A GENERAL FRAMEWORK FOR DEVELOPING MULTI-SURFACE ENVIRONMENTS

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Zheng Huang

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Program:
Software Engineering

August 2015

Fargo, North Dakota

North Dakota State University
Graduate School

Title

A General Framework for Developing Multi-Surface Environments

By

Zheng Huang

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Jun Kong
Chair

Dr. Changhui Yan

Dr. Yechun Wang

Approved:

8/10/2015
Date

Brian Slator
Department Chair

ABSTRACT

Since multi-touch technique has been used widely for personal device, such as smartphone, personal computer and tabletop, many forms of interaction are developed and served for user’s daily usage. But forms of interaction can be barely found between tabletop and smartphone. Therefore, we propose a general framework for facilitating developers and users to develop multi-surface applications. Our general framework uses the smartphone as input and output device to overcome limits and extends features of tabletop. We provide some applications based on treating smartphones as both input and output devices to demonstrate the implementation of our framework for smartphone on the tabletop.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Though electronic communication is developing fast, face-to-face meetings are still important [Wu03], particularly in collaborative applications. With a large screen, tabletop especially suits co-located group work. However, it is challenging to avoid interference during simultaneous interaction in a multi-user environment. For example, each tabletop is only equipped with one speaker, which inevitably causes interference if multiple users are play multimedia contents simultaneously. In addition, one single virtual keyboard prevents multiple users from engaging with data entry at the same time. For example, if user is using the virtual keyboard to type characters, the other users have to wait until the current user finishing typing. In order to solve these limitations completely, we choose personal smartphone as input and output device to offer an alternative way to interact with the tabletop. Though a large screen of a tabletop makes information accessible to multiple users, there are some problems remaining to undermine the multi-user interaction, such as content orientation or physical reachable distance [Shen06]. For instance, users are sitting around a tabletop. Consequently, the orientation of a digital document that is suitable for one side is not appropriate for the other side. Furthermore, the large size may give it hard time for a user who is seating over the reachable distance between the hand and the content.

A cross-device interaction that integrates a mobile device and a tabletop overcomes the interference issue by using a mobile device as an external controller, which produces multimodal feedback and supports touch-based gestures. Briefly speaking, a tabletop provides a public space for multiple users to browse contents while a mobile device extends the public space with multimodal feedback.

Previous studies had explored the benefits of combining mobile devices and a large display together and proposed intuitive gestures to support inter-device interaction. However, few researchers focused on developing a generic framework for supporting inter-device interaction.

Without losing generality, we implemented our framework taking advantage of these benefits. In order to make a selection by tapping object on the smartphone screen, we made smartphone screen “transparent” based on its movement on the tabletop. According to coordinate of tapping on the smartphone, tabletop decides to fire specific events and sends mobile UI components (table 1) messages to mobile. The smartphone receives UI components messages to build corresponding UI on the smartphone.

The remainder of the paper is organized as the following. Chapter 2 discusses related work. Chapter 3 overviews our framework. Chapter 4 provides a formal model to formalize inter-device interaction. Chapter 5 presents different applications based on our framework, followed by conclusion in Chapter 6.

RELATED WORK

Seminal works were proposed to support user-friendly inter-device interaction, such as augmenting a computer with PDAs in the single display groupware [Mye01], exchanging information between a personal device and a public display [Gre99], or offering a continuous workspace including personal computers and information wall/table displays [Rek99]. With diversified gesture modalities and multimedia feedback, modern mobile devices are natural to augment a large display for co-located group work in many different applications, such as oil and gas exploration [Sey13] or collaborative online shopping [Mut14].

One central theme in multi-device interaction is to minimize the effort of connecting two devices for sharing information. Based on different sensing techniques, various solutions have been proposed to address the device pairing issue. Those sensing techniques detected users' actions and accordingly derived users' intention to efficiently establish a virtual connection between devices without manually entering network addresses.

- **Touch-based sensing**. The stitching pen gesture dynamically forged a connection between two devices by continuously moving a pen from one device to another one [Hin04].
- **Radio based sensing**. Based on the radio frequency transponder sensing technology, ConnecTables combined two displays as a larger shared one when two devices moved close to each other [Tan01].
- **Accelerometer**. Accelerometer data are useful to derive users' actions, such as determining a synchronous gesture of bumping two tablets to set up a connection [Hin03] or recognizing a gesture pattern for authentication [Pat04]. Accelerometer is often combined with other sensing techniques. Toss-it integrated an accelerometer sensor

(i.e., detecting users' actions and estimating the strength or the trajectory of an action) with a camera (i.e., identifying the location of each user based on infrared LED markers) to determine a user's intention [Yat05]. PhoneTouch synchronized a touch event detected on an interactive surface with a bump event detected by an accelerometer sensor on the mobile device [Sch10]. The synchronized events are used to connect the mobile device with the surface. Hutama et al. [Hut11] correlate the angle of two touch points with accelerometer data to distinguish different mobile devices on a tabletop.

- **Acoustic sensing.** Point&Connect [Pen09] offered an intuitive gesture by pointing the source device to the intended target device. The intention of the user's gesture is derived by measuring the distance change based on acoustic signals.
- **NFC.** Touch & Interact used an NFC phone as a stylus to touch any position in a dynamic display to establish a connection [Har08].
- **Camera-based sensing.** Visual markers are used to identify objects or devices. For example, the SpotCodes system augmented a display with visual markers and a user aimed his/her phone at a visual tag on the display to select an object [Mad04]. Instead of using built-in cameras, some approaches explored the usage of an additional camera for detecting users' gestures [Lee08], recognizing infrared blinking signals to establish a connection between a mobile device and a table surface [Wil07] or verifying camera flashes to authenticate users [Sch08]. Rather than using a regular RGB camera, Ackad et al. used a depth camera to track users and developed a handshaking procedure based on color detection [Ack12]. In the multi-device interaction, consecutive color changes are used to transfer data between devices, such as FlashLight that illuminated the area on the

tabletop below a mobile device [Hes10], or C-Blink that produced color blinks from a mobile device [Miy04].

- **Synchronous action on buttons.** SyncTap synchronized button pressing and releasing events on two devices to connect them together [Rek03]. Similarly, the Touch-and-Connect framework [Iwa03] offered a two-button interface, which included a plug-button and a socket-button. Two devices are connected by first pressing the plug-button in the source device and then pressing the socket-button in the destination one.

Gesture interaction is intuitive and comfortable in the multi-device ecology [Kra10], and various gestures are proposed to support inter-device interaction, such as Scoop-and-Spread gesture [Aya00], the pen gesture of pick-and-drop [Rek97], the pouring gesture [Sey13], or throw and tilt interaction [Dac09]. McAdam and Brewster proved that using mobile devices together with a tabletop can sometimes produce better performance [Mca11]. With a built-in camera, it is natural to use a mobile device as a remote controller to select and manipulate objects in a distant display, such as sweep and point & shoot [Bal05], touch projector [Bor10], SnapAndGrab for content sharing [Mau08], and remote operation based on display registration [Pea09]. In addition, mobile device is useful to type in sensitive information for protecting privacy in a public environment [Luc08].

User-defined gestures are known to be easy to learn and are more preferred by users [Nac13]. Kray et al. [Kra10] systematically collected and analyzed a set of user-defined gestures in three conditions, i.e., phone-to-phone, phone-to-tabletop, and phone to public display. Focusing on tablet-sized devices, Kurdyukova et al. [Kur12] investigated three gesture modalities (i.e., multi-touch gestures, spatial gestures and direct contact gestures) in three conditions (i.e., iPad-iPad, iPad-tabletop, and iPad-public display).

The research about developing a generic framework that supports the design of cross-device interaction in various applications has not been well explored to date. Based on the PhoneTouch timing approach [Sch10], Schmidt et al. [Sch12] proposed a cross-device interaction style, in which a mobile device was used as a stylus to manipulate contents on a tabletop. However, this interaction style may associate a selected object with a wrong mobile device due to the collision of touch events [Sch12]. On the other hand, our approach gives each mobile device with a unique tag to assure a correct association between a user’s action and a mobile device. Furthermore, our approach solved the sensitivity issue of a tabletop. Due to the embedded infrared camera (i.e., the MS PixelSense technique), hand or finger movements, which may not even touch the surface, can accidentally trigger users’ actions on tabletop contents. Placing a mobile device on top of the tabletop not only tracks a user’s action but also prevents accidental actions caused by hand/finger movements. Recently, Roudaki et al. [Rou14a, Rou14b] implemented a bimanual cross-device interaction (called MobiSurf), in which the non-dominant hand performed a coarse-grained selection through a pointer while the dominant hand held the mobile device for a fine-grained interaction. However, pointers can easily get lost in reality and thus limit the applicability of MobiSurf. Instead, our approach eliminates pointers. In addition, our approach is featured with personalized touch-based gestures.

A GENERIC FRAMEWORK FOR CROSS-DEVICE INTERACTION

This paper proposes a generic framework that supports cross-device interaction among multiple users. During the interaction, a user can be in one of the following four states (See Table 1).

Table 1. Overview of our framework

User State	Description	User Actions
Initialization	Set up the communication between a tabletop and a mobile device	Place the compound NFC/Surface tag behind the mobile device.
Cross-Device interaction	Use a mobile device as an external controller	1. Place the mobile device on top of the tabletop. 2. The mobile device functions as a look-through mirror, which displays the information underneath the mobile device, during the movement. 3. The user taps the mobile device to select an object and lifts the mobile device from the tabletop. 4. Based on the user’s selection, the mobile device delivers multimodal feedback (such as vibration and speech) and the user uses touch-based gestures on the mobile device to remotely manipulate the selected object of the tabletop 5. The user’s input is sent back to the tabletop with visual feedback.
Tabletop and mobile sharing	Transfer data between a tabletop and a mobile device	Place the mobile device on top of the tabletop. The gesture of swiping away from the user triggers the data transfer from the mobile device to the tabletop, and vice versa.
Mobile and mobile sharing	Transfer data between two mobile devices	Place two mobile devices together and tap the shared object on the source device to trigger the data sharing

Initialization

In the beginning, a user is of the initialization state. In order to avoid tedious data entry on a mobile device, we use the NFC technique to automatically set up a wireless connection between a tabletop and a mobile device. More specifically, an NFC tag stores connection related information (e.g., the IP address of the tabletop or the user ID), and a connection is automatically set up when the NFC tag is moving close to the mobile device. In order to track each user's action, each user has a unique ID, which is defined through a visual tag. We stick the NFC and visual tags together, and the visual tag directly faces the tabletop screen so that the tabletop can read the visual tag. In summary, the initialization process sets up a communication between a tabletop and a mobile device without user input.

