Toward Universal Mobile Interaction for Shared Displays

Tim Paek, Maneesh Agrawala, Sumit Basu, Steve Drucker, Trausti Kristjansson,
Ron Logan, Kentaro Toyama, Andy Wilson
Microsoft Research
One Microsoft Way
Redmond, WA 98075
425.703.8647 USA

fluxmsr@microsoft.com

ABSTRACT

Researchers have noted conflicting trends in collaboration technologies between delivering more information on larger displays and exploiting mobility on smaller devices. Large, shared displays provide greater choice in the presentation of information, but mobile devices offer greater flexibility in the access of information. We describe a platform that leverages the best of both worlds by allowing multiple users to access and interact with a large, shared display using their own personal mobile devices, such as a cell phone, laptop, or wireless PDA. We highlight three applications built on top of the platform that demonstrate its generality and utility in a variety of group settings: namely, web browsing, polling, and entertainment.

Categories & Subject Descriptors: H.5.3 Group and Organizational Interfaces

General Terms: Design, Human Factors

Keywords: Mobile Devices, Platform, Shared Displays

1. INTRODUCTION

Researchers have noted two diverging trends in public, or shared computing scenarios [7]: One trend pushes toward larger, higher-resolution, singleton displays, as is the case with digital bulletin boards and kiosks. The use of these displays is usually distributed in time, and input is restricted to a single modality that is often physically bound to the display (e.g., touchscreens [14]). In contrast, the other trend is for mobile devices, such as cell phones, laptops, and wireless PDAs, to act as both input and display, where interaction is distributed in space among a number of users, allowing for synchronized collaboration and communication (e.g., virtual workspaces [13]). While shared displays typically offer greater computational power, more bandwidth, and larger presentation space, they frequently limit interaction to those who have direct access to the input devices (often just one user at a time). Mobile devices, on the other hand, disperse control and access to participating users, though limited computational power and smaller screen sizes often hinder dynamic interaction.

Connecting shared displays to mobile devices is an obvious way to leverage the best of both worlds, but tight synergy comes only when mobile device users can rely on the ubiquity of shared displays.

Fortunately, electronic shared displays are everywhere, though rarely given a second thought. We encounter them in living rooms as televisions, in meeting rooms as projection screens, in airports as flight-information kiosks, in movie theaters, in retail stores, and even in outdoors spaces, such as highway billboards and metropolitan marquees (e.g., New York's Times Square). What is missing from all of these shared displays is interactivity and computation. Even in cases where the displays offer interaction, only a single mode of input, such as the television remote control, is enabled as a means of interaction.

In this paper, we present a platform for generic, inexpensive, interactive shared displays that receives input from any mobile device. As a vital characteristic, the platform does not require additional hardware or software on existing mobile devices, and works without any changes in existing telecommunications infrastructure. Existing shared displays could be converted, one by one, into interactive shared displays, fully capitalizing on the prevalence of personal mobile devices. To demonstrate the generality and utility of the platform, we highlight three applications which were implemented with only UI modifications to the original application functionality.

2. RELATED WORK

Interactive shared displays are nothing new. An extensive body of research literature exists on the design and use of shared displays, focusing on particular applications (e.g., see [3],[11]). For example, researchers have explored how digital bulletin boards foster online and offline information sharing in an organization [2], how people move information between their PDAs and a PC [10] or public display [6], how a portal system accessible through a variety of mobile devices support work group activities [16], and so forth. This previous research exemplifies a variety of scenarios that could be implemented on an interactive shared display.

