Demonstrating Flatland User Interfaces

Takeo Igarashi
Dept. of Info. Eng., Univ. of Tokyo
7-3-1 Bunkyo-ku, Hongo,
Tokyo 113-8656, JAPAN
+81 3 3812 2111 ex.7413
takeo@mtl.t.u-tokyo.ac.jp

Elizabeth D. Mynatt, W. Keith Edwards, Anthony LaMarca
Xerox PARC
3333 Coyote Hill Rd.
Palo Alto, CA 94304, USA
+1 650-812-4405
mynatt,kedwards,lamarca@parc.xeroc.com

ABSTRACT

Flatland is an augmented whiteboard designed to assist informal activities in one's office environment. Current research focus is on the software architecture to support stroke-based interaction. This demonstration illustrates the user interface aspects of the board, focusing on its screen real estate management, flexible control of various behaviors working on the surface, and context based search mechanism. The combination of simple user interface based on strokes and advanced stroke management architecture can greatly enhance the basic functionality of physical whiteboards without damaging original strength of them.

Keywords

Whiteboard, pen computing, user interface

INTRODUCTION

A whiteboard in one's office plays an important role in one's daily activity. It is used to take quick notes, organize infant ideas, sketching diagrams. It is also used as communication tool to discuss something with others. Its physical presence and informal writing interface leads to characteristic usage style quite different from standard desktop systems.

Given the observation, we are working on computationally enhanced whiteboard system, called Flatland. Compared with other research projects on electric boards [2,3], we are especially interested in the software system to support *persistent strokes* on the board. Whiteboards are used daily basis, and strokes on it can be present for weeks. New strokes are drawn on some open space, resulting in a several clusters of strokes on the surface. The user sometimes refers very old drawings.

This demonstration illustrates how Flatland supports the use of independent clusters of strokes, how specific computational support is applied to each cluster, and how strokes on the surface is recorded and retrieved. Great care was taken to preserve the simple interaction style and the informal appearance of physical whiteboards.

BASIC USER INTERFACE

Our design is based on two-button event model. Primarily button events (simple pen down on the surface) are used for writing strokes, and secondary events (pen down with modifier button) is used for controlling the board. Primarily pen movements always generate free form strokes regardless of surface status, which is important to give the feel of real marker. On the other hand, secondary pen movements are interpreted and presented differently based on the context, such as dragging, clicking, and gestures. We use pie menu and marking menu for some operations [4].

SCREEN REALESTATE MANAGEMENT

The system maintains several clusters of strokes (called segments) on the surface. Each segment is explicitly presented to the user by a boundary surrounding its strokes. When the user draws a stroke on some open space, a new segment is assigned for the stroke. If a stroke is drawn within or close to an existing segment, the stroke joins to the segment. Unlike other whiteboard systems based on explicit grouping [2], these segmenting operations are completely automatic. If necessary, the user can also manually control it using splitting and joining operations.

Segments are not allowed to overlap each other. The user can drag the segment by grabbing its boundary, but if the segment collides with another segment, the collided segment pushed away. If no more space is available, the collided segment starts to shrink to give more space (Figure1-A). Pushing and shrinking also happens when a segment expands according to joining of a new stroke. When the user starts working on a shrunk segment, the segment restores its original size. As a result, the user can maintain multiple segments on a single surface without anything obscured.

The user can also flip to a new blank surface by pulling a side of the board. The metaphor is a long, partitioned roll of screen. This mechanism provides a fast way to get a clean board when the board becomes crowded with segments, as well as to hide personal content and to get

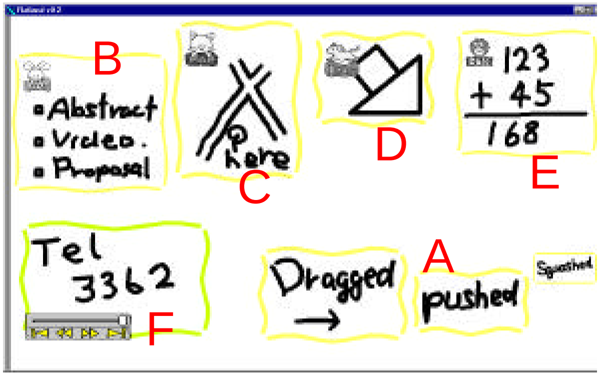

Figure 1. Magic Board example

space for discussion when a visitor comes. The horizontally connected screens can also be seen as a history of the activity, which allows quick access to past drawings.