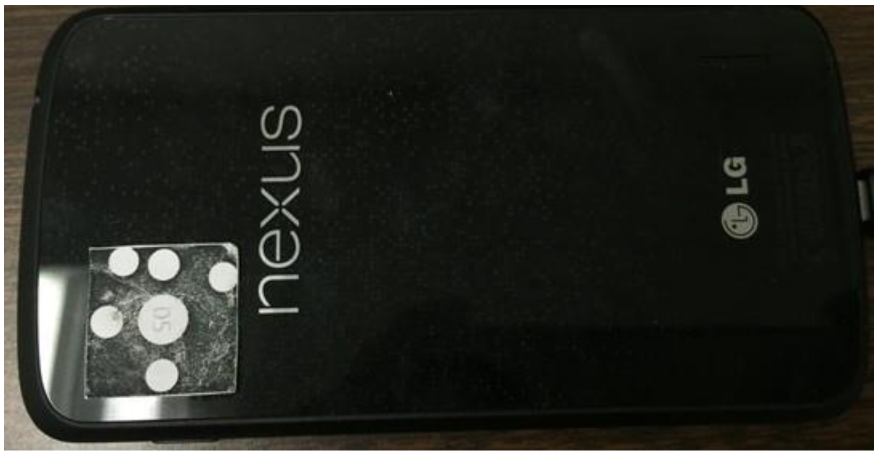


Figure 1. A compound tag sticking to the back of a smart phone

Cross-Device Interaction

In the cross-device interaction, the tabletop serves as a public display that is accessible to all users while a mobile device supplements a public display with a private working space. For collaborative applications that involve multiple users, it is critical to track each user's action, which is correctly transferred to the corresponding mobile device held by the user. Our approach attaches a unique visual tag to the back of each mobile device (See Figure 1). During the interaction, the mobile device is placed on top of the tabletop, and the built-in infrared camera of the tabletop continuously tracks the location of the mobile device. More specifically, the mobile device works like a look-through lens, which displays the content of tabletop underneath the mobile device. In order to select an object, the user moves the mobile device to the target object, and taps the target object on the mobile device. Lifting the mobile device off the tabletop screen actually triggers the selection. According to the application, a personal interface is present on the mobile device, which allows users to enter data and perceive multimodal feedback through the mobile device. Remotely manipulating tabletop contents on the mobile device avoids interference among multiple users and protects users' privacy. Distinct from existing techniques, our approach is featured by supporting personalized touch gestures on the mobile device. In order to remind a user about personalized gestures, tool tips are displayed in proximity to the selected object on the tabletop.

Tabletop and mobile sharing

Our framework supports user-friendly and efficient data transfer between a tabletop and a mobile device. When transferring data from a tabletop to a mobile device, the mobile device is placed on top of the tabletop and works as a look-through lens. A user presses his/her finger on the source file of the tabletop through the mobile device and then moves the mobile device towards the user to complete the data transfer. On the other hand, a user moves the mobile device away from the user to transfer data from a mobile device to the tabletop after selecting the source file of the mobile device.

Mobile and mobile sharing

Transferring data between two mobile devices is triggered by placing them back to back. The mobile sharing is implemented through the NFC technique…

In summary, our framework supports cross-device interaction for collaborative applications. Our approach has the following unique features.

- **Personalized gestures.** Several studies have reveals that the gestures that designers have proposed may not be consistent with users' expectation [Mos14, Sey13]. Personalized gestures enable adaptation to the individual preference. It has been justified that user-defined gestures achieved better memorability and higher user preference [Nac13]. Our approach uses the $1 recognizer algorithm [Wob07] to recognize a user's touch-based gesture on the mobile device and returns the recognized command to the tabletop. Since the recognition is implemented on the mobile device, it provides the flexibility to personalize application dependent gestures for each individual on his/her personal device.
- **Tooltips.** Gestures, especially application dependent ones, are not visible to users. For example, an oil and gas exploration application defined three gestures for data sharing, i.e., *pouring*, *flicking* and *camera* gestures [Sey13]. However, an early design critique revealed that the reaction to those gestures was not positive as users needed additional training to memorize those gestures [Sey13]. In order to facilitate users to recognize application-dependent gestures, our approach gives a tool tip that is displayed in proximity to the selected object on the tabletop. The content of the tooltip is adapted to the personalized gestures of each individual.
- **Thin Client**. In our approach, the mobile device interprets messages received from the tabletop to automatically display a personal interface and returns a user's input to the

tabletop. Since all application-related actions are actually performed at the tabletop, our framework implements an application-independent thin client, which allows interface developers to focus on the layout of UI components on the tabletop.

- **Prevent accidental actions.** The MS tabletop is equipped with infrared cameras, which can detect users' gestures even when users' fingers or hands do not touch the screen. The above sensing technique provides a rich design space, but it makes the sensing sensitive to recognize users' accidental actions. This issue is especially serious in a multi-user environment. When one user manipulates tabletop contents, the action of swiping off the screen by another user can interfere with the first user's operation. In our approach, each user uses his/her mobile device as a look-through lens to browse tabletop contents. The look-through lens not only provides the flexibility to manipulate tabletop contents, but also works as a shield to prevent accidental actions.

MODELING CROSS-DEVICE INTERACTION

Our framework is featured with cross-device interaction. We proposed a 5-state model to formalize the cross-device interactions. This state model facilitates interface developers to customizer our framework to develop a cross-device interface for a specific application.

In the beginning, a user is in the *offSurface* state, in which a user's mobile device is off the screen of a tabletop. In this state, a user simply uses his/her fingers to interact with the tabletop in a traditional manner. When a user places his/her mobile on top of the screen, the user moves to the *StaticOnSurface* state. A visual tag is attached to the back side of the mobile and closely touches the screen, which allows the built-in infrared camera to read the tag. As specified in Definition 1, each user has a unique tag to assure that the user's actions are correctly transferred to the corresponding mobile device.

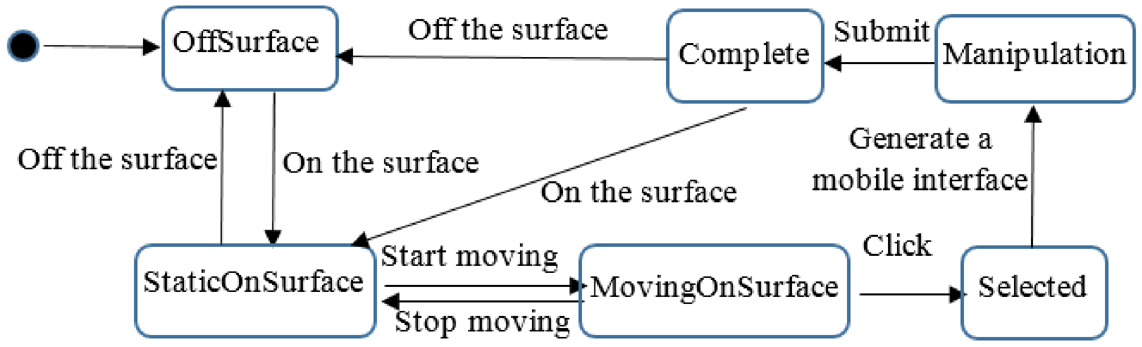


Figure 2. A 5-state model

The notations of TAGs and IDs represent the sets of tags and users, respectively.

Definition 1

Bijective function *pair*: TAGs → IDs defines an association between a visual tag and a user. Given a tag *t*, *pair(t)=u* if and only if visual tag *t* is paired with user *u*.

In summary, each user is identified with a unique tag. By placing a mobile device on a tabletop, our framework automatically registers the mobile and associates the user and the corresponding mobile together.

In order to browse the content, a user moves his/her mobile on top of the tabletop, which transits a user to the *MovingOnSurface* state. By default, the mobile device functions as look-through lens, which display the content beneath the mobile device. Such a design not only prevents accidental actions due to the sensitivity of multi-touch screens, but also provides the flexibility to provide personalized content of each individual user. More specifically, instead of delivering the identical content to all users, each mobile device personalizes information according to the user's personal interest while he/she is browsing the tabletop. For example, when users are browsing a map, someone is interested in restaurants while another is looking for gas stations. Accordingly, taking the tabletop as a public space for displaying the map, additional personalized information (such as restaurants or gas stations) overlays the public map on the mobile device. After a user moves the mobile to the destination, the user can select an object by clicking the object on the mobile touch screen, which moves the user to the *Selected* state. The notion of WIDGETs indicates the set of UI components in a tabletop application.

Definition 2

Partial and injective function *association*: WIDGETs → TAGs defines whether a GUI widget is selected by a user or not. If $w \in WIDGETs$, then we write *association* (w) ↓ and say that *association* (w) is defined to indicate that w is in the domain of *association*. If w is not in the domain of *association*, we write *association* (w) ↑ and say that *association* (w) is undefined. A GUI widget w is selected by a user if and only if *association* (w) ↓.

Formally speaking, a user *u* is in the *Selected* state if and only if $\exists w \in WIDGETs$, pair°association(w)=u.

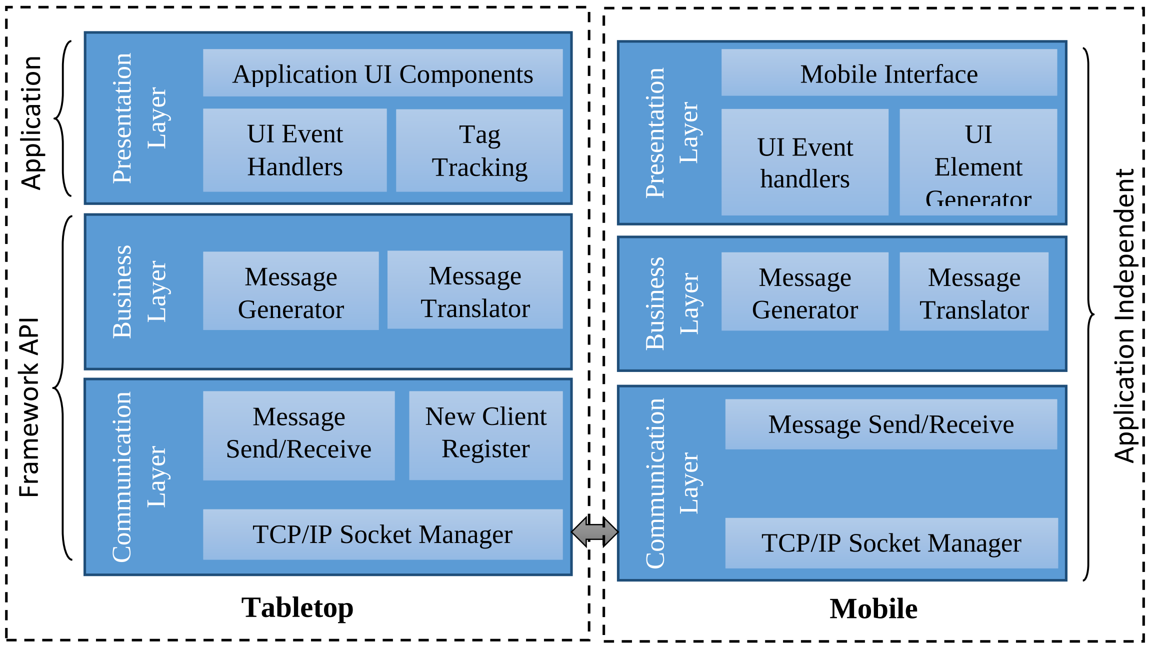


Figure 3. System architecture

Our framework implements an event architecture to support cross-device interaction (See Figure 3), and serves as a middleware to set up two-way communications between a tabletop and a mobile device. Based on a user's selection on a tabletop application, our framework generates a message, which defines a mobile interface and is forwarded to the associated mobile device. According to the received message, the associated mobile device automatically generates a mobile interface (such as displaying a button) that allows the user to manipulate the selected object through the mobile device. Upon the user's input, the tabletop application is updated accordingly. Such an event-driven design achieves a thin client which makes the mobile side application-independent, and thus significantly reduces the development effort. From the perspective of interface designers, they only need to define the inter-device interaction of each UI components and the spatial relations of those UI components on the tabletop, while our framework can handle the underlying communications. In other words, based on our framework, one critical task is to specify a mobile interface that responds to a user's action of clicking an object on the tabletop through his/her mobile device.