Outside of research, several large, interactive shared displays have also been deployed. For example, BBCi has built street-level window displays that allow passersby to not only see and hear interviews in progress, but also to submit their own questions using SMS text messaging on their cell phones [15]. Likewise, the Vodafone Lisbon office contains a giant cube display on
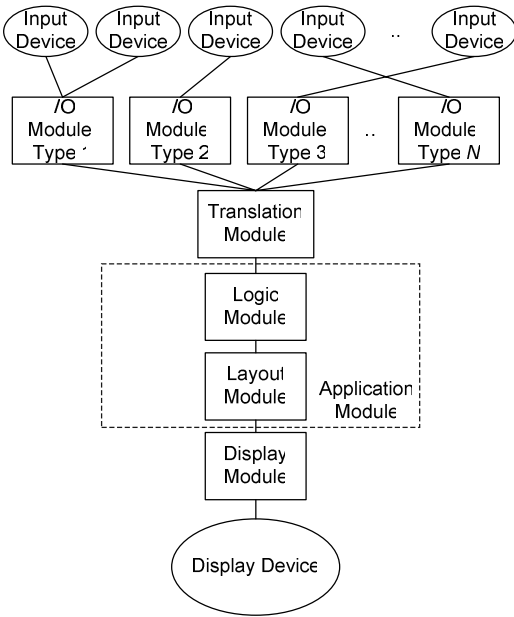
Figure 1. A modular architecture for shared displays.

which passersby can request news, short animations, and games, again using SMS text messaging [15]. Taking advantage of an even larger shared display, the art installation, La Bastille, transformed a library building into a giant video game console on which passersby could play Tetris using a radio-frequency video game controller [8]. Also converting a building into a large display, Blinkenlights allowed cell phone users to play Pong, and even to post emoticons [1].

While all of these aforementioned displays were custom-built for a single application and a single mode of input, we focus on a generalized platform for inexpensively converting any shared display into an interactive one. Our platform is modularized so that designers can, with minimal effort, port existing applications onto the interactive shared display, while accommodating multiple methods of input.

3. PLATFORM DESIGN
In order to encourage audience participation, a shared display must be capable of supporting multiple modes of input using a variety of devices. As shown in Figure 1, the platform adheres to a modular design in which diverse Input Devices send data to Input/Output (I/O) Modules, each of which is specifically designed to understand data from a single mode of communication. Examples of communication modes include e-mail, instant messaging, SMS text messaging, touch-tone dialing, voice, keyboard, mouse/joystick, camera-based input and photo output, video-camera input and video output, and so forth. We have implemented the first two thus far in our platform. Many of these modes are characterized by asynchronous, low-bandwidth communication with potentially high latency, a consideration which must be taken into account by any Application Module. Unlike synchronous architectures for collaboration (e.g., [12]), users are not necessarily required to maintain a connection to the

platform. It depends on the type of input mode. For example, Input Devices that are capable of sending email, such as a cell phone and laptop, send data packets to an I/O Module custom-built for email. Other I/O Modules, such as the one for handling instant messaging, are designed to support and maintain synchronous connections. Once any I/O Module receives data coming in through its input channel, it parses the data into discrete message units of content which are then passed to the Translation Module. For security purposes, protocols can be established to authenticate message units at this level.

The Translation Module takes these message units and converts them into commands or requests, sharing a common syntax, which can be processed by an Application Module. In particular, requests conform to a XML schema so that the complexities of getting from an Input Device to a request are hidden from the Application Module. As such, the platform allows a shared display to swap in Application Modules as desired.

The Application Module comprises a Logic Module and a Layout Module. The Logic Module consists of any application or set of applications which are running on the shared display. Software applications of just about any type can be created, as long as they account for the possibility of asynchronous inputs from multiple devices. The Logic Module outputs its data to a Layout Module, with optional auxiliary information which assigns data priorities, time limits, layout suggestions, and other hints as to how the data should be displayed. The Layout Module uses this information to generate appropriate graphics for the shared display. In other words, the Layout Module is where application designers specify how they would like their application rendered on specific types of displays.

Once a layout has been determined, the Display Module converts the layout along with any data into a form that can be readily displayed on the display device. The Display Module can support arbitrarily sophisticated levels of graphics depending on the capabilities of the display device, ranging from simple textual displays, HTML pages, to full 3D, dynamic elements combined with video.

