NVUI Framework: Natural View User Interface Framework for Consistent and Intuitive Appliance Control*

Hyoseok Yoon, Woontack Woo
GIST U-VR Lab.
500-712 Gwangju, S. Korea
{hyoon,wwoo}@gist.ac.kr

Sang-Goog Lee
The Catholic University of Korea
420-743 Bucheon, S. Korea
sg.lee@catholic.ac.kr

Abstract

In this paper, we propose a novel user interface framework for controlling appliances in smart spaces based on the natural look of UI on physical appliances. When a user selects a specific target appliance to control, the proposed architecture and procedures provide NVUI (Natural View User Interface) on a user's mobile device to maximize user's a priori familiarity with the user interface. We present a concept behind NVUI and how it can be generated, implemented and deployed in smart spaces.

1. Introduction

In everyday life at work and at home, we interact with a dozens of electronic appliances. The interaction frequency will be increased in ubiquitous computing environment, as a great number of computational devices come embedded into our environments. With this phenomenon, end-users inevitably face heavy burdens of learning a new user interface for different varieties. Similarly, interface designers undergo time-consuming yet costly UI re-design phase for different platforms with various hardware and software specifications in mind.

Many research efforts attempted to solve this problem by introducing universal controllers which can be used to control a number of different products. Even with commercial universal controllers in the market, they suffer from lack of flexibility and usability due to fixed arrangements of controls. Another research direction focuses on automatically generating user interfaces from an abstract and common base for multiple appliances. However, in the process, these approaches artificially alters the look and dismantles the structure of original user interface resulting in loss of consistency and aesthetics.

Therefore, we propose a concept called *Natural View User Interface (NVUI)*, which utilizes the consistent look and structure of physical interface of an appliance as well as equivalent functionalities on the mobile device to provide intuitive and natural user interfaces. With a NVUI-equipped mobile device, end-users can benefit from its intuitiveness and UI designers can save their time and cost since NVUI builds upon pre-designed UIs.

2. Related works

There are a number of projects and systems developed in UbiComp where a user's mobile device is used as a ubiquitous input device [1]. Especially, mobile device has an advantage of being a private mobile unit yet computationally powerful enough to manage personal data and interact with services in smart spaces. In Pebbles project[1], they investigated the use of PDA and smart phones as a personal universal control (PUC). They implemented multiple user interfaces on the mobile device to control a series of different devices. The recent work by Jeffrey Nichols [3][4] addresses automatic user interface generation and consideration of consistency in the generated user interface. But their work artificially rearranges control components and loses the original aesthetics of user interface, since the composition of control components are replaced with simple GUI buttons and menu controls.

U-Photo [7] is an interactive approach that is based on taking a picture metaphor. In U-Photo application, a control user interface is acquired by taking a picture of target smart object. Even though U-Photo intuitively maps a target device and control user interface, the menu-based user interface is far different from the physical user interface in look-and-feel.

Our previous work, Personal Companion [8] also provides a camera-based mobile controller for smart home en-

*This research is supported by the Foundation of UCN Projects, the MKE, and the 21C Frontier R&D Program in Korea as a result of subproject UCN 08B3-O1-20S

[1] http://www.pebbles.hcii.cmu.edu/

978-0-7695-3259-2/08 $25.00 © 2008 IEEE
DOI 10.1109/ISUVR.2008.20

vironment. However, it also possesses the same limitation and usability problems found on other mobile systems. The generated UI does not conform with the physical user interface found on the device which becomes cognitive loads for users to strive for consistency in user interface.

For usability issues, several user studies reveal insights on mobile interaction behaviors. Roduner et al. [5] found that operating appliances with mobile phones or AID (Appliance Interaction Device) was faster for control and problem solving tasks but for everyday tasks traditional interaction method which operates on physical interface was faster. Interestingly, most participants would not give-up on the traditional interface and do not consider a software-only UI as an option for them. Participants also reported that AID could improve on its poor visual appearance.

Rukzio et al. [6] explored on three physical mobile interaction techniques and found that choice of interaction techniques depend on given situations. For example, an indirect interaction is used to compensate a physical distance between user and target and avoid physical effort to getting close to the target.