Table 2. Mobile UI components

Mobile UI Components	**Description**
speech	Speak a text on mobile device
Lighting	Flash the LED light
Beep	Generate a beep sound
TextMode	Display a textbox on the mobile device
Text	Display textual contents on the mobile device
Image	Show an image on the mobile device
Media	Play a voice or video file
AlertDialogue	Show a text alert message on mobile device
webLink	Open a web page on the mobile device4
vibration	Vibrate the mobile device
button	Display the button on the mobile device
editText	Display the editText on the mobile device
drawing	Display the customize drawing view on the mobile device
linearLayout	Add a layout that arranges its children in a single column or a single row on the mobile device.
Dropdown list	Add a spinner which provide a quick way to select one value from a set on the mobile device.

The notation of MobileUIs indicates the set of mobile UI components that are supported in our framework.

Definition 3

Total function *map*: WIDGETs $\rightarrow 2^{\text{MobileUIs}}$ defines the mobile UI components that are displayed on the mobile device when a tabletop UI component *w* is selected. Given a GUI widget *w*, map(*w*)=Ø if and only if *w* does not need cross-device interaction.

Based on Definition 3, a mobile interface is displayed on the associated mobile device, which triggers the state transition to the *manipulation* state. For example, the tabletop application has a login interface, which includes a UI component, called *password*, for a user's input. In order to protect a user's privacy, we can define a mobile interface in the *map* function, i.e., map(password)={speech, vibration, editText}. Figure 4 illustrates a sample user case in which our framework creates a private, text-entry interface on a mobile device. The sequence of the control flow is described as follows: 1) The user selects the *password* edit field in a login interface on the tabletop through a mobile device; 2) our framework raises the "ObjectSelect" event; 3) The event handler in the tabletop application defines a mobile interface that includes "Speech", "Vibration", and "editText"; 4) our framework packs the message and sends it to the mobile application through the network; 5) the mobile application receives the message, processes it, and produces the corresponding mobile interface; 6) The user types required information on the mobile device; 7) the mobile application sends the user's input back to our framework on the tabletop; 8) the framework API receives the message and raises the "returnValue" event; 9) the tabletop application handles the "returnValue" event and updates the tabletop application accordingly.

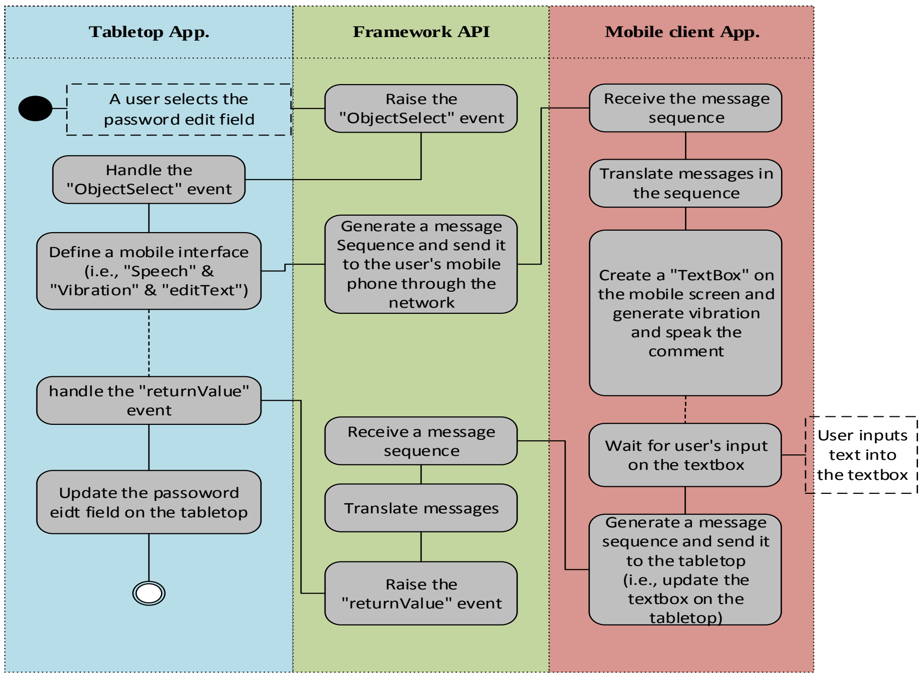

Tabletop App.
Framework API
Mobile client App.
A user selects the password edit field
Raise the "ObjectSelect" event
Receive the message sequence
Handle the "ObjectSelect" event
Translate messages in the sequence
Define a mobile interface (i.e., "Speech" & "Vibration" & "editText")
Generate a message Sequence and send it to the user's mobile phone through the network
Create a "TextBox" on the mobile screen and generate vibration and speak the comment
handle the "returnValue" event
Receive a message sequence
Wait for user's input on the textbox
User inputs text into the textbox
Update the passoword eidt field on the tabletop
Translate messages
Generate a message sequence and send it to the tabletop (i.e., update the textbox on the tabletop)
Raise the "returnValue" event

Figure 4. Message passing

MULTI-DEVICE INTERACTION

Table 3. Inter-device interaction technique

Issue	Technique
Data Transfer	**Tabletop-to-Mobile Transfer:** Transfer data from a tabletop to a mobile device
	Mobile-to-Tabletop Transfer: Transfer data from a mobile device to a tabletop
	Mobile-to-Mobile Transfer: Transfer data between two mobile devices
Personalization	**Autofill:** Make the personal data stored in a mobile device accessible to applications on the tabletop
	Content Customization: Personalize information on a mobile device
	PhoneMap: Overlay personalized digital information on a map through a mobile device
User Interface Composition	**Extended Screen:** Use a mobile device as an extended screen to issue commands
Authentication	**Access Control:** Provide a lightweight access control to lock or unlock tabletop content
	Password: Input the password on the mobile device
	LockPattern: Use touch-based gesture on a mobile device to authenticate users
Localized & Private Feedback	**Multimodal feedback:** Provide multimodal feedback on mobile devices
Input Expressiveness	**PhoneGesture:** Use touch-based gestures on a mobile device to issue commands
	Personalized Gesture (include tooltips)

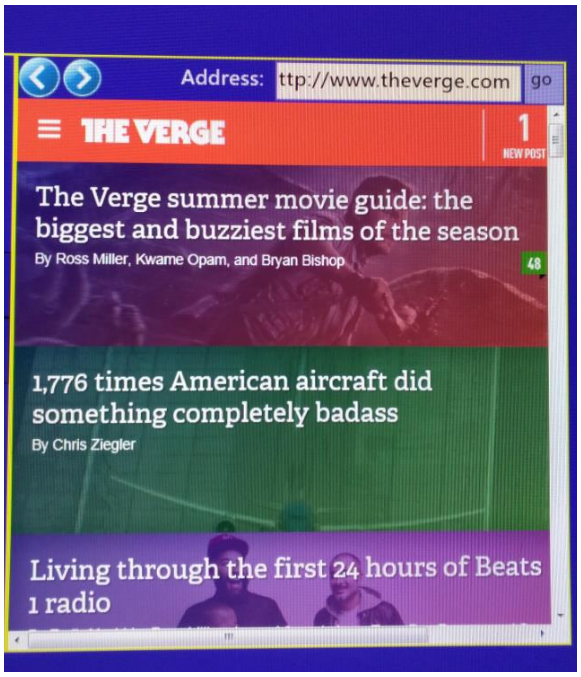
Personalization

Being a personal device, a smartphone stores much personal information, such as the contacts, browsing history, bookmarks and etc. By identifying each user with a unique tag, our framework can efficiently share personal information between a tabletop and a mobile device. We implemented three technique

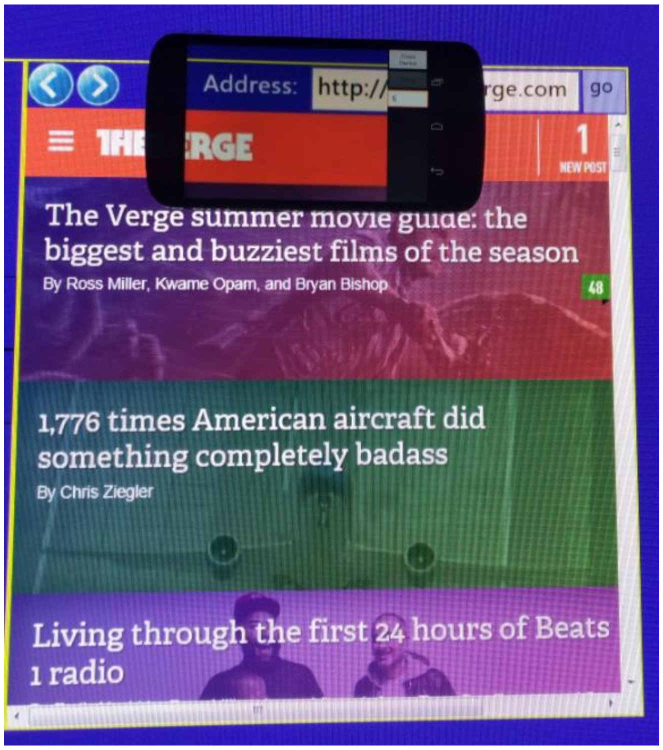
s to demonstrate personalization, i.e. autofill, content customization and PhoneMap.

Autofill

The autofill technique explores the feature of sharing personal information from a mobile device to a tabletop. Since the tabletop represents a public space which makes it inapplicable to store personal information in order to protect privacy, a user has to manually type in a web URL even when the web site was already visited by the user in the past. Manual input is error prone and time consuming, particularly when the URL link is long. In the above example, the autofill technique allows a user to choose a web URL from his/her browsing history saved in the mobile device and immediately displays the selected web site on the tabletop. Using selection to replace free-style typing potentially reduces the error rate [Shn09]. We demonstrate the autofill technique in Figure 5. Figure 5(a) shows the initial user interface after a user starts a web browser on the tabletop. By placing a mobile device on top of the tabletop, the mobile device functions as a look-through lens, which displays the tabletop content beneath the mobile device, as shown in Figure 5(b). Once the user taps the address bar on the smartphone screen (See Figure 5(c)), our framework automatically produces a dropdown list on the mobile device by defining a map between the address bar and the dropdown list according to Definition 3. The user can choose a bookmark from the dropdown list on the mobile device, as presented in Figure 5(d). Finally, the tabletop browser displays the website the user chooses (See Figure 5(e)).

