Phone N' Computer (PnC)
Teaming up an Information Appliance with a PC

Min Yin Jeffrey Pierce Shumin Zhai

IBM Research - Almaden
650 Harry Road, San Jose, California, USA
zhai@almaden.ibm.com http://www.almaden.ibm.com/u/zhai

Abstract. Facing the ever growing complexity and usability problems of the PC, some propose specialized computers as a solution, while others argue that such "information appliances" are unnecessary. Rather than pitting information appliances and PCs against each other, we argue for instead exploring the design space for using them together. We experimented with a *device teaming* approach that takes advantage of both types of devices: the familiar and high bandwidth user interface of the PC, and the task specific form factors of an information appliance. In our experimentation, we designed and developed a PnC (phone n' computer) by teaming up an IP phone with a general-purpose PC. We outline the design space for such a combination and describe several point designs we created that distribute functions between the two devices according to their characteristics. In comparison to separately using phones and computers, our designs provide new and richer user experiences including drop-to-call, sharing visual information, and caller information display.

Keywords: Phone, information appliance, device integration.

Introduction

Many researchers and thinkers in the human-computer interaction field, such as Norman ("The invisible computer" [8]) and Buxton ("Less is more" [2]), have advocated information appliances, or computers designed with specialized functions and form factors, as a solution to the ever increasing complexity and user frustration of the PC. Being special purpose, information appliances have advantages over the general purpose PC. They tend to be more reliable and simpler to use, at least for the most basic functions.

The IP phone, rapidly deployed in many corporations due to the cost savings in networking, is a particularly interesting example of an information appliance (Figure 1). An IP phone is nothing but a specialized computer, connected to the same network as an office desktop or laptop PC, and it often sits near a general purpose PC.

The function of an IP phone can be entirely implemented in software. So why do customers pay hundreds of dollars to have a separate hardware device while a much cheaper software solution is available? A plausible reason lies in the advantages of an information appliance. First, due to its specialized purpose and narrow functionality, an IP phone can be more reliable than a general purpose PC, which can halt due to failures in the numerous programs, device drivers, and processes. In case of a crash, a special purpose computer reboots much faster than a PC that has a complex operating system to boot and many applications to launch. The management and maintenance (e.g., software upgrades) of IP phones, especially in a large corporation, is also easier.

Second, because of its size and form, an IP phone offers better affordances for making and receiving calls. When the phone rings, a user can immediately pick up the handset of a hardware phone with a fixed location on the desk. By contrast, responding to an alert from a PC often involves finding the right window, reading the message, and figuring out which control to push.

The drawbacks of an IP phone as an information appliance are also easily observable. Beyond making and receiving calls, it is often not obvious how to perform more complex operations, such as making a multi-party conference call. Without a keyboard and a large display, adding entries to the speed dial menu or phone book is also difficult. It is also inconvenient to manually dial phone numbers stored in a PC (email, web page, etc) on the phone.

The proposition that information appliances should replace PCs is not without opposition (e.g. “The visible problems with the invisible computer” [9]). In comparison to special purpose devices, the general purpose PC is relatively inexpensive considering its powerful hardware due to the economy of scale. It also saves space because it is multi-purpose (one machine instead of many on a desk or in an office). In terms of usability, a PC has at least three advantages over appliances. First, most users have explicitly or implicitly invested a large amount of time learning interaction skills for common GUI applications. GUI conventions and their de facto standard operations allow a user to rely on past experience when operating a new software application. In contrast there has been little standardization to performing operations on an appliance, particularly if the operation is beyond the most basic. For example, recording a greeting message on different phones often requires different procedures. Second, PCs have high bandwidth input (mouse and keyboard) and output (large display) channels that most specialized devices lack. Third, information (e.g. a phone number sent over email) re-

lated to daily tasks is often stored in applications on a PC and users can easily share that information among other applications on the same machine.
Rather than pitting information appliances against the PC, we argue instead for exploring the design space when users employ both types of devices together. In this paper we explore the design space for teaming an IP phone (an information appliance) with a PC. By teaming up information appliances with PCs, one can approach the best of both worlds: the special form, size and affordances of an information appliance and the powerful I/O and GUI capabilities as well as the access to the wealth of information and tools of a general PC.
In the next section we describe previous research exploring user experiences that span multiple computing devices. We then discuss the possible design space for experiences that span IP phones and PCs, and describe the architecture we created to allow us to implement several point designs within that space. Finally, we present the prototype designs that we created and discuss potential next steps.