3. Natural view user interface framework

In this section, we define NVUI and present a conceptual framework of how it can be implemented. NVUI aims to deliver a high quality user interface, both intuitive and usable by embracing the idea of user preference on familiar and already available traditional interfaces [5]. NVUI is a novel user interface concept that can be interchangeably used with traditional interfaces when the direct interaction on traditional interface is hindered, e.g., from remote distance. To realize this, NVUI supports the same structure and visuals of the traditional user interface. So NVUI is defined as follows. NVUI is a user interface data model composed of a *base image* of a controllable unit, *functional descriptions* of the control components and *mappings* of the base image and functional descriptions.

In specific, a *base image* refers to an actual image of user interface for the target appliance which is presented as a base of NVUI in a user's mobile device. When the base image is displayed in the mobile device, the XML-based *functional descriptions* of the control components and *mappings* are analyzed and parsed to maintain consistency in user interface structure and organization of each control components. Control components include buttons and LED status indicators that can be seen on the physical surface of the appliance, which is physically used to control the device or to show internal status of device.

There are three main communicative components in NVUI framework as shown in figure 1. *Natural View Console (NVC)* is a mobile device with touch screen and camera that is used to display NVUI and control the target device. NVC can be PDA, smart phone, cellular phone, laptop and UMPC[2]. NVC is responsible for discovering and detecting targets, processing NVUI data model for mapping and NVUI output and handling interactions with the generated NVUI. *Target* refers to any IP networked device that can be communicated and controlled. Examples include networked household appliances and smart objects and services with network capabilities. *Target Server* is a separate or an integrated server for holding the database of contents for the target. An example of target server is product web page which contains product specification along with a number of product images. *NVC-Target Network* is a network interface designed to work with service discovery protocols such as UPnP[3] protocols to provide control connection between NVC and targets. *NVC-Target Server Network* is a dedicated network connection between NVC and Target server to send and receive NVUI base images, functional descriptions and mapping relations from target server to NVC.

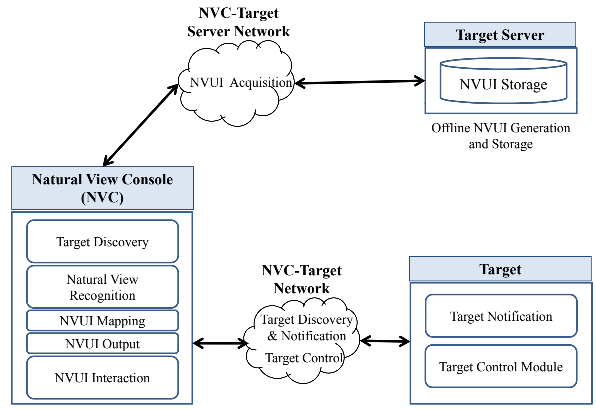

Figure 1. Conceptual Framework for NVUI provision

4. Procedures

A typical procedure of getting NVUI for controlling a target device is as follows. First, the user selects a target device with pointing, touching or scanning methods. For selecting a target device, users can utilize traditional GUI selection method and select a device from a list of services generated from scanning methods. Other selection techniques based on visual markers or natural view recognition can also be used in the selection step.

Second, from the selection step, an identifier of the target is acquired. This identifier acts as a key or URL to con-

[2]http://en.wikipedia.org/wiki/UMPC
[3]http://www.upnp.org

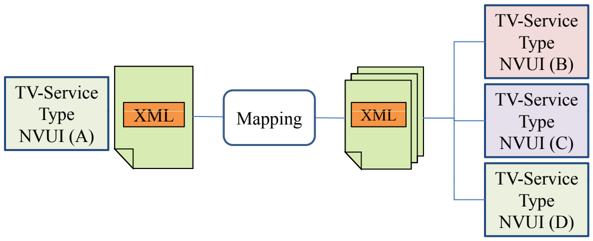

Figure 2. Block diagram for NVUI mapping

nect to the target server which host detail information of the target device. With an acquired target ID, a NVC-Target Server Network connection is established and NVUI base image and XML functional description are transferred to NVC. NVUI base image can be then rendered on the display of NVC and control functions are assigned as described in the XML functional description.