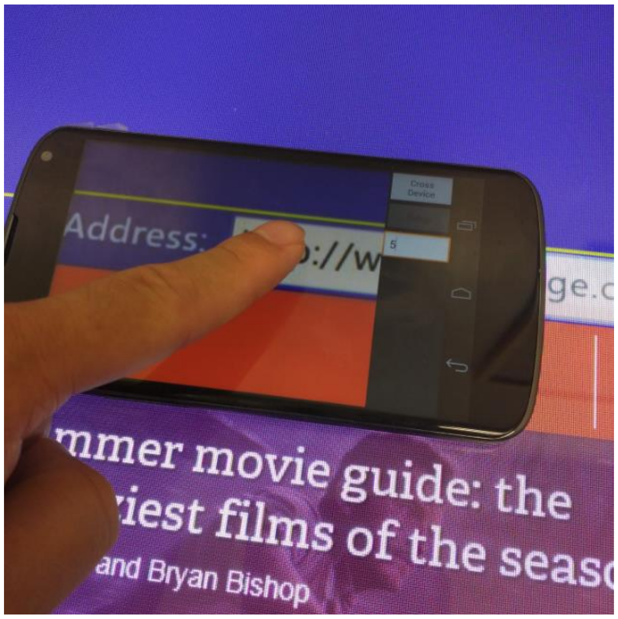

Address:
ttp://www.theverge.com
go
THE VERGE
1 NEW POST
The Verge summer movie guide: the biggest and buzziest films of the season
By Ross Miller, Kwame Opam, and Bryan Bishop
48
1,776 times American aircraft did something completely badass
By Chris Ziegler
Living through the first 24 hours of Beats 1 radio

(a) Initial user interface on the tabletop

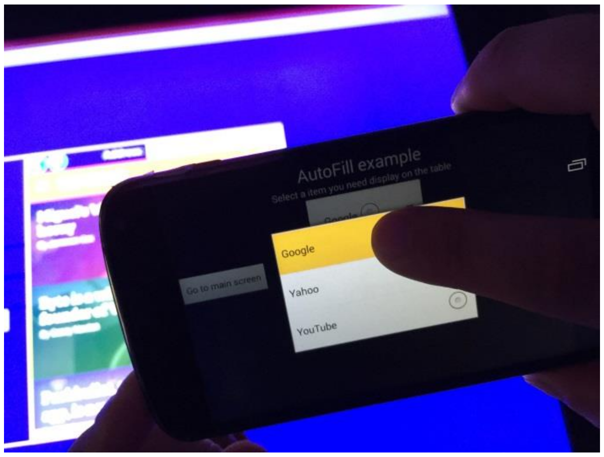

Address:
http://
rge.com
go
THE RGE
1 NEW POST
The Verge summer movie guide: the biggest and buzziest films of the season
By Ross Miller, Kwame Opam, and Bryan Bishop
48
1,776 times American aircraft did something completely badass
By Chris Ziegler
Living through the first 24 hours of Beats 1 radio

(b) Moving the mobile device on top of the tabletop

Figure 5. Autofill

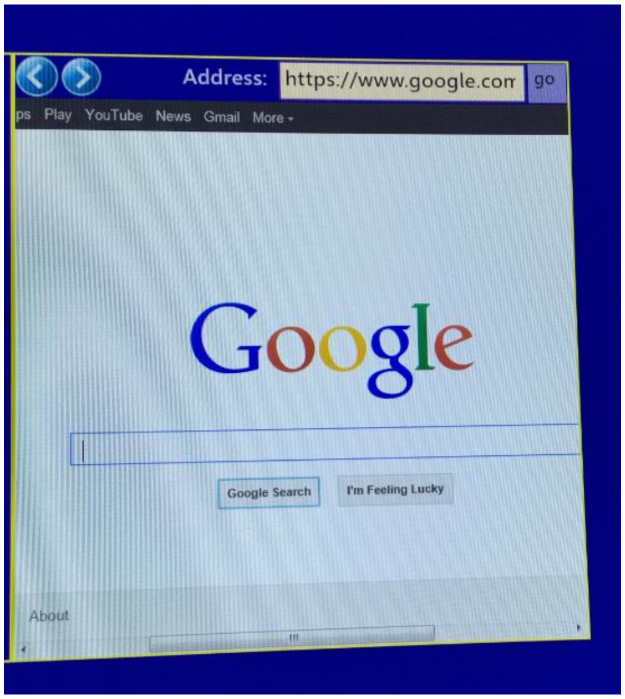


(c) Tapping the address bar on the mobile device



(d) Selecting a URL from a dropdown list on the mobile device

Figure 5. Autofill (continued)

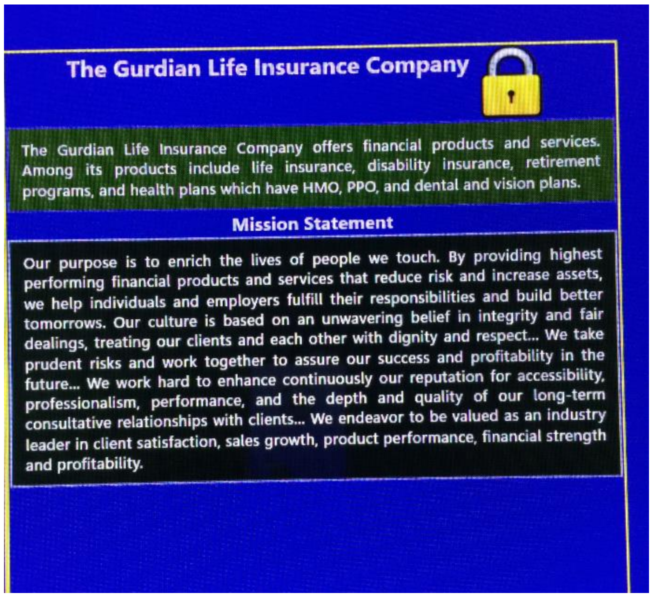


(e) Content updated on the tabletop

Figure 5. Autofill (continued)

Content Customization

In co-located work, though users have a common goal, each one can have a unique personal preference due to background, skill level, culture and etc. Our framework extends a tabletop with multiple mobile devices, which provide personalized information to fit each user's need. For instance, a tabletop is placed in the lobby of an international company to introduce the company (See Figure 6(a)). Since potential customers may have different interests, it is desirable to supplement the public tabletop display with a mobile device. The supplementation is especially useful in a multi-user environment since the customization on a personal mobile device does not interfere with other users. In the above example, we translate sentences in English to a different language that the user prefers to. Figure 6(b) shows the screenshot of moving a mobile device to the destination and selecting a paragraph for reading. Then, the user uses his mobile device to translate the selected paragraph from English to a different language (See Figure 6(c)).



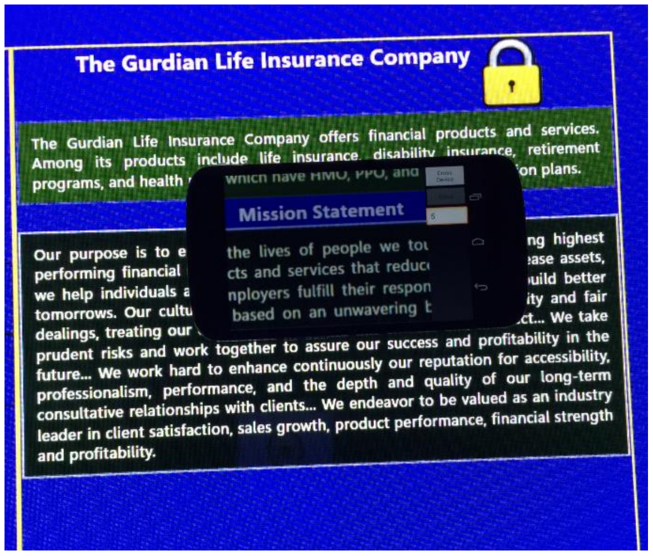
The Gurdian Life Insurance Company

The Gurdian Life Insurance Company offers financial products and services. Among its products include life insurance, disability insurance, retirement programs, and health plans which have HMO, PPO, and dental and vision plans.

Mission Statement

Our purpose is to enrich the lives of people we touch. By providing highest performing financial products and services that reduce risk and increase assets, we help individuals and employers fulfill their responsibilities and build better tomorrows. Our culture is based on an unwavering belief in integrity and fair dealings, treating our clients and each other with dignity and respect... We take prudent risks and work together to assure our success and profitability in the future... We work hard to enhance continuously our reputation for accessibility, professionalism, performance, and the depth and quality of our long-term consultative relationships with clients... We endeavor to be valued as an industry leader in client satisfaction, sales growth, product performance, financial strength and profitability.

(a). initial screen



The Gurdian Life Insurance Company

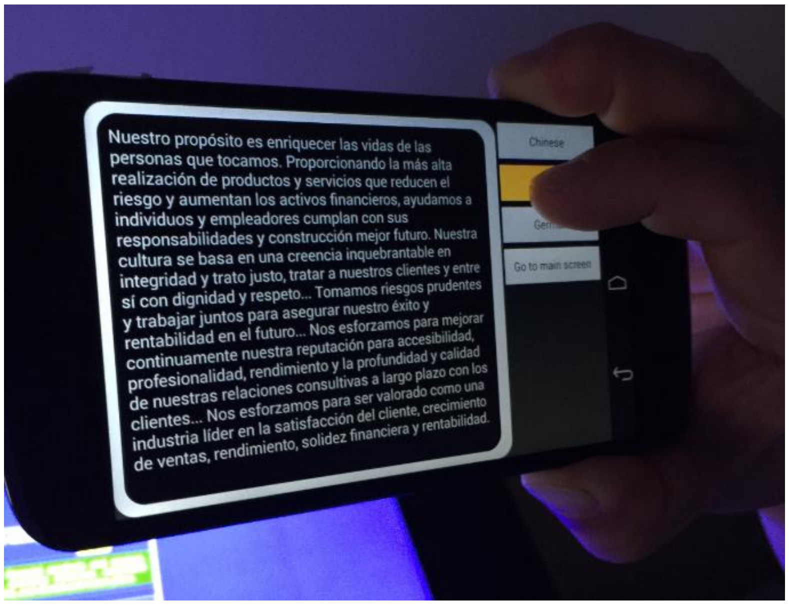
The Gurdian Life Insurance Company offers financial products and services. Among its products include life insurance, disability insurance, retirement programs, and health which have HMO, PPO, and on plans.

