Dynamic User Interface Creation based on Device Descriptions

Mir Farooq Ali
Human Interaction Research,
Motorola Labs
1295 E. Algonquin Rd.
Schaumburg, IL 60196, USA
farooq.ali@motorola.com

Kibum Kim
Human Interaction Research,
Motorola Labs
1295 E. Algonquin Rd.
Schaumburg, IL 60196, USA
kibum.kim@motorola.com

Dale Russell
Human Interaction Research,
Motorola Labs
1295 E. Algonquin Rd.
Schaumburg, IL 60196, USA
dale.russell@motorola.com

Zhuli Xie
Human Interaction Research,
Motorola Labs
1295 E. Algonquin Rd.
Schaumburg, IL 60196, USA
zhuli.xie@motorola.com

CHI 2008, April 5 – 10, 2008, Florence, Italy
ACM 1-xxxxxxxxxxxxxxxxx.

Abstract
In the past few years, the number of consumer devices and appliances possessing networking capabilities has increased dramatically. On one hand, we have non-mobile appliances like photocopiers, printers, media storage devices, sound systems and televisions, which are extremely powerful with a large number of features. Many of these devices are capable of advertising their features and functionality in some sort of declarative specification. On the other hand, we have cell phones, which are also increasingly powerful and have become pervasive and ubiquitous with their networking capabilities. In this position paper, we present our view of dynamically generating user interfaces (UI) on the cell phone for newly discovered devices based on their declarative device descriptions. The dynamically created UI can then be used to control and operate those devices.

Keywords
Dynamic UI creation, declarative UI specification, device description.

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI):

Introduction

In recent years, there has been an ongoing effort to standardize access and control of appliances. UPnP [13], HAVi [7], ZigBee [5], Bluetooth [2] etc. are some of the standards that provide guidelines for networked access and control of appliances and devices. For example, Bluetooth's service discovery protocol (SDP) provides an API for applications to discover the set of services that are available based on RF proximity of devices. DLNA [1] is a UPnP-based interoperability framework that seeks to tie three different domains: PC/Internet, mobile devices, and consumer electronics, to enhance the sharing and distribution of digital media in a home context. The framework consists of guidelines for interoperability and standards for various media formats, media management, device discovery and control etc. With the availability of these standards, and devices supporting them, it is possible for various appliances and devices to communicate with each other in a distributed fashion.

Another concurrent trend is the ubiquitous availability and usage of cell phones. Cell phone usage has rapidly increased, and it is estimated that there will be more than 2.5 billion cell phones worldwide by 2007 [10]. For most users, the cell phone has become an indispensable and ubiquitous accessory. Many of these phones come equipped with data-networking capabilities along with the voice channel. As people move in new environments containing networked devices that advertise their capabilities, it is reasonable to think of using cell phones to control and operate these devices, thereby treating cell phones as remote controls. As the functionality of consumer products expands, they are becoming more complicated, so that a user may be overwhelmed with a new device. A UI that maps the user's goals to the functions of the device [8] can greatly simplify the learning experience. Using a multimodal front-end to handle the user's commands via speech, touch, and/or keyboard, and a reasoning system on the backend to determine which functions/services the user is trying to invoke, will make the UI more effective and user-friendly. In this paper, we present one view of dynamically creating UIs on the cell phone based on the discovery of new devices, allowing the cell phone to be treated as a universal remote control.

The following scenario demonstrates the motivation for our approach:

> Mary is attending a technical talk in an unfamiliar convention center. She is interested in the talk and wants to get the presentation slides. She takes out her cell phone and knowing that the slides probably reside on some nearby media server, does a search for discoverable media servers. After discovering a media server, she is able to browse the contents and find the relevant slides based on the title of the talk. She then searches for discoverable printers in her vicinity and is able to print the slides on a printer outside the conference room.

In this scenario, there is not necessarily a built-in UI for operating the media server and the printer on Mary's cell phone. The UI for operating these devices is dynamically created based on the device type and description that is extracted from each of the devices *after* their discovery. To facilitate the process of creating the UI, we use a declarative frame-based UI

representation language that allows Natural Language interaction [10]. The details of the UI description languages and the transformation process that utilizes the device descriptions to create the UI are presented below.

