Tangible Menus and Interaction Trays: Core tangibles for common physical/digital activities

Brygg Ullmer, Rajesh Sankaran, Srikanth Jandhyala, Blake Tregre, Cornelius Toole, Karun Kallakuri, Christopher Laan, Matthew Hess, Farid Harhad, Urban Wiggins, and Shining Sun
Louisiana State University, Dept. of Computer Science + Center for Computation and Technology (CCT)
ullmer@cct.lsu.edu

ABSTRACT
We introduce core tangibles: physical interaction elements which serve common roles across a variety of tangible and embedded interfaces. We describe two such tangibles: tangible menus and interaction trays. These may be composed together to dynamically bind discrete and continuous interactors to various digital behaviors. We discuss our approach, implementation, and early usage experiences.

Author Keywords
core tangibles; tangible menus; interaction trays; blades+tiles

INTRODUCTION
Tangible human-computer interaction dates back at least to the 1976 "Slot Machine" of Perlman, Hillis, and Papert [13]. The three decades between then and now is similar in duration to the time between the earliest graphical interfaces (e.g., the 1958 SAGE) and the 1984 Apple Macintosh. Graphical interfaces (GUIs) evolved over these decades from isolated, idiosyncratic islands of software and hardware into ecosystems of diverse applications interlinked by core, consistent sets of operations (e.g., launch/quit, load/save, and copy/paste). We see this evolution as an important factor in the Star and Macintosh's launch of GUIs into ubiquity.

We find lessons here for tangible and embedded interfaces. We expect these interfaces will be marked by greater diversity than GUIs. Still, we believe both users and developers would benefit from common subsets of tangibles which serve common roles and maintain interoperability across diverse systems. Users could move between systems with knowledge and expectation of certain common, interoperable operations and information. Developers could reuse these, focusing on interface aspects specific to their application domain.

Toward this, we are pursuing the idea of *core tangibles.* Tangibles are the manipulable physical elements of tangible interfaces (TUIs) that serve as representations and controls for digital information [18]. Core tangibles support operations common to diverse applications employing tangibles and other styles of interaction. They complement *domain tangibles* (specific to particular application domains).

TEI 2008, 18 February - 21 February, 2008, Bonn, Germany.

Figure 1. Tangible menus and interaction trays: *several data and parameter menu cards are shown with a three-wheeled parameter tray*

We introduce two kinds of core tangibles: *tangible menus* and *interaction trays*. Tangible menus are tagged, printed, structured surfaces of varying size and material, used to represent and access diverse data, parameters, operations, and other associations. Interaction trays are devices which integrate different combinations of interactors (RFID readers, knobs, LEDs, etc.) with receptacles for tangible menus.

The combination of tangible menus and interaction trays provides an open-ended means for physically invoking "core operations" such as opening and closing applications; loading, saving, and navigating data; and adjusting parameters. This initial palette of operations is similar to commands familiar from the "file," "edit," and "start" menus of GUIs.

We do not intend to physically export the full set of digital behaviors from GUIs. Rather, our interest is to provide physical access to a small subset of digital operations which are common to a wide spectrum of computational activities. We seek to make these operations accessible from a variety of physical and social interaction contexts, without forcing users to change their mode of interaction. For example, we wish to allow users of a tangible interface to (e.g.) physically access new applications or data, without needing to switch to a GUI to complete this operation.

As in [5, 14] et al., we have adopted a hybrid interaction approach, combining high-resolution "printables;" tagged and force-feedback graspables; and small and large screens. We seek to complement and bridge diverse interfaces (TUI, GUI, VR, et al.) with common, repurposable elements, supporting broader efforts to grow TUIs beyond isolated point systems.

We begin by describing tangible menus and interaction trays in greater detail, along with related work. We then describe our implementation and early usage experiences.