Third, before rendering NVUI base image and associating with described functions, NVUI mapping takes place optionally. If the acquired NVUI is a new NVUI that the user has not seen previously, then the system asks the user for mapping NVUI for one of the same type service NVUI that the user is familiar with. Otherwise, the user can try the new NVUI or this step is entirely omitted if the user has previous experience with acquired NVUI. The idea is depicted in figure 2. The strategy for mapping is ordered as follows, mapping of the same XML tags found on both XML functional descriptions, mapping between similar keywords, and readjustment of remaining functions.

Lastly, the user interacts with the rendered NVUI by selecting a proper control component to send its corresponding commands to the target via NVC-Target Network.

5. Implementation

We are currently developing a prototype of NVUI-based controller for a touch screen devices such as UMPC. We are using touch screen UMPC since it can simulate buttons and provide sound feedback. Also UMPC is a good platform to start prototyping due to easier debugging options than other mobile device platforms. For target devices, we chose two different TV devices (Samsung PAVV, LG XCanvas) and one printer (Canon LBP5960) user interface. Figure 3 shows the base images used for each device.

To prepare base images for these devices, we first captured the image of physical UI with a high quality digital camera and revised it to remove any present noise. Then we described each control component in the base image in XML according to information found on homepage of the device and printed product manuals. The XML describes a NVUI name, product homepage, location for NVUI base

Figure 3. Target device and implemented NVUI

image, size of base image, coordinates of each control component in the base image and corresponding commands. as shown in figure 4.

6. Challenges and discussion

There are several limitations in our prototype and many challenging and practical issues in deploying NVUI in real environment.

- Current NVUI is only visually natural and intuitive, and lacks other modalities. Other modalities such as haptic and tactile interfaces in recent mobile phone development and speech interface[2] would make NVUI more intuitive and easy to use. Also moving from 2D base images to 3D base models can increase visual perceptions.

- Selection or recognition of target from a distance is challenging problem. Popular mobile interaction techniques such as touching, pointing and scanning techniques [6] can be used with NVUI. However, these techniques are efficient for a near distance interaction and not appropriate for a considerably remote interaction. Similar phenomenon occurs with camera performance, since it becomes practically impossible to recognize an object from the far distance. Therefore these are technically bounded ranges for NVUI which we expect to be increased as the related technologies continue to develop.