The platform runs on Windows XP with the .NET Framework installed. Input Devices need not be a Windows client.

4. APPLICATIONS
The primary issues for an application written for a shared display, are that (1) it must be able to handle multiple, simultaneous input from users, (2) it must tolerate unpredictable lags in communication, and (3) it must keep input from users short, because rapid input on mobile devices can be cumbersome.

These constraints also suggest that interactive shared displays are most suited to certain types of applications. We identify four:

- *Jukebox applications* treat the display as a limited resource that must be shared (perhaps reluctantly) by users.

- *Collaboration tools* allow multiple people to contribute to a single goal. Surveys or cooperative games constitute examples.

- *Communication tools* facilitate communication between individual users; for example, bulletin boards.
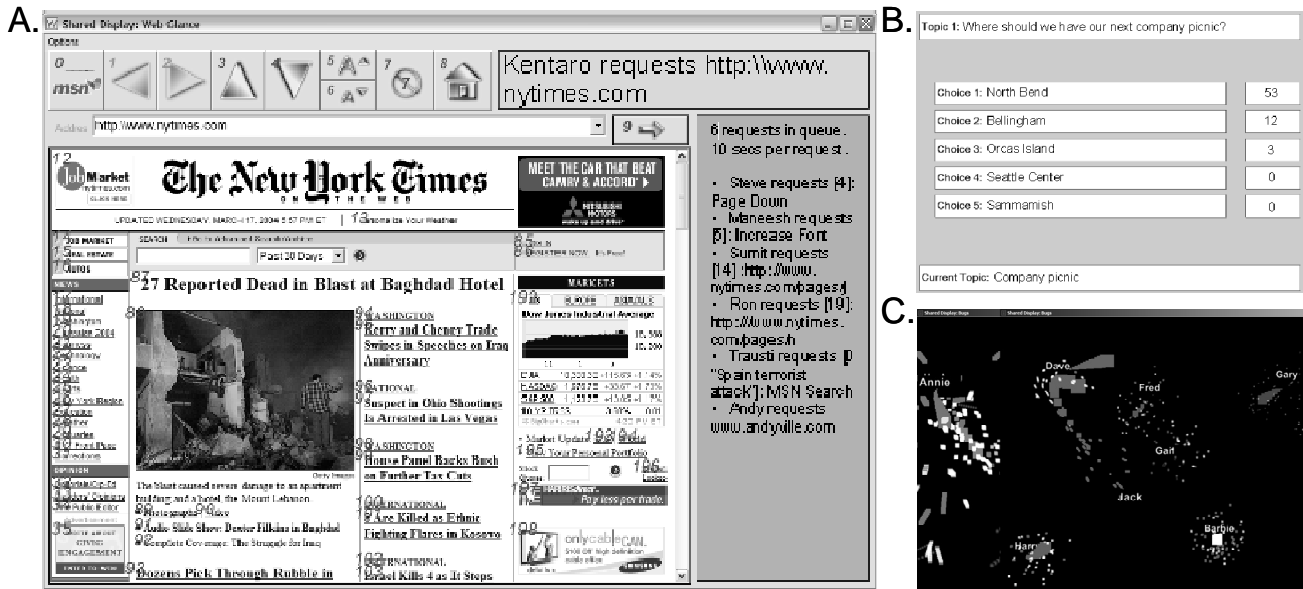
Figure 2. A. The *Web Glance* **browser for navigating the Internet on a shared display. B. A voting application for conducting polls. C.** *Bugs***, a 3D artistic expression display, where users can create creatures that interact with other creatures.**

■   *Arena applications* involve competitive interaction.

We highlight three implemented applications, which serve as exemplars of the first three categories. In our implementations, users interacted with the shared display on their mobile devices by sending brief e-mails to a specified address, or by engaging in an IM conversation with a specified account. Built on top of the platform, all three applications were independent and interchangeable in real-time. Only the Application Modules were swapped.