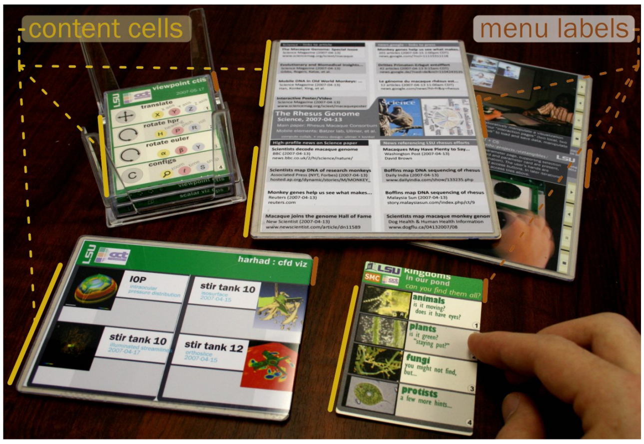

Figure 2. Tangible menus: *clockwise from top-left, parameter cards in card deck; data menus; data menu card; and short data menu*

TANGIBLE MENUS

Tangible menus represent a series of digital associations on a physical surface, indicating these with text, visual, and/or relief labels (Figures 1-3, 4b, 6b). They are used in composition with interaction trays and in other interaction contexts.

Tangible menus (t-menus) integrate aspects of graphical and culinary menus; file dialogs; web pages; phicons [18]; and the container concept [10]. They may be of varying thickness, and fabricated of diverse materials (e.g., paper, wood, plastic). They may be single- or double-sided, and used individually or as multiples in decks or books. They may also be diversly tagged, including RFIDs and front or rear barcodes.

Several tangible menus are pictured in Figure 1, 4b, and 6b. Each couples a paper label with an RFID tag inside a clear plastic holder. This is reminiscent of CD-ROM caddies, suggesting holder-free versions are also possible. The examples illustrate menus containing data; operations; and parameters.

Tangible menus are spatially partitioned according to a consistent format, within which many visual and physical variations are possible. This partitioning supports the representation of diverse aggregates of information, including data, parameters, operations, applications, people, and places.

Our tangible menus are based on an RFID-tagged, printed surface which is divided into two kinds of elements: *content cells* and supporting graphics. Content cells are fixed-dimension regions which can be linked to one or multiple associations. These regions each typically include an icon or thumbnail image, and one or more lines of descriptive text, all describing the contents of each cell. Supporting graphics include headers, footers, and other information indicating the menu's function, authorship, history, etc.

We currently use tangible menus of three standard sizes[1]. Our *menu cards* are the size of playing cards, and are divided into up to four content cells. We also use a *standard menu* of double height and width, with up to 16 content cells per side; and a *short menu* with up to eight content cells. Each format brings tradeoffs in real estate, footprint, composability, etc.

[1]normal menus, 5"x7"; short menus, 5"x3.5"; menu cards, 2.5"x3.5". These are common US media formats. In other regions, other common (metric) dimensions might be more appropriate.

Figure 3. Several tangible menus and interaction trays. *From top right: application tray; param tray w/ two param cards; menu tray w/ short menu*

We hope tangible menus may be useful within many tangible and embedded interaction contexts. These systems will likely employ diverse tangibles. Still, analogous to the way the menu bar is present in all Macintosh applications, an ability to access some form of tangible menus from many physical contexts and applications could hold substantial value. Decks of standard menus are possible. Variations on interaction trays could be used, or tangible menus could be touched directly (e.g., with capacitive sensing or computer vision).

INTERACTION TRAYS

To access the digital contents of tangible menus, we use t-menus in composition with other interaction elements. We call our first such family of devices *interaction trays* (Figure 3). The "tray" term was first used in an HCI context by Fitzmaurice et al. in [7]; we continue and extend its use.

Interaction trays are interaction devices which compose a changing palette of tangible menus with complementary physical interactors (e.g., buttons, knobs, and integrated displays). Interaction trays typically provide controls for accessing the contents of tangible menus. In turn, some tangible menus are coupled to behavior which rebinds the functions of interaction tray controls (e.g., knobs or sliders).

Interaction trays integrate aspects of GUI windows, widgets, and controls. Some applications can be serviced by a single tray (or one per user). Other applications may require multiple trays, potentially complemented by (e.g.) PDAs, speech, gesture, or other interaction modalities. We have chosen letter size (8.5"x11") and A4 as basic footprints, allowing us to leverage pre-existing binders, boxes, bookshelves, etc.