Fig. 1. An IP phone and general purpose PC can form a PnC team taking advantage of both devices

Related Work

The concept of device integration is not unique to the present work. Schilit and Sengupata [13] articulated the advantage of "device ensembles". Pierce and Mahaney [10] proposed "opportunistic annexing" to enhance mobile devices by borrowing resources from other devices. Pierce and Nichols [11] developed an infrastructure for extending applications' user experiences across multiple per-

sonal devices. Dearman and Pierce [3] studied the problem of owning and using multiple computing devices. Hutchings and Pierce [5] discussed how and why users might divide an application's interface across devices in private, semi-private, and public environments. Barton et al [1] devised a SIP-based approach to connecting mobile devices with PCs. Johnson et al [6] provided the ability to move existing web pages or linked information among multiple displays. Yin and Zhai [15] used a device integration approach to team up a PC and a phone to tackle the IVR voice navigation problem.

The Design Space for PnC

A PnC (phone n' computer) system, as a team between a phone and a computer, should provide stronger functionality and usability than each of the two devices alone. As such, the potential design space for user experiences leveraging such a team should draw on the functionality that each device can provide. We will first elucidate the functionality that the phone can provide, and then follow it with a discussion of the functionality that a PC can provide. We then discuss possible experiences that a PnC system combining the functionality offered by each could provide to users.

IP Phone Functionality

While the primary functionality of IP phones is to allow users to make and receive calls, IP phones have information around that functionality that they can potentially share with a PC. That information includes when the phone is making or receiving a call; the phone number of the other party; the start time, end time, and duration of a call; any buttons pushed during a call; and potentially even the incoming and outgoing audio streams. IP phones also possess microphones that could be used to capture sound or speech and hardware buttons that could provide input.

In addition to acting as a data source, IP phones can also serve as targets for displaying information. Such phones typically possess digital screens that can dis-

play both text and images. A phone's screen has the advantage that it is often positioned in the periphery of a user's workspaces in a persistent location (unlike an application window that may float around the large PC screen or be hidden by other applications windows), making it a convenient destination for displaying information that the user might want to be aware of or take a quick glance at. In addition, IP phones with speakerphone capabilities typically possess speakers that can play audio.

PC Functionality

PCs by their nature offer a larger and more varied set of functionality. Rather than listing all of the functionality that a PC could potentially offer, therefore, we will concentrate on functionality that centers on input and output and on communication and coordination with other people.

The display screen offered by PCs is a natural fit for any experience that needs to provide a large amount of graphical or textual information. Similarly, the high bandwidth input mechanisms provided by PCs (typically through a keyboard and mouse) offer the ability to create experiences that require complex user actions. The communication and coordination functionality offered by most PCs is also quite rich. PCs offer applications for communication (e.g., email, instant messaging) and store the resulting artifacts (e.g., email and IM messages, transmitted documents, contact information for other people). PCs also contain information about how users coordinate with others, particularly their scheduled meetings.

Potential combination advantages

Our discussion of possible experiences drawing on combined IP phones and PCs will start with making and receiving phone calls and then expand to other communication functions.

Make calls. In order to make a call the user must specify a number to dial. This can always be done with the old-fashioned touchtone dial on the phone. In addition, there are several possible ways that the phone and PC could collaborate to specify one or more candidate numbers. First, the user could specify the number to call by selecting phone numbers from a variety of sources: an address book, email messages, web pages, meeting invitations, etc. Alternately, the PC could proactively suggest a call for the user, for example by sending the call-in number for a meeting at the meeting's designated start time.

Alternately, the PC could suggest a candidate set of numbers that the user might want to call and the user could select from the appropriate number on the phone. Drawing on the idea of Recent Shortcuts [14], the candidate numbers might be associated with people that the user had recently had contact with (e.g., via email or instant messaging) or with documents or pages that the user had recently read.

Receive calls. When users receive calls two key pieces of information are the identity of the caller and the context for the call. By sharing the number of the calling party, an IP phone could allow a PC to initiate a search both for the identity of the calling party as well as for information that might help ground the conversation. The PC could either display potentially relevant context itself, or it could send that context back to the phone for it to show on its built-in display. Another approach to teaming the PC and phone when the latter receives calls would be to use information from the PC about the user's activities to determine whether incoming calls ring silently or go directly to voice mail. For example, if the user's PC calendar shows that he is in a meeting, the IP phone could send incoming calls to voicemail to prevent interruptions.