4.1 Web Browsing

Although most cell phones are equipped to allow users to browse the Internet, and most laptops and PDAs come with built-in wireless networking capabilities, many web sites are not designed for small displays; the affordances of a large, shared display can be useful, even if the display must be shared among several individuals. To support ad-hoc group web browsing on the platform, we built a web browser called *Web Glance* that was specifically designed for mobile input; that is, without the luxury of a keyboard or mouse. Figure 2(A) displays the Web Glance browser.

Given that mobile devices typically send low bandwidth messages, Web Glance facilitates quick and easy access to the functionalities of a web browser by exposing many relevant browser features as buttons on the user interface and labeling them with numbers. Furthermore, all the hyperlinks in a displayed HTML document are numbered as well. This technique is similar to the use of numeric mouse grids for allowing disabled users to navigate an operating system through speech (e.g., [5]). Figure 2(A) shows a user accessing the home page of the New York Times website. Sending the number '0' plus any keywords allows people to send the keywords to MSN Search. Sending any number between 1 and 9 executes a browser feature, such as

"Back," "Forward," "Font Increase," "Hide Numbers," and "Home." Sending any number shown on the web page sends a request for the URL associated with the numbered hyperlink. For example, sending a '12' on the shown web page directs the browser to navigate to the "Job Market" page of the web site. Since messages from mobile devices can be asynchronous with a potentially high latency, the numbering of the hyperlinks between displayed pages is consecutive up to a specified limit. Hence, if a user requests link '12' on the current page and the browser does not receive the request until a different page is being displayed, the number will still refer to the correct link. An optional auto-scroll feature allows entire web pages to be read without manual scrolling. Finally, sending a string by itself that is not preceded by a number defaults to loading that string as a URL. This allows users to simply request web pages, such as their own personal page, by URL.

In making group web browsing possible on a shared display, it cannot be assumed as is the case with most groupware technologies that users will treat the browser as a collaborative tool. In smaller group settings, a single person may take the lead in coordinating navigational choices into a joint decision. In a larger public setting, where users are among strangers, and collaboration is unlikely [4], it is incumbent upon the browser to manage non-collaborative and possibly competitive web page requests, rendering the browser a type of jukebox application. Web Glance accomplishes this in simple fashion by maintaining a queue of requests, and timing out the display of web pages. For example, the panel to the right of the web page in Figure 2(A) shows 6 requests currently in the queue. Users can also see who is making the request by viewing the top panel next to the "Home" button. Note that web pages are timed out unless there is no other request in the queue. This allows Web Glance to also

support the smaller group setting where navigation choices are collaboratively made as a joint decision.

4.2 Polling

In many situations where distinct assemblies of people are congregated before a shared display, it may be useful to conduct a poll for consensus building, group decision-making, or simply to gather locale-sensitive polling data. For example, movie screens could be used to conduct polls, such as whether or not a film was enjoyable. As an incentive, a lottery could notify random participants that they won a prize.

Inspired by this kind of group interaction, we built a polling application on which users could not only vote on whatever topic was currently displayed, but also introduce new topics for a requested period of time and a set of choices using a simple command language. Figure 2(B) shows the voting application for the topic "Where should we have our next company picnic?" Users can vote on this topic by sending the topic number and their choice number. For example, to vote for Orcas Island, a user would send '1' (the topic number) followed by '3' (for Orcas Island). Votes are instantly tallied and the statistics are updated. We suspect that a polling application for facilitating ad-hoc group voting would be extremely useful for promoting discussion and consensus building in workspaces.

4.3 Facilitating Real Interaction

Pet owners are familiar with the phenomenon where walking a pet sparks casual conversation with passersby, particularly other pet owners. As a communication tool, the *Bugs* application aspires to duplicate this effect using a virtual terrarium, possibly for use in pubs or cafes where customers often look for opportunistic interaction with other people. Public displays have been used elsewhere as a means for creating openings for conversation [9].