We have implemented three interaction trays: a *menu tray*; a *parameter tray*; and an *application tray* for a microscopy application (Figure 3). We believe the menu and parameter trays can be useful within many TEI applications. The application tray is more specific. While potentially useful for other simple media applications, it also illustrates prospects for other application- and domain-specific trays.

menu tray: The menu tray (Figure 3) has a well[2] which accomodates either one normal menu; two short menus; or four

[2]The 2006 Apple HIG introduces the GUI *image well* as "a drag-and-drop target for an icon or picture" [1]. We use "menu well" to describe physical receptacles serving similar purposes for t-menus.

menu cards. The menu well is flanked by selection buttons and side-illuminating LEDs, placed adjacent to the t-menus' content cells. Menu tray buttons can be associated with varying semantics, as a function of both tangible menu contents and the associated application. One variation is mapping button press events to simple "invocation" of the adjoining menu content cell (e.g., playing an associated video). Alternately, one menu well could be mapped to "save" and a second to "load;" or a second menu could be used to specify the operation. Other mappings are also possible.

parameter tray: Our current param(eter) tray has two wells for menu cards, and three actuated knobs (Figures 1, 3). Each menu card holds up to four sets of knob mappings. With this approach, selection of a content cell maps all three knobs to the specified mappings (Figure 4b). As a concrete example, in a scientific visualization application now under development, three param card t-menus are *viewpoint controls; scalar viz ops*; and *vector viz ops*. For instance, with the *viewpoint* t-menu, the first cell maps the tray's knobs to translation in X, Y, and Z (Figure 4b). The remaining cells bind rotational, temporal, and zooming operations to the knobs. The haptic knobs support parameter-specific force feedback, and also help mediate collaborative use (where multiple users may simultaneously access a parameter).

application tray: We have also implemented an application tray supporting image capture and playback by several kinds of microscopes (handheld and scanning electron). This tray integrates elements of the menu and parameter trays. It is made in a different size and of different materials, partly corresponding to a different intended audience (K-12 students, in contrast with the current university end-users of the other trays). Nonetheless, it uses the same general menu cards (albeit often with different visual style; see Figure 2).

RELATED WORK

We have been influenced by many systems. Tangible menus draw from the cards of the 1976 Slot Machine [13], and build on extensive work with paper-based tangibles (e.g., [11, 12, 16]). They also build upon other recent card-based research [3, 4, 17], while adding a structured visual format with multiple subelements per card. The idea of file and edit menus for physical spaces has been discussed in [6], albeit there in the context of mobile telephone and PDA interaction. Interaction trays build on the tray of Bricks [7] and the Magic Book [2], along with other more recent work [14].

IMPLEMENTATION

Tangible menus and interaction trays could be implemented in many different ways. The menus and trays could each be tagged (whether with RFIDs or optical codes), and tracked on instrumented workbenches, tables, or walls with RFID- or vision-based infrastructures. They could also be implemented with existing tangible toolkits (e.g., [8, 9]); or with custom ad-hoc electronics. From a physical design perspective, tangible menus and interaction trays could take quite different physical forms while serving similar ends.

Our implementation of core tangibles has been shaped by several objectives. We wished to deploy 10-20 interaction tray prototypes at moderate unit cost. We sought to structure these prototypes in a way that maximizes our ability to incrementally, economically evolve these prototypes. Also, while our initial interaction trays are fairly general-purpose devices, we wished to follow an approach supporting the creation of diverse, task/domain-specific tangible interfaces.

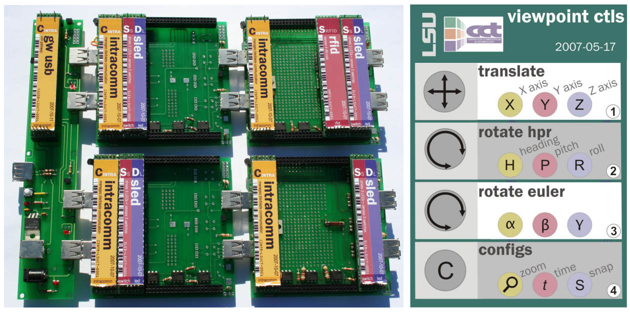

Figure 4. a) Interaction tray implementation; b) param card example

These goals have lead us to pursue an approach based on an evolving toolkit of "blades" and "tiles" [15] (Figure 4a). This choice has resulted in unusual flexibility for rapidly evolving and customizing our prototypes as we gain usage experience. For example, we initially designed our hardware around menu cards. When we decided to introduce short and regular menus, less than two hours of electronics alteration and one line of firmware modification was required. Similarly, our example application tray uses the same underlying hardware as the menu and parameter trays, despite its different physical form. While our current interaction trays support relatively simple interactions, our architecture scales to support far more ambitious interfaces – e.g., integrating hundreds of LEDs, motors, and RFID sensing points.