Mission Statement

Our purpose is to e the lives of people we tou ng highest performing financial cts and services that reduc ease assets, we help individuals a nployers fulfill their respon uild better tomorrows. Our cultu based on an unwavering b ity and fair dealings, treating our ct... We take prudent risks and work together to assure our success and profitability in the future... We work hard to enhance continuously our reputation for accessibility, professionalism, performance, and the depth and quality of our long-term consultative relationships with clients... We endeavor to be valued as an industry leader in client satisfaction, sales growth, product performance, financial strength and profitability.

(b). Moving the smartphone to the content for customization

Figure 6. Content customization

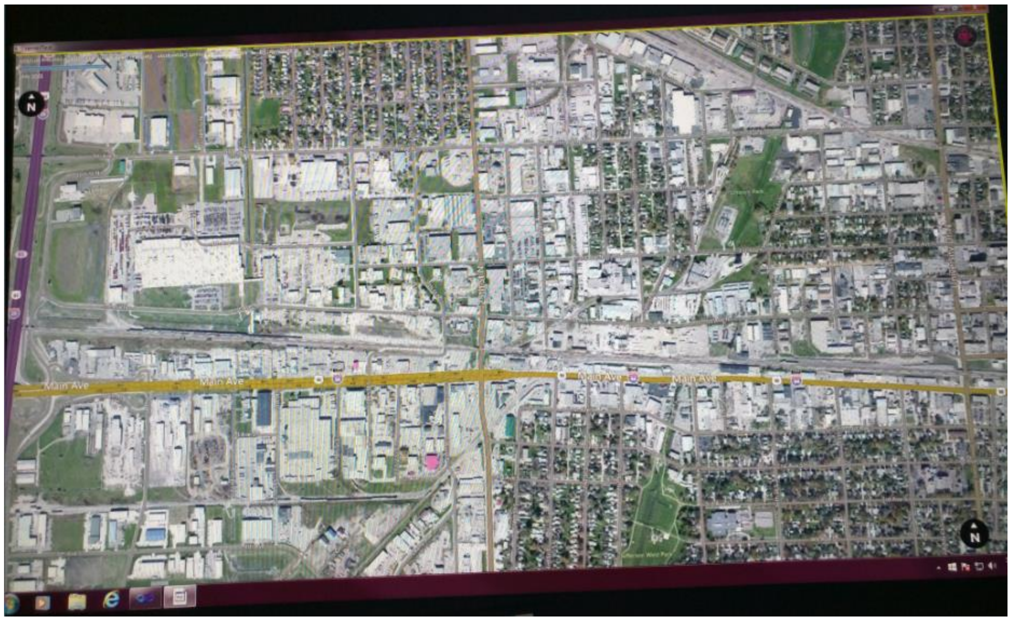

Nuestro propósito es enriquecer las vidas de las personas que tocamos. Proporcionando la más alta realización de productos y servicios que reducen el riesgo y aumentan los activos financieros, ayudamos a individuos y empleadores cumplan con sus responsabilidades y construcción mejor futuro. Nuestra cultura se basa en una creencia inquebrantable en integridad y trato justo, tratar a nuestros clientes y entre sí con dignidad y respeto... Tomamos riesgos prudentes y trabajar juntos para asegurar nuestro éxito y rentabilidad en el futuro... Nos esforzamos para mejorar continuamente nuestra reputación para accesibilidad, profesionalidad, rendimiento y la profundidad y calidad de nuestras relaciones consultivas a largo plazo con los clientes... Nos esforzamos para ser valorado como una industria líder en la satisfacción del cliente, crecimiento de ventas, rendimiento, solidez financiera y rentabilidad.

(c). The mobile interface on the smartphone

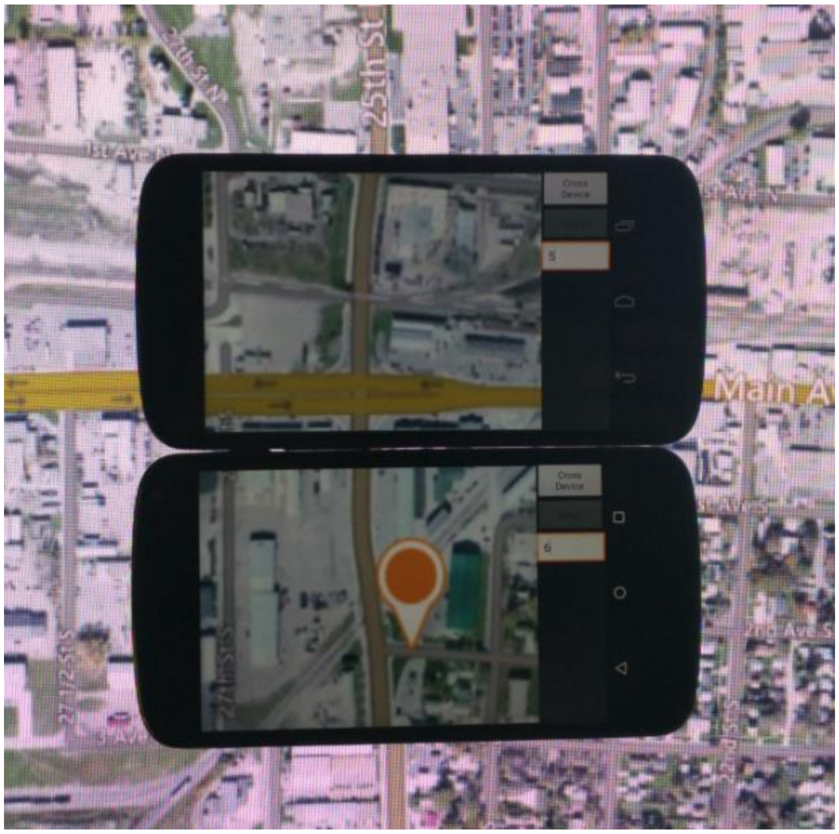
Figure 6. Content customization (Continued)

PhoneMap

Our framework is distinct by browsing and manipulating tabletop contents through a mobile device. By default, the mobile device displays the content beneath the mobile device when a user is moving his/her mobile device on top of a tabletop. This design provides a flexible design space to overlay personalized digital information on the mobile device when browsing the tabletop. Based on our framework, we implemented a PhoneMap technique, which allows multiple users to browse a map simultaneously while each one can view personal points of interest through his/her mobile device without interrupting others. For example, during map browsing, one user is looking for restaurants while another one is interested in gas stations. Corresponding to the map displayed on a tabletop in Figure 7(a), the top mobile device shows the default map while the bottom one overlays a personal point of interest on the default map in Figure 7(b).



(a). A map on a tabletop



(b) Personal points of interest

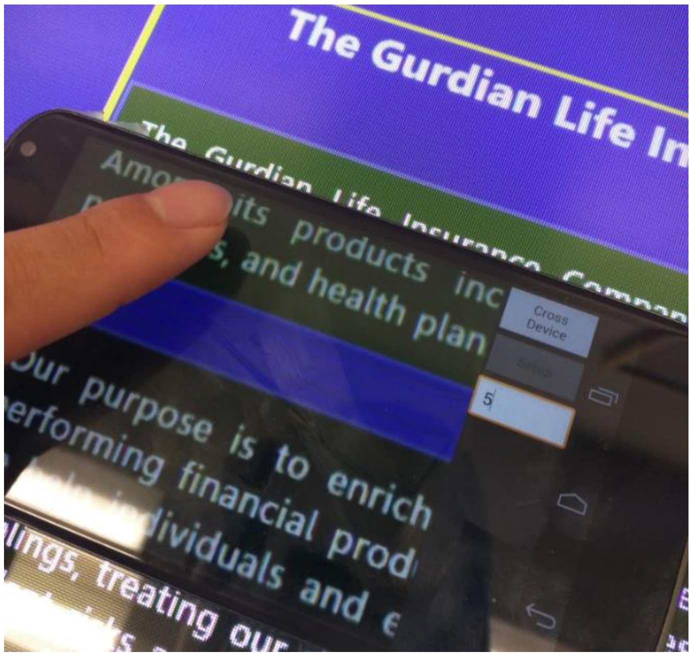
Figure 7. PhoneMap

User Interface Composition

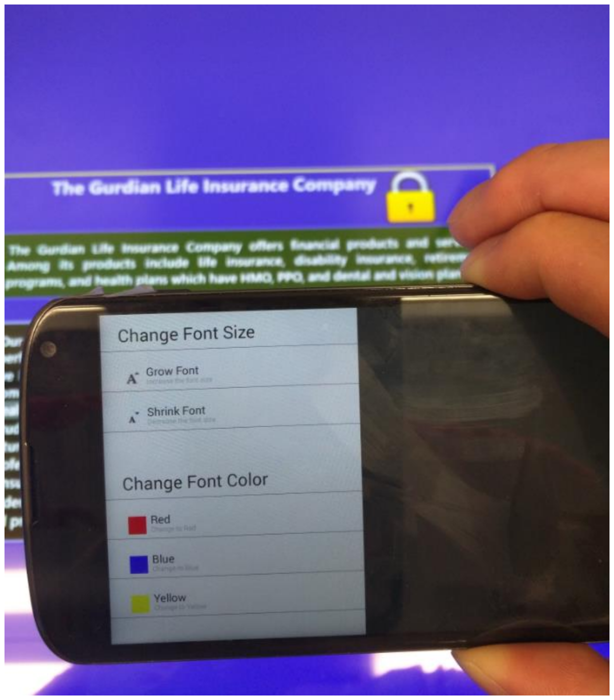
From the hardware perspective, a tabletop provides a large screen for sharing information. However, traditional interfaces are not suitable for multi-user simultaneous interaction. For example, if a user taps an action menu to issue a command on a specific component of a tabletop application, other users cannot issue the same command to other components. Furthermore, a tabletop only provides a virtual keyboard, which prevents users from editing different components simultaneously on a tabletop. Our framework extends a tabletop by moving commands to a mobile device so that each user can issue a command independently and simultaneously.

Extended Screen

The *extended screen* technique provides an action menu on a mobile device to manipulate a selected UI component on a tabletop. Figure 8(a) shows that a user selected a paragraph for editing through his/her mobile device. The selection triggers to generate an action menu on the associated mobile device (See Figure 8(b)). Using the mobile interface, the user can edit the selected paragraph through his/her mobile device while other users can edit other components simultaneously without interfering with each other. After the user taps the desirable command on the mobile device (See Figure 8(c)), the contents on the tabletop is updated accordingly (See Figure 8(d)). In summary, the extended screen technique takes a mobile device as an external controller to manipulate tabletop contents without interfering with each other..