Related work

Work has been done under the Pebbles project (http://www.pebbles.hcii.cmu.edu/) in automatically generating UIs on handheld devices to control household appliances [12]. However, the UIs that are generated are primarily graphical. Rather than just GUIs, our work focuses on creating multimodal UIs, including speech I/O. The SUPPLE system automatically generates consistent UIs using optimization techniques [6]. The Roadie system is a goal-oriented system that presents a UI to the user in terms of the user's goals, which can be different from the device's functionality and the actual controls it provides [9]. UIML [1], UsiXML [8] and Teresa [3] are well-known XML-based UIDLs. However, given the objectives of our project, described below, our frame-based UIDL was a more reasonable choice.

Research Issues/Parameters

Some of the research issues associated with this endeavor are discussed below:

1. *Differing form factors and/or computing power* - The devices could have widely differing capabilities, computing power and/or features. Devices may range in complexity from a light switch, with the sole capability of switching the light on or off, to a multi-functional device such as an all-in-one fax machine, copier, scanner, and printer. Obviously, the UI for the latter will be much more complicated than for the former.
2. *Multiple standards* - In a heterogeneous environment it is necessary to accommodate multiple standards of device discovery and description, where different devices describe and advertise their capabilities using different standards. In some cases, the devices might have incomplete or non-compliant device descriptions. The UI generator must then be robust, with enough intelligence to account for such shortcomings and still generate a reasonable UI.

Some of the desired features of the dynamically generated UIs are:

1. *Multimodality* - One of the desired capabilities is the generation of a multimodal UI. For example, the end user might want to control the newly discovered device using both speech and a graphical user interface. The generated UI could also support natural language processing (NLP) to allow simpler and more intuitive speech interaction between the user and the device.
2. *Personalization* - If information is available about the user's usage of other devices and personal preferences, this could potentially be used to personalize the generated UI.
3. *Context-sensitivity* - It might be useful and/or necessary to incorporate the notion of "context" into the generated UI to account for the context-sensitive nature of certain device capabilities. We are assuming that it is possible to glean such information from the device upon its discovery.
4. *Generality and specificity* – If a user can use multiple devices simultaneously, it may be desirable to present the user with a common UI which does not distinguish which functions/actions are performed by which devices. Alternatively, a user may specifically request a certain action to be performed on a specific device. The dynamically

generated UI should accommodate both general and device-specific user commands.

The above list enumerates a few of the research issues and parameters that affect the generation of the UI for newly-discovered devices.

Dynamic UI creation process

As an initial starting point, we are limiting ourselves to UPnP enabled devices. The UPnP forum (http://upnp.org/) provides guidelines for various devices. Each UPnP-conformant device must adhere to these guidelines, which include the device description. This is an XML-based representation that indicates the manufacturer, device ID, and some other metadata about the device. Part of the information it contains is the service description. The service description provides information about what actions the device can execute, and what arguments are required for each action. The sequence of steps leading up to the UI creation is shown in Figure 1.

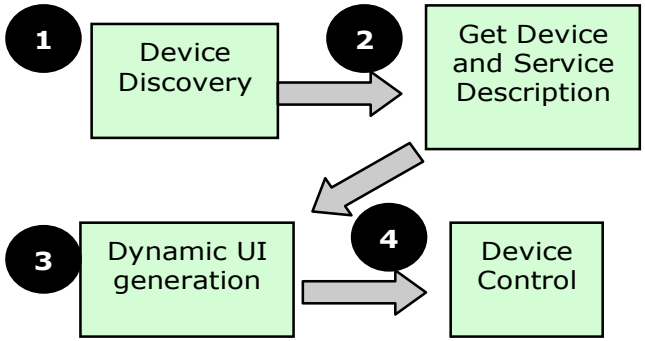

Figure 1: Sequence of steps for UI generation

Each UPnP device periodically broadcasts its presence. When the device is discovered, we extract the device and service descriptions from the device and use these to dynamically generate a UI, which can then be used to control the device. This is shown in Figure 2.