For the housings of interaction trays, we use a desktop 25-watt laser cutter. For tangible menus, we have re-purposed clear baseball card boxes and picture sleeves, with laser-cut RFID holder inserts. For printing tangible menus, we use inkjet, dyesub, and thermo-rewrite printers.

The utility of these interaction devices is deeply dependent on their supporting software. We have completed a number of early software prototypes, and are working to develop these further. In the medium term, we envision a *core operations API* which might allow tabletop applications to accept bindings from diverse interaction modalities in a uniform way for operations like load, save, copy, and paste. The input retargeting work of [5] provides a valuable step in this direction. This work was oriented at legacy GUI applications; we suspect additional evolution will be necessary to support the design of new applications.

EARLY USAGE EXPERIENCES

As pilot uses, we have employed our core tangibles in a number of contexts. Our most frequent usage has been in support of remote collaborative visualization. Some of these have been among scientists and graduate students (e.g., Figure 5); others have involved child users (Figure 6). Other pilot uses have included video conferencing support; interactive kiosks at museums, conferences, and open houses; laboratory use supporting visualization segmentation; and interaction consoles for handheld and scanning electron microscopes.

Figure 5. Co-located and distributed users collaboratively manipulate *a visualization application in an AccessGrid meeting room. Four interaction trays (two local, two remote) are used with tangible menus to load and save data, manipulate parameters, etc.*

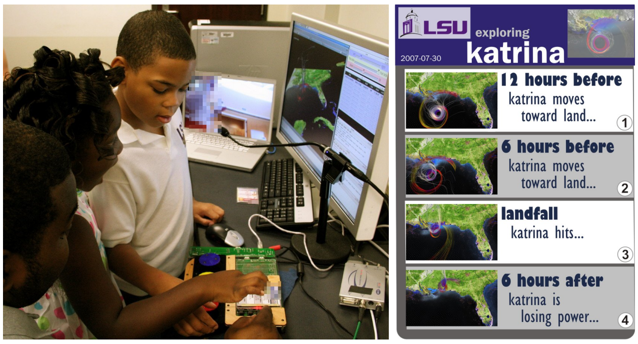

Figure 6. Child users engaging in distributed, collaborative interaction *a) with two children in another state, collaboratively loading and manipulating several scientific datasets; b) one of several menu cards used.*

We have chosen a technically challenging path for implementing interactions trays, with hopes this may pay dividends in supporting wider tangible development and deployment efforts. As such, while we have realized working prototypes in each of the above contexts, our software is just beginning to reach a stage supporting production use.

Our tangibles have seemed to lower the threshold for manipulating complex software applications by diverse user populations. At the same time, where graphical interfaces are familiar to most users, our tangibles introduce new interaction styles which must be learned. This learning step (even if modest) must be accounted for. Also, in our work with child users, we have observed distinctions between play and comprehension. Learning how to optimize these interactions, and understanding their limitations, will require more work.

The example applications illustrated in Figures 5 and 6 combine core tangibles with graphical displays. We have also explicitly designed core tangibles for use as elements of other TUI, augmented and virtual reality, and other interactive systems such as [2–4, 7, 11–14, 19], as well as in new systems.

CONCLUSION

We have introduced the concept of core tangibles. This has been supported with two instances: tangible menus and interaction trays. We have described several variations of each, including tangible menus in regular, short, and card form; and menu, parameter, and application trays.