Call information. IP phones could also help reduce interruptions by notifying PCs when users are in the midst of a call. The PC could then act to reduce external interruptions by, for example, setting the user's instant messaging status to indicate that he is on a call or reducing the volume on the PC's audio output.
An IP phone that provides information about the numbers a user calls and how long those calls last could also allow the PC to provide more advanced functionality. The PC could provide recent shortcuts that allow users to easily email or IM people that they have talked to on the phone. The PC could also open up an additional communication channel (e.g., an instant messaging session or a web-based meeting) to allow users to easily exchange textual or graphical information during a call. A PC address book could capture information about when users had last called individual contacts. Social networking applications on the PC could use the call data to model users' relationships with other people.

Display. In addition to displaying potential context for received calls, IP phones could serve as visual or audio peripheral displays, allowing PCs to display notifications or alerts without occluding the work that users are engaged in on their PC.

Input. The microphone on IP phones could be used for voice input on PCs, or it could serve as a sensor to allow a PC to attempt to model the current activity. The IP phone could also share the incoming and outgoing audio streams during a call to allow the PC to capture users' phone conversations.

While all of these designs have potential advantages (and disadvantages), we chose to create three prototypes to demonstrate promising points in the design space. These prototypes allow users to initiate calls on the PC by dragging and dropping phone numbers, easily share web pages as context during a call, and view information about the other party to help users determine both the identity of the other party and potential reasons for the call. We next describe the system that we created to support these prototypes and then describe the prototypes themselves.

System and implementation

One important design criteria we had for PnC was that we did not want to introduce any direct physical connection between the PC and the phone or burden our users with additional device discovery steps before they could use the system. Instead, we took a server-side integration approach where server considers the PC and the phone of a given user as a virtual PnC team, as shown in Figure 2. Such an approach is easy, convenient, flexible, and cost effective.

Figure 2: Computer and Phone Virtual Integration

Figure 3 shows the system architecture of a PnC system. In this system, PnC is offered as a set of functional software services.

The user first registers with the PnC Server via a web interface and then subscribes to one or more PnC services, such as Drop-to-Call, Conference-call-dropping, Caller Kaleidoscope (see below), multicast for a certain group etc. During registration, the PnC server will verify the user's name and phone number with the IP Phone Manager Server (authenticating the user by means of a password), initialize the user's phone, and later deliver the subscribed services to the same phone. The PnC server keeps a list of available PnC services and a list of subscribers with their individual subscriptions. The Subscription Management Module allows a subscriber to modify, customize, or cancel his/her subscriptions. For each PnC service, the complete service delivery consists of two parts: one delivered through the PC interface to the subscriber's computer and the other part through the phone interface to the IP Phone. The Service Management Module manages and maintains these services to ensure their successful delivery to the PC and to the IP phone.

On the subscriber's PC, a small PnC proxy receives delivered PnC services and collects user input through the PC's keyboard and mouse. On the IP phone side, PnC delivers services through the standard IP phone service platform provided by the IP Phone vendor.

In our implementation of the prototype PnC system, the PnC server runs on an IBM ThinkPad T40 (1.5GHz, 1GB RAM), while the IP Phone Server was a Cisco CallManager 4.1 running on a Windows-based IBM e-server xSeries 345. The users' PCs were IBM ThinkPad T41 (1.5GHz, 1GB RAM) laptops, and the IP Phone was a Cisco IP Phone 7970. For Cisco IP Phones, the standard service delivery platform is HTTP. Each Cisco IP Phone runs both an HTTP Server and an HTTP Client, which enables both "Push" and "Pull" service modes. For these IP Phones, the "Phone Interface" running on the PnC server was a set of predefined HTTP services aimed at the Cisco IP Phone browser.

Prototype PnC Functions and User Interfaces

Using the system architecture described in last section, we implemented three prototype PnC functions, described in this section. Before that we give a brief description of the user interfaces between the PC and the IP phone.