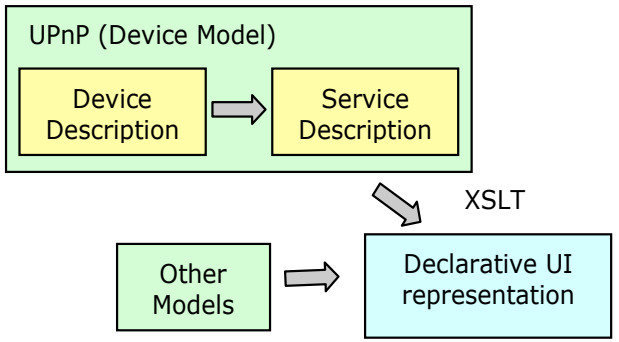

Figure 2: Automatic UI generation process

As mentioned in the introduction, we use a frame-based declarative UI representation for the generated UI, similar in structure to VoiceXML. We use the XML transformation language (XSLT) to dynamically create the UI, based on the UPnP service description. Other information used to enhance the generated UI is extracted from various models.

Initial lessons learned

Our initial experience in working with UPnP-enabled devices has been that it is possible to create UIs based solely on the device and service descriptions that are retrieved from the devices, but the UIs are not very usable. The device descriptions need to be supplemented by other models and information for the UI generation process so that the generated UIs are

more usable, and also achieve the goals that we outlined earlier for the generated UIs.

Conclusion

In this paper, we have presented a vision for the automatic creation of UIs for different devices in an intelligent fashion, allowing those devices to be controlled through a cell phone. We have discussed a number of research issues and challenges associated with this vision. Based on some preliminary work, we have found that more information is needed besides the device descriptions to create usable UIs, but the use of declarative UI representations helps us overcome some of the challenges discussed above.

REFERENCES

[1] Ali, M.F., Perez-Quinones, M., and Abrams, M., Building Multi-Platform User Interfaces with UIML. In Seffah, A., Javahery, H. (Eds.), Multiple User Interfaces: Cross-Platform Applications and Context-Aware Interfaces. John Wiley & Sons, Ltd, Ch. 6, 2004, pp. 95–118.

[2] Bluetooth, http://www.bluetooth.com/bluetooth/.

[3] Berti, S., Correani, F., Paternò, F., and Santoro, C., The TERESA XML Language for the Description of Interactive Systems at Multiple Abstraction Levels. in Proceedings of the ACM AVI'2004 Workshop "Developing User Interfaces with XML: Advances on User Interface Description Languages", 2004, pp.103-110.

[4] Digital Living Network Alliance, http://www.dlna.org.

[5] Egan, D. The emergence of ZigBee in building automation and industrial control Computing & Control Engineering Journal, 16:14-19, 2005.

[6] Gajos, K., Wu, A., and Weld, D. Cross Device Consistency in Automatically Generated User Interfaces, Proceedings of the 2nd Workshop on Multi-User and Ubiquitous User Interfaces. 2005, pp. 7-8.

[7] HAVi, the A/V digital network revolution http://www.havi.org/pdf/white.pdf.

[8] Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., Florins, M., Trevisan, D., UsiXML: A User Interface Description Language for Context-Sensitive User Interfaces, in Proc.of the ACM AVI'2004 Workshop "Developing User Interfaces with XML: Advances on User Interface Description Languages", 2004, pp. 55-62.

[9] Lieberman, H. and Espinosa, J. H., A goal-oriented interface to consumer electronics using planning and commonsense reasoning. In Proc. IUI 2006, pp 226-233.

[10] Thompson, W., and Bliss, H., A Declarative Framework for building Compositional Dialog Modules, in Proc. ICSLP 2000.

[11] Towards the next billion subscribers: Motorola delivers on seamless mobility vision http://www.motorola.com/mediacenter news/detail.jsp?globalObjectId=6405_6355_23/.

[12] Nichols, J., and Myers, B., and Rothrock, B., UNIFORM: Automatically Generating Consistent Remote Control User Interfaces, Proc. CHI 2006.

[13] Universal Plug and Play, http://www.upnp.org.