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<NVUI name="Printer">
  <TargetUrl url="http://uvr.gist.ac.kr/~hyoon/"/>
  <ImageUrl image="http://uvr.gist.ac.kr/~hyoon/img/Printer.jpg"/>
  <ImageSize width="3508" height="842"/>
  <ControlNumbers count="8"/>
  <ControlButtons1 name="Online">
    <Coordinate topLeftX="270" topLeftY="300" bottomRightX="535" bottomRightY="570"></Coordinate>
    <Function commands="Turn On"></Function>
  </ControlButtons1>
  <ControlButtons2 name="CancelJob">
    <Coordinate topLeftX="630" topLeftY="260" bottomRightX="810" bottomRightY="625"></Coordinate>
    <Function commands="Cancel Job"></Function>
  </ControlButtons2>
  <ControlButtons3 name="FeederSelection">
    <Coordinate topLeftX="630" topLeftY="515" bottomRightX="810" bottomRightY="625"></Coordinate>
    <Function commands="Open Feeder Selection"></Function>
  </ControlButtons3>
  <ControlButtons4 name="Utility">
    <Coordinate topLeftX="2365" topLeftY="380" bottomRightX="2535" bottomRightY="490"></Coordinate>
    <Function commands="Open Utility"></Function>
  </ControlButtons4>
  <ControlButtons5 name="Job">
    <Coordinate topLeftX="2565" topLeftY="245" bottomRightX="2735" bottomRightY="360"></Coordinate>
    <Function commands="View Job"></Function>
  </ControlButtons5>
  <ControlButtons6 name="Settings">
    <Coordinate topLeftX="2775" topLeftY="380" bottomRightX="2955" bottomRightY="490"></Coordinate>
    <Function commands="Open Settings"></Function>
  </ControlButtons6>
  <ControlButtons7 name="Reset">
    <Coordinate topLeftX="2565" topLeftY="510" bottomRightX="2735" bottomRightY="610"></Coordinate>
    <Function commands="Reset"></Function>
  </ControlButtons7>
  <ControlButtons8 name="Enter">
    <Coordinate topLeftX="3050" topLeftY="335" bottomRightX="3260" bottomRightY="540"></Coordinate>
    <Function commands="Enter"></Function>
  </ControlButtons8>
</NVUI>
```


Figure 4. An example of functional description for printer

- Automation tools can speed up the production and deployment of NVUI. In our setting, we manually produced NVUI base images and wrote functional descriptions from the scratch. Automation tools that can produce NVUI base images from taken pictures and generate functional descriptions by acquiring necessary information can ease the development of NVUI. Ultimately, we vision that service or device providers open their product's base images and functional descriptions online which can be reused for generating NVUI for different uses.

- When NVUI is deployed into mobile device, its screen size and resolution determine viewing areas as well as user's interaction space. To overcome these inherent limitations, NVUI can be manipulated with mobile interaction techniques such as Apple's cover flow[4] and multi-touch[5] which can zoom in and expand interaction spaces beyond physical specification.

- As a typical observation, the household appliances manufactured by the same company may share similar, if not the same physical user interface. This characteristic can be advantageous for users who can easily control a product line of the same company with a given NVUI, and for designers who also can save much time by avoiding the repetitive work.

[4] http://www.apple.com/itunes/jukebox/coverflow.html
[5] http://en.wikipedia.org/wiki/Multi-touch

7. Conclusions and future work

In this paper, we introduced a novel user interface called NVUI and its framework. We described the motivation for NVUI and how it can be implemented with the described framework. Also through our prototype, we discussed several practical issues and challenges upon deploying NVUI in real environment. Despite its primitive stage of development, we expect that NVUI can be an effective solution for providing natural, intuitive and consistent interfaces, especially for use with mobile devices with further research.

In future work, we will integrate NVUI for home appliance controller and conduct user studies to compare traditional interaction methods and our approach. Moreover, we will expand our approach and address issues for smart objects and services in UbiComp that do not have an explicit physical user interface.

References

[1] R. Ballagas, J. Borchers, M. Rohs, and J. Sheridan. The smart phone: a ubiquitous input device. *IEEE Pervasive Computing*, 5(1):70–77, 2006.

[2] B. LaPlant, S. Trewin, G. Zimmermann, and G. Vanderheiden. The universal remote console: A universal access bus for pervasive computing. *IEEE Pervasive Computing*, 3(1):76–80, 2004.

[3] J. Nichols, B. A. Myers, and B. Rothrock. Uniform: automatically generating consistent remote control user interfaces. In *CHI 2006: The SIGCHI Conference on Human Factors in Computing Systems*, pages 611–620, 2006.

[4] J. Nichols, B. Rothrock, D. H. Chau, and B. A. Myers. Huddle: automatically generating interfaces for systems of multiple connected appliances. In *UIST 2006: The 19th annual ACM Symposium on User Interface Software and Technology*, pages 279–288, 2006.

[5] C. Roduner, M. Langheinrich, C. Floerkemeier, and B. Schwarzentrub. Operating appliances with mobile phones – strengths and limits of a universal interaction device. In *Pervasive 2007: 5th International Conference on Pervasive Computing*, pages 198–215, 2007.

[6] E. Rukzio, K. Leichtenstern, V. Callaghan, P. Holleis, A. Schmidt, and J. Chin. An experimental comparison of physical mobile interaction techniques: Touching, pointing and scanning. In *UbiComp 2006: 8th International Conference on Ubiquitous Computing*, pages 87–104, 2006.

[7] G. Suzuki, S. Aoki, T. Iwamoto, D. Maruyama, T. Koda, N. Kohtake, K. Takashio, and H. Tokuda. u-photo: Interacting with pervasive services using digital still images. In *Pervasive 2005: 3rd International Conference on Pervasive Computing*, pages 190–207, 2005.

[8] H. Yoon, H. Kim, and W. Woo. Personal companion: Personalized user interface for u-service discovery, selection and interaction. In *Universal Access in Human-Computer Interaction*, pages 1052–1061, 2007.