From the PC to the phone

There are many possible interface designs for manually transferring information objects from the PC to the phone. One uses a "toss-over" metaphor: an edge of the PC screen near the phone is dedicated as boundary between the two devices, where any objects that the user drags over the edge goes to the phone. Another possible interface is a pop-up menu attached to each information object that a user can transfer to the phone by a right mouse click and selection. We settled on a third alternative: an interface design we called "graphical proxy" that uses a dedicated window/icon as a virtual representation of the phone on the PC screen. Any object dropped on the proxy will be interpreted and transferred to the phone (See Figure 4).

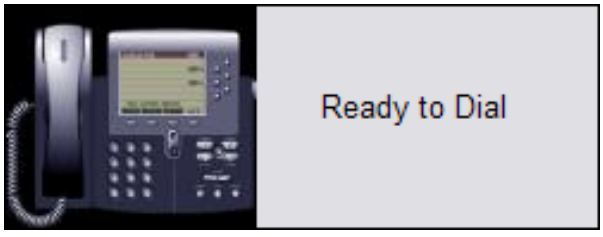

Figure 4: The graphical phone proxy on the PC desktop. Any object dropped on the proxy will be interpreted and transferred to the phone

From the phone to the PC

There are also many possible interface designs for transferring information objects from the phone to the PC. We choose a simple button-based "shovel" mechanism. If a user presses a button with the label "To PC", the object on the phone screen is transferred and displayed on the PC.

Drop-to-call

The first service we implemented was Drop-to-Call, which enabled users to drag and drop a phone number in a PC application onto the graphical phone proxy to cause the IP phone to automatically dial that number.

Drop-to-Call also allowed users to drop a directory entry onto the phone proxy. The phone will dial the number in the entry when there is no ambiguity. When multiple phone numbers (e.g. office number, mobile number, etc.) are contained in the entry, the user will be prompted to select one of them through a pop-up menu.

Sharing web pages

During a phone call it is common to refer to a web site for sharing some visual information. To achieve this today, one has to read the URL over the phone (which is often tedious), send an email, or start an instant messaging conversation. We implemented a PnC function to facilitate sharing web pages. During the call, either of the two parties with PnC installed can drag and drop a web page from the web browser of the PC onto the phone proxy, which causes PnC to display an image of the page on both phone screens. If interested in more detail, the recipient can "shovel" the page from the phone display to the PC, which will display the page in a web browser. Note that what is transferred in the last step is really the URL of the web page, although the appearance is that the "page" is copied from one PC to another via two phones.

Caller Information Display: Kaleidoscope

When receiving a call, it is often useful to have some background information about the caller on display. Conventional caller ID provides a small bit of such information in the form of a phone number and caller's name. We have developed a PnC service, Kaleidoscope, that provides a variety of information on the caller as well as recent collaboration and communication history and documents shared

between the caller and the recipient. With this service, when one receives a call, PnC displays an enhanced version of caller ID on the phone screen, including: name, affiliation, email address, as well as any shared calendar entry and shared activities (Figure 5). If interested in seeing more detailed information during the call, the receiver can shovel the information to the PC and display a full size Kaleidoscope interface on the PC screen. Depending on availability, the Kaleidoscope displays the caller's home page, corporate directory information and most recent email messages exchanged with the caller (Figure 6).

The information displayed in Kaleidoscope is collected by the Caller Kaleidoscope service which, once subscribed, starts to monitor the inbound calls of the subscriber's IP Phone. Whenever a call comes in, the service detects the calling party's caller ID (if it is available). The service will then collect various kinds of information about the caller. This service is especially valuable in a corporate environment where information such as the caller's location, department, manager, and position can be easily found in the corporate directory. If the caller is from outside of the receiver's corporation, Caller Kaleidoscope can search information on the web and present whatever is found to the receiver. Figure 5 shows a snapshot of the Caller Kaleidoscope delivered to a recipient's IP Phone. A full-sized version of Kaleidoscope on the PC is shown in Figure 6.