We have gained early use experiences in scientific visualization, video conferencing, and media manipulation contexts, with input from scientist, student, and child users. Having stabilized our electronics, mechanics, and firmware, our efforts now concentrate on development of supporting software, and deploying and evaluating our tangibles over extended use. We are planning to release our implementations as open-source hardware and software, in hopes of stimulating development and adoption of interoperating tangibles, and engaging a broader community of users and developers.

ACKNOWLEDGEMENTS

This research is supported by NSF MRI 0521559, with co-PIs S.D. Beck, E. Seidel, S.S. Iyengar, L. Butler, and collaborators. An NIH P20 RR16456 supplement supported additional deployment. Special thanks to Miriam Konkel, Frederic Konkel, John Quebedeaux, Werner Benger, Hans-Christian Hege, Olaf Paetsch, Christa Hausmann-Jamin, Almuth Barta, Matthias Kranz, Tolulope Ogunbakin, Courtney Oliver, Michael Carroll, and Mariel Losso.

REFERENCES

1. Apple. Human Interface Guidelines: The Apple Desktop Interface. `http://tinyurl.com/vgge`, 2006.
2. Billinghurst, M., Kato, H., and Poupyrev, I. The MagicBook: moving seamlessly between reality and virtuality. In *IEEE Computer Graphics Applications*, v20n3, pp. 6-8, 2001.
3. Blackwell, A., Stringer, M., et al. Tangible Interface for Collaborative Information Retrieval. In *Proc. of CHI'04*, pp. 1473-1476, 2004.
4. Buur, J., and Soendergaard, A. Video card game: an augmented environment for user centred design discussions. In *Proc. of DARE 2000*, p.63-69, 2000.
5. Carter, S., Hurst, A., et al. Dynamically Adapting GUIs to Diverse Input Devices. In *Proc. of ASSETS'06*, pp. 63-70, 2006.
6. Fano, A. What are a location's "file" and "edit" menus? In *J. of Personal and Ubiq. Computing,* v5n1, pp. 12-15, 2001.
7. Fitzmaurice, G., Ishii, H., and Buxton, W. Bricks: Laying the Foundations for Graspable User Interfaces. In *Proc. of CHI'95,* pp. 442-449, 1995.
8. Greenberg, S., and Fitchett, C. Phidgets: easy development of physical interfaces through physical widgets. In *Proc. of UIST'01*, pp. 209-219, 2001.
9. Hartmann, B., Klemmer, S., et al. Reflective physical prototyping through integrated design, test, and analysis. In *Proc. of UIST'06*, pp. 299-308, 2006.
10. Holmquist, L., Redström, J., and Ljungstrand, P. Token-Based Access to Digital Information. In *Proc. of Handheld and Ubiquitous Computing (HUC)'99,* pp. 234-245, 1998.
11. Mackay, W. and Pagani, D. Video mosaic: laying out time in a physical space. In *Proc. of Multimedia'94* pp. 165-172, 1994.
12. Nelson, L., Ichimura, et al. Palette: a paper interface for giving presentations. In *Proc. of CHI'99,* pp. 354-361, 1999.
13. Perlman, R. Using Computer Technology to Provide a Creative Learning Environment for Preschool Children. MIT Logo Memo #24, 1976.
14. Rekimoto, J., Ullmer, B., and Oba, H. DataTiles: A Modular Platform for Mixed Physical and Graphical Interactions. In *Proc. of CHI'01,* pp. 269-276, 2001.
15. Sankaran, R., Ullmer, B., et al. Blades and Tiles: an extensible hardware architecture for interaction devices. In *Proc. of Ubicomp'07*, pp. 260-263, 2007.
16. Signer, B., and Norrie, M. PaperPoint: a paper-based presentation and interactive paper prototyping tool. In *Proc. of TEI'07*, pp. 57-64, 2007.
17. Sokoler, T., Edeholt, H., et al. VideoTable: a tangible interface for collaborative exploration of video material during design sessions. In *CHI'02 Extended Abstracts*, pp. 656-657, 2002.
18. Ullmer, B., and Ishii, H. Emerging frameworks for tangible user interfaces. In *IBM Systems Journal,* 30(3-4), pp. 915-931. 2000.
19. Underkoffler, J., Ishii, H. Urp: A Luminous-Tangible Workbench for Urban Planning and Design. In *Proc. of CHI'99* , pp. 386-393, 1999.