To engage in the Bugs environment, users send an eight-digit sequence of characters that serves as a "genome" for their virtual 3D bug. When the genome is received by the system, a graphical realization is generated and let loose as an autonomous creature in the environment. Figure 2(C) displays the Bugs environment. Each creature, labeled by its creator's name, swarms around the screen exhibiting traits determined by its genome, and upon encountering other creatures, engages in various behaviors, such as fighting, mating, or simply moving on. Interaction within the Bugs world serves as a talking point for interaction in the user world; if two bugs mingle and henceforth travel together, users may wonder which user possesses such a compatible genome. Finally, to simulate an ecosystem where creatures not only interact, but live and die in a potentially overcrowded environment, bugs maintain a lifespan range; for example, in Figure 2(C), "Annie" is undergoing an explosive death.

5. CONCLUSION

Motivated by the ubiquity of both large displays and mobile devices, we presented a platform that allows multiple users to access and interact with any enabled large, shared display, where individuals can use their own personal mobile devices as input devices, for whatever purpose the designers choose.

We described three implemented applications which were specifically designed to operate under the latency and bandwidth constraints that characterize mobile input: namely, Web Glance, a shared web browser which supports both collaborative and jukebox-style navigation of the Internet, a voting application for promoting informal collaboration and communication, and finally, Bugs, a virtual terrarium designed to entice users into casual conversation and provide a backdrop for entertainment.

As for future direction, usability studies need to be conducted for all of the implemented applications in both private and public spaces. Plans are underway to deploy the applications in the common areas of our corporation, such as the cafeteria or waiting rooms.

6. REFERENCES

[1] Blinkenlights, http://www.blinkenlights.de/

[2] Churchill, E. Nelson, L. and Denoue, L. (2003). Multimedia Fliers: Informal information sharing with digital community bulletin boards. *Communities and Technologies 2003*.

[3] *CSCW 2002 Workshop on Public, Community and Situated Displays*, November 16, New Orleans, http://www.appliancestudio.com/cscw/workshophome.htm

[4] Dourish, P. and Bly, S. (1992). Portholes: Supporting awareness in a distributed work group. *CHI-92*, 541-547.

[5] Dragon Naturally Speaking, http://www.scansoft.com/naturallyspeaking/

[6] Greenberg, S., Boyle, M. and LaBerge, J. (1999). PDAs and shared public displays: Making personal information public, and public information personal**.** *Personal Technologies*,3(1):55-64.

[7] Grudin, J., and Poltrock, S.E. (1997). Computer-supported cooperative work and groupware. In M. Zelkowitz (Ed.), *Advances in Computers Vol. 45*, pp.269-320. Orlando, FL: Academic Press.

[8] La Bastille, http://bastilleweb.techhouse.org/

[9] McCarthy, J. (2003). Using public displays to create conversation opportunities. *CSCW 2002 Workshop on Public, Community and Situated Displays*.

[10] Myers, B.A., Stiel, H., and Gargiulo, R. (1998). Collaboration using multiple PDAs connected to a PC. *CSCW '98*, 285-294.

[11] Ohara, K., Perry, M., Churchill, E., and Russell, D. (Eds.) (2003). *Public and situated displays: Social and interactional aspects of shared display technologies*. London: Kluwer.

[12] Patterson, J.F., Hill, R.D., Rohall, S.L., Meeks, W.S. (1990). Rendezvous: An architecture for synchronous multi-user applications. *CSCW `90*, 8-10.

[13] Pinelle, D., Dyck, J., and Gutwin, C. (2003). Aligning work practices and mobile technologies: Groupware design for loosely coupled mobile groups. *Mobile HCI 2003*, 177-192.

[14] Russell, D.M., Drews, C. and Sue, A. (2002). Social aspects of using large public interactive displays for collaboration. *Ubicomp 2002*, 229-236.

[15] Scanlon, J. (2003). If walls could talk, streets might join in. *New York Times*, September 18.

[16] Siirtola, H. and Heimonen, T. (2001). Scalable support for work groups and groupwork. *Mobile-HCI 2001*, 129-134.