Figure 5: Caller Kaleidoscope Phone Screenshot

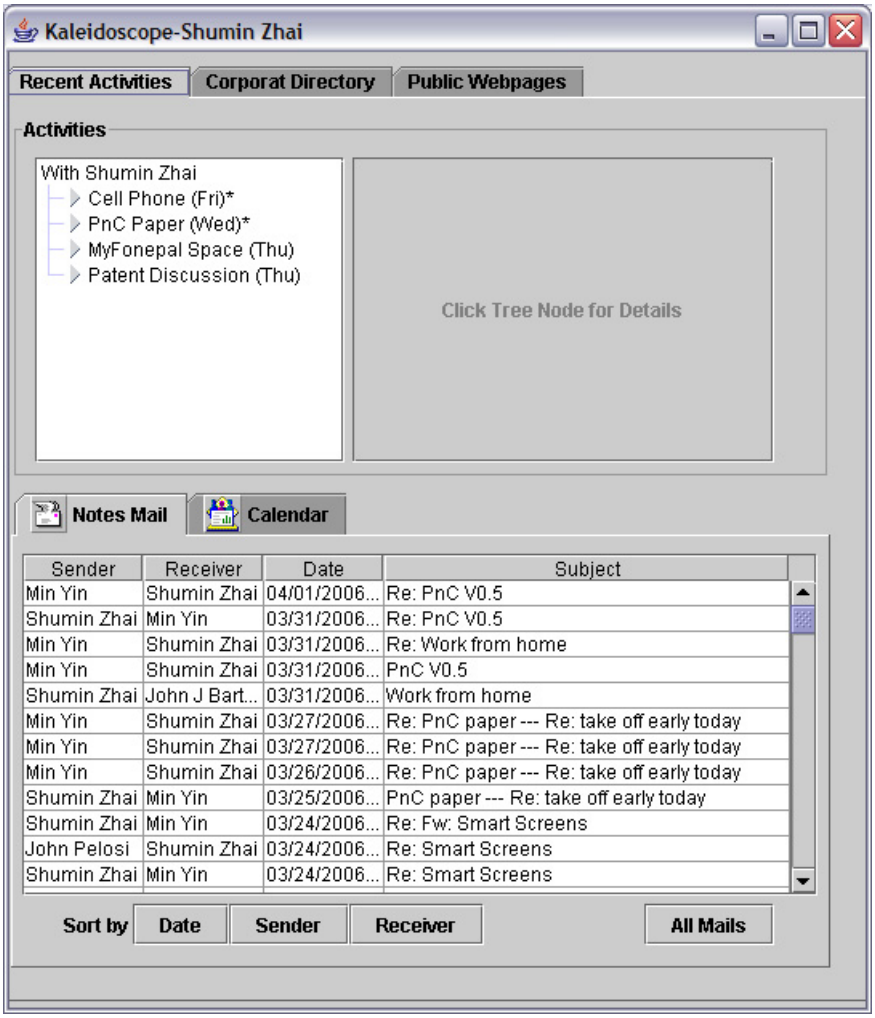

Figure 6: Kaleidoscope: Conversation background with regard to the caller (Shumin Zhai in this case) is displayed on the PC screen of the recipient (Min Yin). Note that the recipient can also select other taps for additional information

Beyond PnC: IP Phone + PIE

A key advantage of the PnC approach is the access to and leverage of contextual information residing on the PC. To further that advantage we have also explored leveraging information resources in a user's personal information environment (PIE), which potentially consists of multiple personal device (such as desktop and laptop PCs, mobiles phones, and PDAs) [11]. We implemented a sample PIE service where a user's IP phone shows up as one of devices in his or her PIE and broadcasts call information to the user's other devices (you're calling someone, someone's calling you, the call connected, the call ended). A PIE service designed to run local searches (via Spotlight, Google Desktop Search, or Beagle) for information on all of these devices can take such call events and return potentially relevant context for the phone call. This example shows that the IP phone can not only be teamed with a PC, but also a greater number of personal devices, opening up an even greater design space.

Discussion and Conclusions

Information appliances offer simplicity and special physical forms that can be easy to use, at least for their core functions. The general purpose PC has powerful input and display capabilities as well as a wealth of information in its various applications. Often both types of devices already coexist. We argue that teamed together they could be made even more powerful than either of them used separately due to the greater contextual information resources and UI characteristics of both types of devices.

As an example of information appliance and PC teaming we developed PnC (Phone N' Computer), which teams up a particular type of information appliances - IP phones - and the PC, both as a general concept and as a prototype system. PnC allows a user to easily pick up the phone handset to answer a call as is usual with a physical phone, but it also allows the user to drag a phone number from another PC application and drop it onto the graphical proxy icon of phone to dial a call. With PnC, information is displayed either on the phone (which provides a small peripheral screen in a consistent location), or on the PC (which has a large dynamic display that may be crowded with many application windows). In addition to default automatic assignments according to the type of information and characteristics of each type of device, users can also manually pass information between the PC and the phone.

