTiles' physically embodied form, there are also real concerns such as scalability, clutter, and loss. The thickness and material (rigid acrylic) of our present tiles may contribute to this, making the tiles more closely resemble electronic devices than, say, printed media such as baseball, business, or playing cards. However, we believe this observation may point to additional opportunities, such as "decks" of literally and metaphorically "lighter-weight" tiles. This analogy suggests decks of standardized tiles; collectible tiles; advertising tiles; "home tiles" representing online personal and commercial spaces; and many other possibilities. These fusions of physical and digital media suggest many rich opportunities for continuing research.

ACKNOWLEDGEMENTS

We thank Tota Hasegawa and colleagues at Sony CSL Interaction Laboratory for the initial exploration of ideas described in this paper. Masaya Shimomatsu and Shigeru Tajima contributed to the hardware implementation. Hiroshi Ishii supervised and encouraged parallel research on LCD-backlit acrylic objects at the MIT Media Lab, which contributed to our efforts. We also thank Toshi Doi and Mario Tokoro for their continuing support of this research.

REFERENCES

1. David Anderson, James L. Frankel, Joe Marks, Aseem Agarwala, Paul Beardsley, Jessica Hodgins, Darren Leigh, Kathy Ryall, Eddie Sullivan, and Jonathan S. Yedidia. Tangible Interaction + Graphical Interpretation: A new approach to 3D modeling. In *SIGGRAPH 2000 Conference Proceedings*, pages 393–402, 2000.

2. Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, and Tony DeRose. Toolglass and Magic Lenses: The see-through interface. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 73–80, August 1993.

3. George W. Fitzmaurice, Hiroshi Ishii, and William Buxton. Bricks: laying the foundations for graspable user interfaces. In *CHI'95 Conference*, pages 442–449, 1995.

4. M.G. Gorbet, M. Orth, and Hiroshi Ishii. Triangles: Tangible interface for manipulation and exploration of digital information topography. In *CHI'98 Proceedings*, pages 49–56, 1998.

5. System Watt Inc. ROBOCUBE. http://www.watt.co.jp.

6. Texas Instruments. Radio frequency identification systems. http://www.ti.com/tiris/.

7. Hiroshi Ishii and Brygg Ullmer. Tangible Bits: Towards seamless interfaces between people, bits and atoms. In *CHI'97 Proceedings*, pages 234–241, 1997.

8. W.E. Mackay, D.S. Pagani, L. Faber, B. Inwood, P. Launiainen, L. Brenta, and V. Pouzol. Ariel: augmenting paper engineering drawings. In *CHI'95 Conference Companion*, pages 420–422, 1995.

9. Tim McNerney. Tangible programming bricks: An approach to making programming accessible to everyone. *MS Thesis, MIT Media Lab*, 2000.

10. Jun Rekimoto. Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments. In *Proceedings of UIST'97*, pages 31–39, October 1997.

11. Jun Rekimoto. Time Machine Computing: A time-centric approach for the information environment. In *Proceedings of ACM UIST'99*, pages 45–54, 1999.

12. Jun Rekimoto and Katashi Nagao. The world through the computer: Computer augmented interaction with real world environments. In *Proceedings of ACM UIST'95*, pages 29–36, November 1995.

13. Jun Rekimoto and Masanori Saitoh. Augmented Surfaces: A spatially continuous workspace for hybrid computing environments. In *Proceedings of ACM CHI'99*, pages 378–385, May 1999.

14. Dieter Schmalsteig, Miguel Encarnacao, and Zsolt Szalavari. Using transparent props for interaction with the virtual table. In *Proc. of Interactive 3D Graphics (I3DG'99)*, 1999.

15. Itiro Siio. InfoBinder: a pointing device for a virtual desktop system. In *6th International Conference on Human-Computer Interaction (HCI International '95)*, pages 261–264, July 1995.

16. H. Suzuki and H. Kato. AlgoBlock: a tangible programming language, a tool for collaborative learning. In *Proceedings of 4th European Logo Conference*, pages 297–303, August 1993.

17. Y. Tanaka and T. Imataki. IntelligentPad: A hypermedia system allowing functional composition of active media objects through direct manipulations. In *Proc. of th IFIP 11th World Computer Congress*, pages 541–546, 1989.

18. Brygg Ullmer, Hiroshi Ishii, and Dylan Glas. mediaBlocks: Physical containers, transports, and controls for online media. In *SIGGRAPH'98 Proceedings*, pages 379–386, 1998.

19. John Underkoffler and Hiroshi Ishii. Illuminating Light: An optical design tool with a luminous-tangible interface. In *CHI'98 Proceedings*, pages 542–549, 1998.

20. Mark Weiser. The computer for the twenty-first century. *Scientific American*, pages 94–104, September 1991.

21. Pierre Wellner. The DigitalDesk calculator: Tangible manipulation on a desk top display. In *Proceedings of UIST'91, ACM Symposium on User Interface Software and Technology*, pages 27–34, November 1991.

22. P. Yarin. Towards the distributed visualization of usage history. *MS Thesis, MIT Media Lab*, 1999.