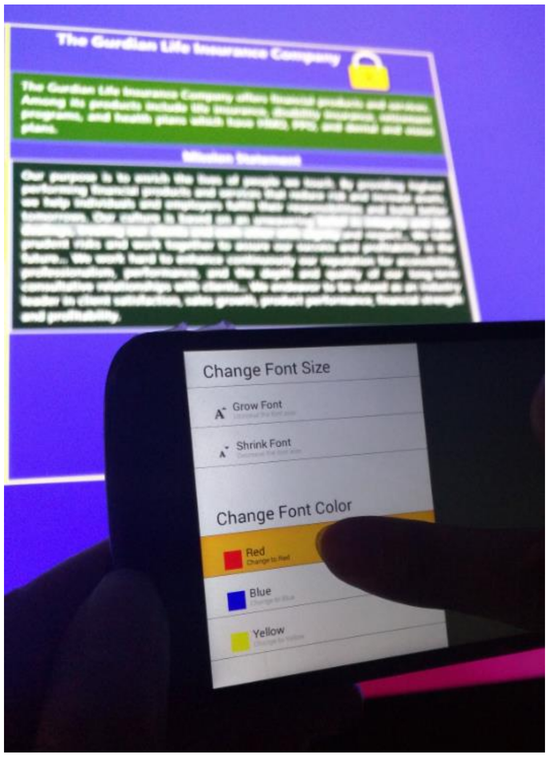


(a) Selecting a paragraph through a mobile device

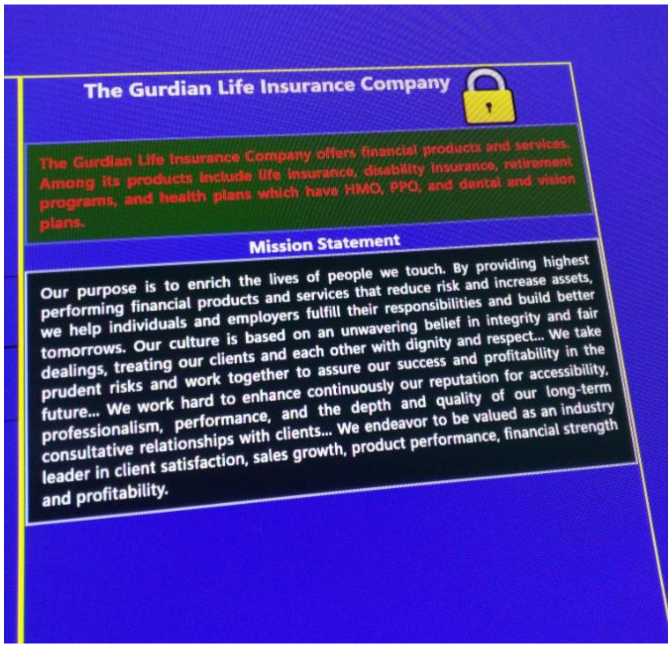


(b) Displaying an action menu on a mobile device

Figure 8. Extended screen



(c) Editing the selected paragraph



The Gurdian Life Insurance Company

The Gurdian Life Insurance Company offers financial products and services. Among its products include life insurance, disability insurance, retirement programs, and health plans which have HMO, PPO, and dental and vision plans.

Mission Statement

Our purpose is to enrich the lives of people we touch. By providing highest performing financial products and services that reduce risk and increase assets, we help individuals and employers fulfill their responsibilities and build better tomorrows. Our culture is based on an unwavering belief in integrity and fair dealings, treating our clients and each other with dignity and respect... We take prudent risks and work together to assure our success and profitability in the future... We work hard to enhance continuously our reputation for accessibility, professionalism, performance, and the depth and quality of our long-term consultative relationships with clients... We endeavor to be valued as an industry leader in client satisfaction, sales growth, product performance, financial strength and profitability.

(d) Updating the tabletop accordingly

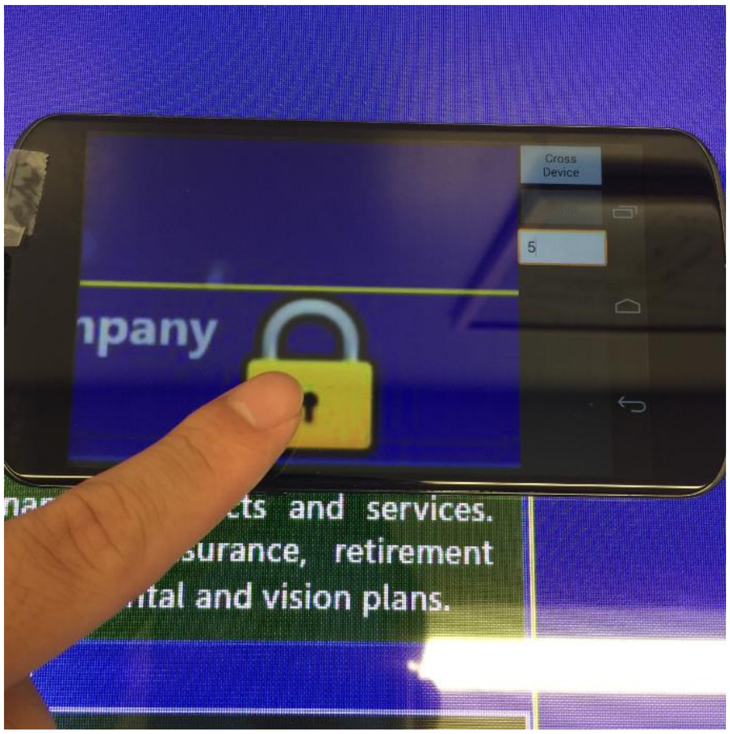
Figure 8. Extended screen (Continued)

Authentication

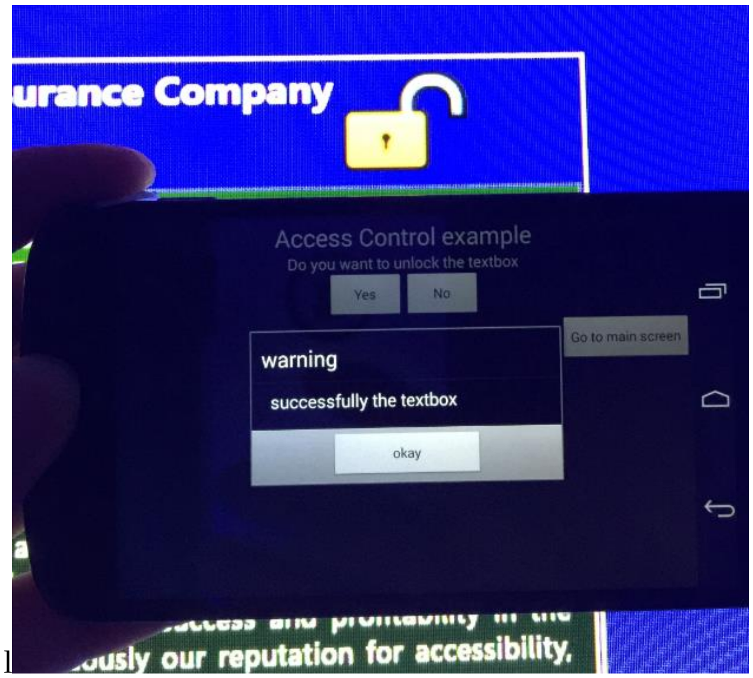
Authentication is a serious problem in front of a large shared screen. Due to privacy, users do not desire to expose their personal information on a shared screen, such as username and password. In addition, a shared screen requires a flexible access control mechanism so that a component is locked when a user uses the component and is unlocked when a user releases it.

Access Control

In a multi-user environment, mutual exclusion has to be enforced so that only one user can modify a piece of information at one time. Based on our framework, we implemented a flexible access control mechanism on a tabletop. Figure 9(a) shows that a user locked a paragraph by tapping the lock button through a mobile device. Then, a confirmation is displayed on the mobile device and the tabletop is updated with an unlock icon (See Figure 9(b)). The lock assures that only the user who successfully gets the lock can modify the locked paragraph. If another user wants to modify the locked paragraph, an error message is displayed in Figure 9(c). The locked paragraph can be modified by other users until it is unlocked by the user who locked the paragraph.

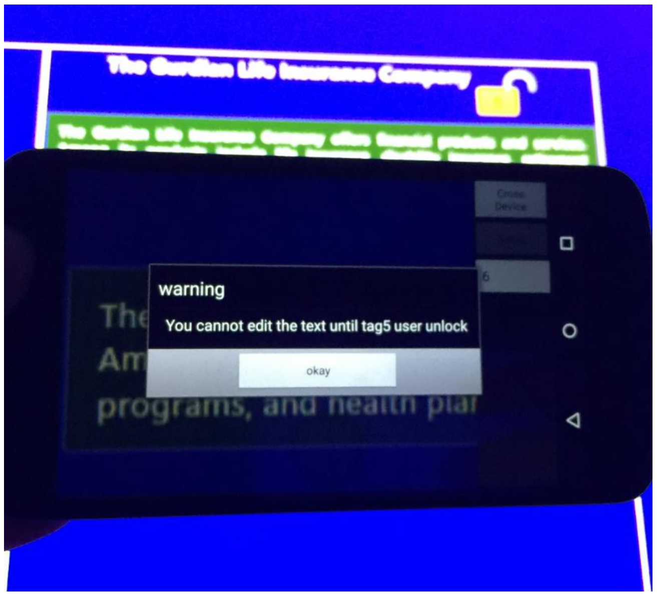


(a) Locking a paragraph



(b) A successful lock

Figure 9. Access control

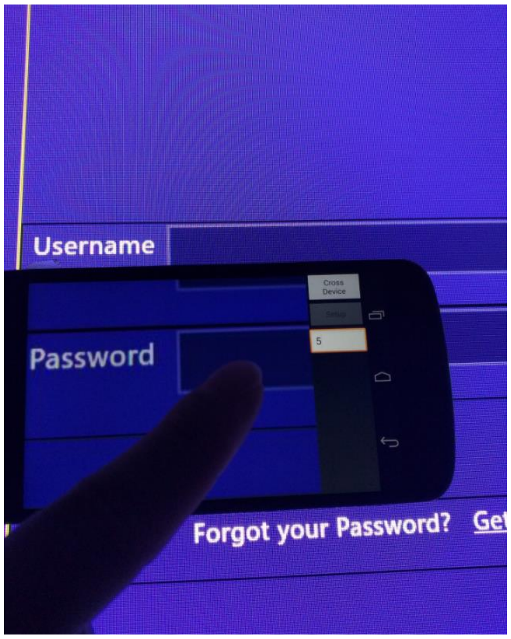

The Guardian Life Insurance Company
warning
You cannot edit the text until tag5 user unlock
okay

(c) A locked paragraph cannot be accessed by other users

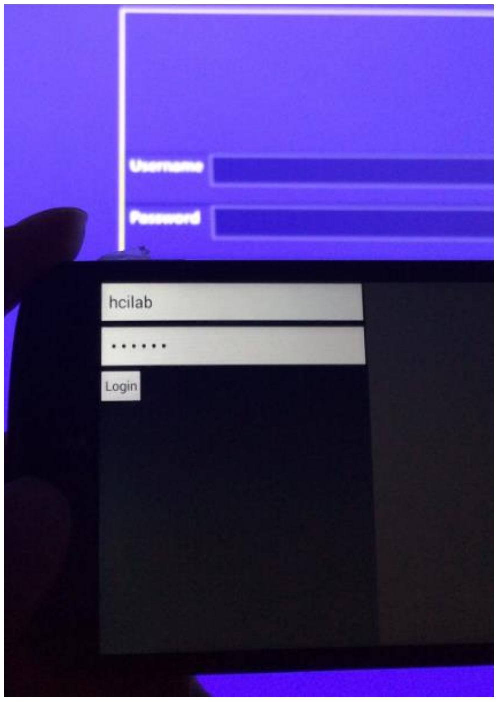
Figure 9. Access control (Continued)

Password

In a public environment, anyone can view the information typed through a virtual keyboard on a large screen. Our framework enhances the security by typing a password through a mobile device. Figure 10(a) shows that a user tapped a login form. Instead of typing the password directly on the tabletop, our framework generates a login form on the mobile device so that the user can complete the login process on this/her mobile device to protect privacy, as shown in Figure 10(b).