BEHAVIORS ON SURFACE

In addition to providing efficient screen real estate management mechanism, one of our research goals is to provide powerful computational support for some specific activities on the surface. This additional computational support is implemented as *behaviors* working on each segment. Multiple behaviors can be dynamically applied to and removed from any segment. Most behaviors work as basic infrastructures for the segment and they are invisible form the user, while some high level behaviors are explicitly applied and serve for specific tasks. The following is the list of currently available high level behaviors.

To do list: it maintains a vertical list of handwriting items with check boxes, and the user can remove and reorder a to do item with a single operation (Figure1-B).

Map Drawing: strokes turns into a double line representing a road. Intersections are handled appropriately (Figure1-C).

2D geometric drawing: it automatically beautifies free strokes considering possible geometric relations. It also predicts the next drawing operation [1] (Figure1-D).

Calculator: the behavior recognizes the handwritten formula in the segment and returns the result of calculation (Figure1-E).

These behaviors are different from desktop applications for standard window systems in multiple ways. First, input and output of these behaviors are always strokes. They receive the user's free stroke, and modify the member strokes of the segment. They do not render the surface by themselves, and segments paint the member strokes to ensure unified appearance. Next, These behaviors can be flexibly applied to and removed from the segment. Finally, different behaviors can be used in combination over time.

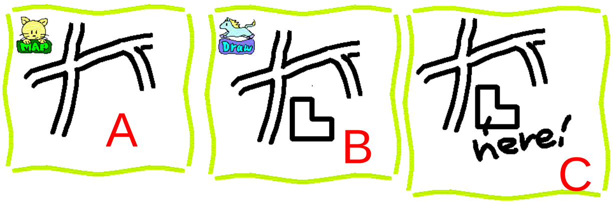

Figure 2. Combining behaviors

For example, in order to draw a map shown in Figure 2, the user draws roads using map behavior (A), draws the building using 2D geometric drawing behavior (B), and draws comments without any high level behaviors (C).

High level behaviors are represented as animal figures on the segments. These animal figures are useful to indicate the behavior in effect without spoiling the natural appearance of a whiteboard. Clicking on these animals shows behavior specific menus.

LONG TERM HISTORY MANAGEMENT

Another feature of Flatland is its automatic history maintenance mechanism. Every event on the surface is continuously recorded, and can be retrieved later. This mechanism frees users from explicit saving operation, which is not suitable for informal activities on whiteboards. Current implementation provides two interfaces for accessing automatically stored strokes and segments. First one is infinite undo and redo. Using undo and redo, the user can access to the any past status of the segment. Next one is time slider. Using the slider, the user can specify the time point directly. In addition, it provides jump buttons to get to discrete conspicuous time points.

SUMMARY

Flatland is an electric whiteboard to support daily activities in office environment. The demonstration introduces the basic functionality of Flatland, including screen real estate management, behaviors on the surface, and history management mechanism. Prototype system is written in Java and tested on Xerox Liveboards and front projected smart boards.

REFERENCES

1. Igarashi,T., Matsuoka, S., Kawachiya, S., Tanaka,H., Pegasus: A Drawing System for Rapid Geometric Design,CHI'98 summary, pp.24-25.
2. Prderson,E., McCall,K., Moran,T.P., Halasz,F., Tivoli: An electronic whiteboard for informal workgroup meetings, INTERCHI'93, pp,391-399.
3. Rekimoto,J., A Multiple Device Approach for Supporting Whiteboard-based Interactions, CHI'98.
4. Tapia,M.A., Kurtenbach,G., Some Design Refinements and Principles on the Appearance and Behavior of Marking Menus, UIST'95, pp.189-195.