We have not pursued PnC as an A vs. B comparison type of research; it has instead been about enabling what was not previously possible. As such this paper has focused on the design space exploration of PnC and on the technical feasibility through prototype system implementation. Our exploration exposed various useful functions that a PnC can offer. We implemented and demonstrated three point design PnC functions, drop-to-call, web page sharing, and a rich caller information display, that are difficult (if not impossible) to achieve with a phone alone. All of these implemented functions ran in real time with off-the-shelf technology. For example there was no perceptible increase in dialing response latency when using drop-to-call over a regular touchtone phone pad.

What we have demonstrated are only point designs. Many more PnC functions and services, some obvious and easy to implement (such as conference calls and adding a third party to an ongoing call) and others that require design creativity, can be developed similarly. More broadly, our experiment has illustrated the approach of teaming up information appliances with a general purpose PC to achieve a more productive user experience.

Acknowledgments

We thank colleagues Jon Graham, John Tang, John Barton, Steve Ferrell, Tom Moran, Steve Cousins, and Barton Smith. Min Yin currently works at SRI International.

References

1. Barton, J.J., Nylander, S., Folowosele, F. and Harrison, B., Dialing for Displays: Session Initiation Protocol for Opportunistic Augmentation. Proc. Fourth IEEE Intl conf Pervasive Comp and Comm, 2006, 60-65.

2. Buxton, W. Less is More (More or Less). in Denning, P. ed. The Invisible Future: The seamless integration of technology in everyday life., McGraw Hill, New York, 2001, 145 -179.

3. Dearman, D., Pierce, J. "It's on my other computer!: Computing with multiple devices." Proc. of CHI 2008: ACM Conference on Human Factors in Computing Sytesm. 767-776.

4. Fox, A., Johanson, B., Hanrahan, P., and Winograd, T. Integrating Information Appliances into an Interactive Space. IEEE Computer Graphics and Applications, 20, 3 (May/June 2000), pp. 54-65.

5. Hutchings, H. and Pierce, J. Understanding the whethers, hows, and whys of divisible interfaces. In Proceedings of AVI 2006, pp. 274-277.

6. Johanson, B., Pennekanti, S., Sengupta, C., and Fox, A. Multibrowsing: Moving Web Content across Multiple Displays. In Proceedings of Ubicomp 2001, pp. 346-353.

7. Newman, M., Izadi, S., Edwards, W.K., Sedivy, J., and Smith, T. User Interfaces When and Where They are Needed: An Infrastructure for Recombinant Computing. In Proceedings of UIST 2002: ACM Symposium on User Interface Software and Technology, pp. 171-180.

8. Norman, D.A. The Invisible Computer: Why Good Products Can Fail, the Personal Computer Is So Complex, and Information Appliances Are the Solution. MIT Press, Cambridge, Mass, 1998.

9. Odlyzko, A.M. The Visible Problems of the Invisible Computer: A Skeptical Look at Information Appliances First Monday, 1999.

10. Pierce, J.S. and Mahaney, H.E., Opportunistic Annexing for Handheld Devices: Opportunities and Challenges. Proc. Human-Computer Interaction Consortium (HCIC), 2004.

11. Pierce, J., Nichols, J. “An infrastructure for extending applications’ user experiences across multiple personal devices.” Proc. of UIST 2008: ACM Symposium on User Interface Software and Technology, 101-110.
12. Roman, M., Hess, C., Cerquiera, R, Ranganathan, A, Campbell, R., Nahrstedt, K. A Middleware Infrastructure for Active Spaces. IEEE Pervasive Computing, 1 (2002), pp. 74-83.
13. Schilit, B. N. and Sengupta, U. Device Ensembles. IEEE Computer, 37, 12 (Dec. 2004), pp. 56-64.
14. Tang, J.C., Lin, J., Pierce, J.S., Whittaker, S., and Drews, C. Recent Shortcuts: Using Recent Interactions to Support Shared Activities. In Proceedings of CHI 2007: ACM Conference on Human Factors in Computing Sytesm, pp. 1263–1272.
15. Yin, M. and Zhai, S., Dial and see: tackling the voice menu navigation problem with cross-device user experience integration. Proc. UIST 2005 -- 18th ACM Symposium on User Interface Software and Technology, 2005, ACM, 187-190.