(a) Tapping a login form

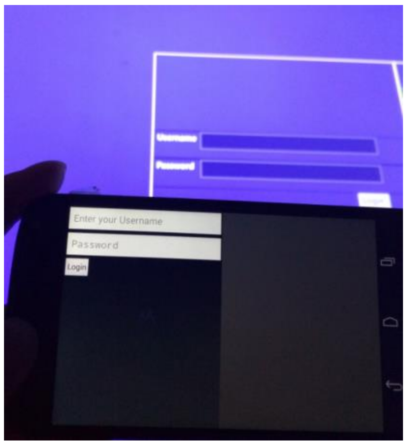


(b) Completing the login on a mobile device

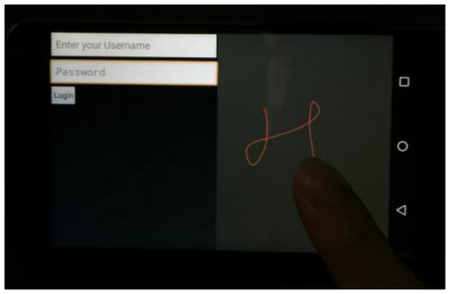
Figure 10. Password protection

LockPattern

Free style typing on a virtual keyboard is inefficient and error-prone. Therefore, new techniques have been proposed to authenticate a user's identity on mobile devices, such as fingerprint or Android password pattern. Our framework implemented a LockPattern technique, which allows a user to complete the login process through drawing instead of typing. After a user tapped a login form (See Figure 10(a), a drawing panel (corresponding to the drawing component in Table 1), which is displayed beside a traditional login form (See Figure 11(a)), avoids typing a password. Instead, the user can draw a predefined sketch to verify the identity (See Figure 11(b)). The predefined sketch replaces the user's password and is stored in the user's own personal smartphone. The sketch is meaningless to others except the user himself/herself.



(a) A drawing panel on the mobile device



(b) Replacing free-style typing with drawing

Figure 11. LockPattern

Private Feedback

Normally, a shared tabletop can be used by multi-user. There is only one screen, one virtual keyboard, and one speaker. Once two or more users did different operations, system may give wrong feedbacks to the users. For example, users cannot play the different audio or video at same time on the tabletop. Therefore, we introduce multimodal feedback technique in our framework.

Multimodal feedback

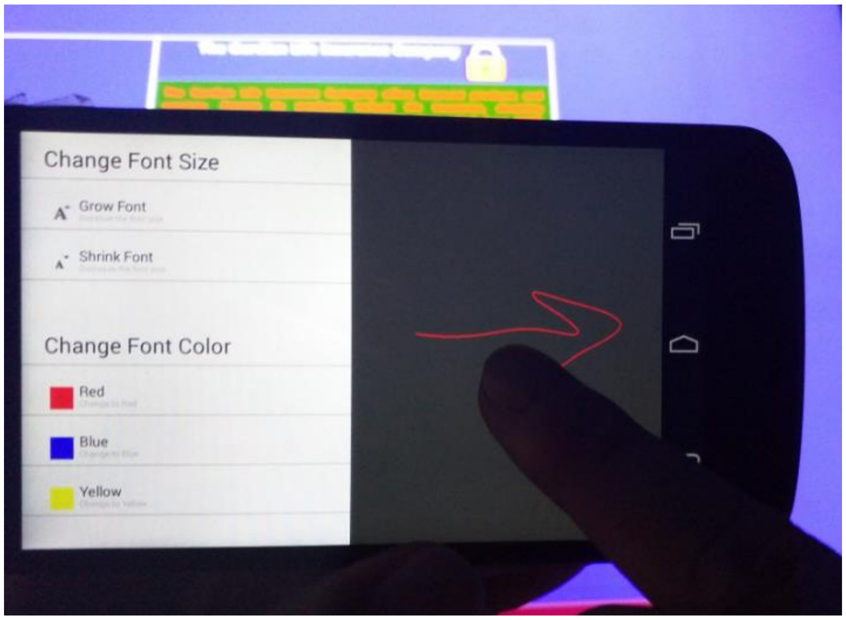
Multimodal feedback technique implements mobile UI components (table 1) for activating by touch-based event. Developer can build mobile UI components in order to match different event. For example, Figure 10 (a) shows that the user taps a login from on the smartphone, which fires an event on the tabletop. Tabletop sends corresponding mobile UI components messages to the smartphone based on result of event. If the user taps a right object (the login form Figure 10 (a)), smartphone receive the message about creating linearLayout, editText, button, speech message Figure 10 (b). Multimodal feedback technique do not avoid interference between multiple users, but it also helps developers easily build the unique UI on smartphone in the tabletop application side based on requirement.

Input Expressiveness

Gestures are the most popular interaction method on devices with a touch screen. Our framework is featured by supporting personalized gestures. In other words, each user can define his/her own set of gestures. Personalized gestures have been proven to increase the memorability and user preference [Nac13].

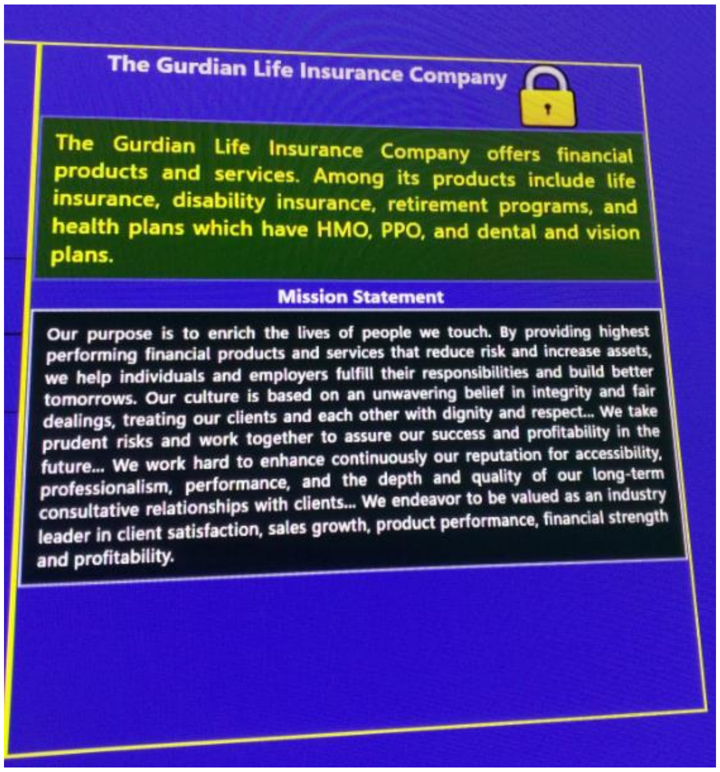
PhoneGesture

Gestures potentially solve the fat finger error when tapping a button on a small touch screen. The PhoneGesture technique provides a drawing panel on the mobile device and allows users to issue a command by drawing a gesture instead of tapping a button. We extended the example discussed in the Figure 8 with a set of gesture commands. After a user selected a paragraph on the tabletop, a mobile interface is displayed on the associated mobile device, as shown in Figure 12(a). The mobile interface includes two areas, the left area with an action menu and the right one with a drawing panel. A user can draw a gesture within the right area. Specifically speaking, an arrow gesture indicates the command of growing font size. Accordingly, the tabletop is updated after the user issues this gesture (as shown in Figure 12(b)).



(a) Drawing a gesture to issue a command

Figure 12. Gesture based interaction



The Gurdian Life Insurance Company

The Gurdian Life Insurance Company offers financial products and services. Among its products include life insurance, disability insurance, retirement programs, and health plans which have HMO, PPO, and dental and vision plans.

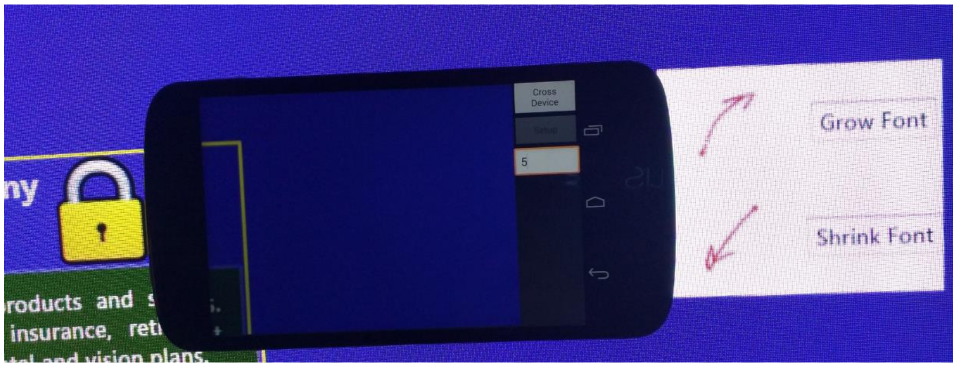
Mission Statement

Our purpose is to enrich the lives of people we touch. By providing highest performing financial products and services that reduce risk and increase assets, we help individuals and employers fulfill their responsibilities and build better tomorrows. Our culture is based on an unwavering belief in integrity and fair dealings, treating our clients and each other with dignity and respect... We take prudent risks and work together to assure our success and profitability in the future... We work hard to enhance continuously our reputation for accessibility, professionalism, performance, and the depth and quality of our long-term consultative relationships with clients... We endeavor to be valued as an industry leader in client satisfaction, sales growth, product performance, financial strength and profitability.

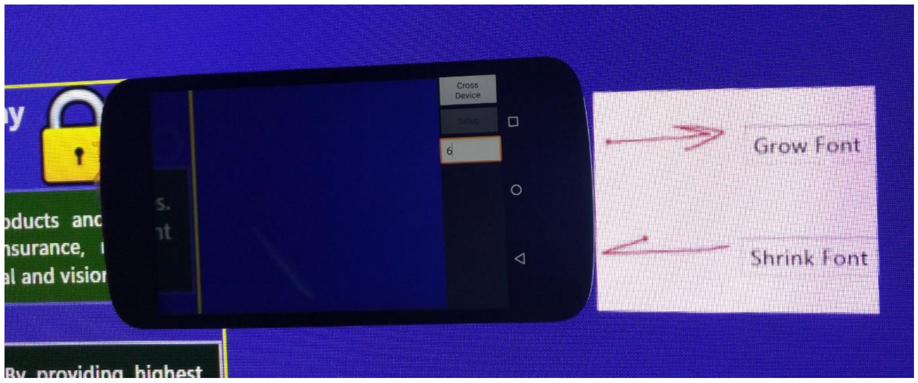
(b) Updating the tabletop accordingly

Figure 12. Gesture based interaction (Continued)

In order to cater to universal usability, experienced users prefer using a highly customizable system. For example, the Dolphin browser allows users to define their personalized gesture to open favorite websites. Our framework introduces personalized gestures to a multi-device interaction environment. In other words, each user can create his/her personal set of gestures. By recognizing the ID of a mobile device through a unique tag, our framework can appropriately determine the semantics of an input gesture. Furthermore, our framework can facilitate novel users to memorize personalized gestures by providing tooltips. Figure 13 presents the tool tips of personalized gestures, corresponding to two individual users.



(a) User 1’s gestures



(b) User 2’s gestures

Figure 13. Personalized gestures and tool tips

CONCLUSION

Our work provides fluid and seamless interaction between personal smartphones and a shared tabletop. Especially, we implemented a generic framework that facilitates interface developers to develop inter-device interactions. In summary, our contributions are summarized as the following: (1) implementing a smartphone as a look-through lens when browsing tabletop contents; (2) extending a tabletop through a mobile device with multimodal feedback; and (3) supporting customizable gestures-based commands. Based on our framework, we implemented various applications to justify the feasibility of our framework.

Our future work will focus on evaluating the usability of our framework with real-world applications.

REFERENCES

[Ack12] C. J. Ackad, A. Clayphan, R. M. Maldonado, and J. Kay, "Seamless and Continuous User Identification for Interactive Tabletops Using Personal Device Handshaking and Body Tracking", *Proc. CHI EA,* 2012.

[Aya00] Y. Ayatsuka, N. Matsushita, and J. Rekimoto, "HyperPalette: a Hybrid Computing Environment for Small Computing Devices", *Proc. CHI '00 Extended Abstracts on Human Factors in Computing Systems*, pp.133-134, 2000.

[Bal05] R. Ballagas, M. Rohs, and J. Borchers, "Sweep and Point & Shoot: Phonecam-based Interactions for Large Public Displays", *Proc. CHI EX. Abstracts*, 2005.

[Bor10] S. Boring, D. Baur, A. Butz, S. Gustafson, and P. Baudisch, "Touch Projector: Mobile Interaction Through Video", *Proc. CHI'10*, 2010.

[Dac09] R. Dachselt and R. Buchholz, "Natural throw and tilt interaction between mobile phones and distant displays", *CHI Ext. Abstracts*, 2009.

[Gre99] S. Greenberg, M. Boyle and J. Laberge, "PDAs and Shared Public Displays: Making Personal Information Public, and Public Information Personal", *Pers. and Ubiq. Comp., vol. 3,* p. 54–64, 1999.

[Har08] R. Hardy and E. Rukzio, "Touch & Interact: Touch-based Interaction of Mobile Phones with Displays", *Proc. MobileHCI'08,* pp.245-254, 2008.

[Hes10] T. Hesselmann, N. Henze and S. Boll, "FlashLight: Optical Communication between Mobile Phones and Interactive Tabletops", *Proc. ITS,* 2010.

[Hin03] K. Hinckley, “Synchronous Gestures for Multiple Persons and Computers”, *Proc. UITS’03*, pp.149-158, 2003.

[Hin04] K. Hinckley, G. Ramos, F. Guimbretiere, P. Baudisch, and M. Smith, “Stitching: Pen Gestures that Span multiple Displays”, *Proc. AVI’04*, 2004.

[Hut11] W. Hutama, P. Song, C. Fu and W. Goh, "Distinguishing Multiple Smart-Phone Interactions on a Multi-touch Wall Display using Tilt Correlation", *Proc. CHI*, 2011.

[Iwa03] Y. Iwasaki, N. Kawaguchi, and Y. Inagaki, “Touch-and-Connect: A Connection Request Framework for Ad-hoc Networks and the Pervasive Computing Environment”, *Proc. PerCom’03*, 2003.

[Kra10] C. Kray, D. Nesbitt, J. Dawson, and M. Rohs, “User-Defined Gestures for Connecting Mobile Phones, Public Displays, and Tabletops”, *Proc. MobileHCI’10*, pp.239-248, 2010.

[Kur12] E. Kurdyukova, M. Redlin and E. Andre, “Studying User-defined iPad Gestures for Interaction in Multi-display Environment”, *Proc. IUI’12*, pp.93-96, 2012.

[Lee08] H. Lee, H. Jeong, J. Lee, K. W. Yeom, H. J. Shin, and J. H. Park, “Select-and-Point: A Novel Interface for Multi-Device Connection and Control Based on Simple hand Gestures”, *Proc. CHI '08 Extended Abstracts on Human Factors in Computing Systems*, pp.3357-3362, 2008.

[Luc08] A. De Luca and B. Frauendienst, “A privacy-respectful input method for public terminals", Proc. NordiCHI, 2008.

[Mad04] A. Madhavapeddy, D. Scott, R. Sharp, and E. Upton, “Using Camera-Phones to Enhance Human-Computer Interaction”, *Proc. UbiComp’04, 2004.*

[Mar11] N. Marquardt, R. Jota, S. Greenberg, and J. Jorge, "The Continuous Interaction Space: Interaction Techniques Unifying Touch and Gesture on and above a Digital Surface", *Proc. the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part III*, pp.461-476, 2011.

[Mau08] A.J. Maunder, G. Marsden and R. Harper, "SnapAndGrab – Accessing and sharing contextual multi-media content using Bluetooth enabled cameraphones and large situated displays", *Proc. CHI, 2008.*

[Mca11] C. McAdam and S. Brewster, "Using Mobile Phones to Interact with Tabletop Computers", *Proc. ITS'11, 2011.*

[Miy04] K. Miyaoku, S. Higashino, and Y. Tonomura, "C-Blink: A hue-difference-based light signal marker for large screen interaction via any mobile terminal", *Proc. UIST, 2004.*

[Mos14] M. Mostafapour and M. Hancock, "Exploring Narrative Gestures on Digital Surfaces", *Proc. ITS 2014,* pp.5-14, 2014.

[Mut14] M. Muta, K. Mukai, R. Toumoto, M. Okuzono, J. Hoshino, H. Hirano, and S. Masuko, "Cyber Chamber: Multiuser Collaborative Assistance System for Online Shopping", *Proc. ITS'14, pp.289-294, 2014.*

[Mye01] B. A. Myers, "Using handhelds and PCs together", Comm. ACM, 44:34–41, 2001.

[Nac13] M. A. Nacenta, Y. Kamber, Y. Qiang, and P. O. Kristensson, "Memorability of Pre-designed and user-defined Gesture Sets", *Proc. CHI'2013, 2013.*

[Pat04] S. N. Patel, J. S. Pierce and G. D. Abowd, "A gesture-based authentication scheme for untrusted public terminals", *Proc. UIST, 2004.*

[Pea09] N. Pears, D.G. Jackson, and P. Oliver, "Smart phone interactions with registered displays"*, IEEE Perv. Comp., 8:14–21, 2009.*

[Pen09] C. Y. Peng, G. B. Shen, Y. G. Zhang, and S. W. Lu, "Point&Connect: Interaction-based Device Pairing for Mobile Phone Users", *Proc. the 7th international conference on Mobile systems, applications, and services,* pp.137-150, 2009.

[Rek97] J. Rekimoto, "Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments", *Proc. UIST'97,* pp.31-39, 1997.

[Rek99] J. Rekimoto and M. Saitoh, "Augmented Surfaces: A spatially Continuous Work Space for Hybrid Computing Environments", *Proc. CHI'99,* pp.378-385, 1999.

[Rek03] J. Rekimoto, Y. Ayatsuka, and M. Kohno, "SyncTap: An Interaction Technique for Mobile Networking", *Proc. 5th International Symposium on Mobile HCI,* LNCS 2795, pp.104-115, 2003.

[Rou14a] A. Roudaki, J. Kong, and G. Walia, "A Framework for Bimanual Inter-Device Interactions", *Proc. DMS, 2014*

[Rou14b] A. Roudaki, J. Kong, G. Walia, and Z. Huang, "A Framework for Bimanual Inter-Device Interactions", *Journal of Visual Languages and Computing, 25(6), 727-737, 2014 (An extension of the DMS conference publication).*

[Sch08] J. Schoning, M. Rohs, and A. Kruger, "Using Mobile Phones to Spontaneously Authenticate and Interact with Multi-Touch Surfaces", Proc. Workshop on Designing Multi-Touch Interaction Techniques for Coupled Private and Public Displays, 2008.

[Sch10] D. Schmidt, F. Chehimi, E. Rukzio and H. Gellersen, "PhoneTouch: A Technique for Direct Phone Interaction on Surfaces", *Proc. UIST, pp.13-16, 2010.*

[Sch12] D. Schmidt, J. Seifert, E. Rukzio and H. Gellersen, "A Cross-Device Interaction Style for Mobiles and Surfaces", *Proc. DIS, 2012.*

[Sey13] T. Seyed, M. C. Sousa, F. Maurer, and A. Tang, “SkyHunter: A Multi-Surface Environment for Supporting Oil and Gas Exploration”, *Proc. ITS’13, pp.15-22, 2013.*

[Shn09] B. Shneiderman and C. Plaisant, “Designing the user interface: strategies for effective human-computer interaction”, *Addison Wesley, 2009.*

[Tan01] P. Tandler, T. Prante, C. Muller-Tomfelde, N. Streitz, and R. Steinmetz, “ConnecTables: Dynamic Coupling of Displays for the Flexible Creation of Shared Workspaces”, *Proc. UIST’01, pp.11-20, 2001.*

[Wil07] A. Wilson and R. Sarin, “BlueTable: Connecting Wireless Mobile Devices on Interactive Surfaces Using Vision-Based Handshaking”, *Proc. Graphics Interface 2007, pp.119-125, 2007.*

[Wob07] J. O. Wobbrock, A. D. Wilson, and Y. Li, “Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes”, *Proc. the ACM Symposium on User Interface Software and Technology, pp.159-168, 2007.*

[Wu03] M. Wu and R. Balakrishnan, “Multi-Finger and Whole Hand Gestural Interaction Techniques for Multi-User Tabletop Displays”, *Proc. UIST’03, pp.193-202, 2003.*

[Yat05] K. Yatani, K. Tamura, K. Hiroki, M. Sugimoto, and H. Hashizume, “Toss-it: Intuitive Information Transfer Techniques for Mobile Devices”, *Proc. CHI’05, 2005.*

[Shen06]C. Shen, K. Ryall, C. Forlines, A. Esenther, F. D. Vernier, K. Everitt, M. Wu, D. Wigdor, M. R. Morris, M. Hancock, E. Tse, “Informing the Design of Direct-Touch Tabletops”, *IEEE Computer Graphics and Applications, (26)5. P. 36-46.*