Universal Interaction with Networked Home Appliances

Tatsuo Nakajima and Atsushi Hasegawa
Department of Information and Computer Science
Waseda University
3-4-1 Okubo Shinjuku Tokyo 169-8555 JAPAN

Abstract

In this paper, we propose universal interaction for networked home appliances, which is a simple mechanism to fill the gap between traditional user interface systems and advanced user interaction devices. Our system enables us to control appliances in a uniform way at any places, and the system allows us to select suitable input and output devices according to our preferences and situations. Also, the devices can be changed dynamically according to a user's preferences.

1 Introduction

In ubiquitous computing environments[6], one of the most important problems is how to interact with a variety of objects embedding computers. The interaction between us and computers embedded in various objects has been developed by several research groups[1, 4]. These devices enable us to interact with embedded computer more naturally. However, current standard middleware components for networked home appliances have adopted traditional standard graphical user interface systems such as Java AWT or GTK+. Therefore, it is not easy to control home appliances from advanced interaction devices such as PDAs, cellular phones, or a variety of research prototypes described above. Also, natural interaction is changed according to a user's current situation. For example, if a user is cooking a dish. S/he likes to control appliances via voices, but if s/he is watching TV on a sofa, a remote controller may be better. This means that the most appropriate interaction device should be dynamically chosen according to a user's current situation and preference, and the selection of interaction devices should be consistent whether s/he is living in any spaces such as at home, in offices, or in public spaces.

In this paper, we propose universal interaction for networked home appliances, which is a user interface system to fill the gap described above. Our system allows us to control various appliances in a uniform way at any places, and the system enables an application to use traditional standard graphical user interface systems such as Java AWT or GTK+, but a user can navigate the interface through a variety of devices such as PDAs, cellular phones, or advanced technologies. We show that it is possible to realize the goal very easily based on the stateless thin-client system such as Citrix Metaframe, Microsoft Terminal Server, Sun Microsystems Sun Ray, and AT&T VNC(Virtual Network Computing) system. We have built a prototype system, and shown that a user can use a variety of interaction devices carried by him. The prototype system is currently integrated with our home computing system[5] that have implemented HAVi(Home Audio/Video Interoperability)[3], which is a standard distributed middleware specification for home appliances, and shows that our system is useful to control home appliances.

2 Design and Implementation

2.1 Universal Interaction

In our approach, we call the protocol that can be universally used for the communication between input/output interaction devices and appliances *universal interaction. Universal interaction* enables us to control a variety of home appliances in a uniform way. This means that our behavior is not restricted according to where we are or which appliance we like to control. Therefore, our approach provides very natural interaction with home appliances.

The output events produced by appliances are converted to *universal output interaction events*, and the events are translated for respective output interaction devices. Also, input events generated in input interaction devices are converted to *universal input interaction events*, and the events are processed by applications executed in appliances.

A universal interaction proxy that is called the UniInt proxy described in the next section plays a role to convert between the universal interaction protocol and input/output events of respective interaction devices in a generic way. The proxy allows us to use any input/output interaction devices to control appliances if the events of the devices are converted to the universal interaction protocol. This approach offers the following three very attractive characteristics.

The first characteristic is that input interaction devices and output interaction devices are chosen independently according to a user's situation and preference. For example, a user can select their PDAs for their input/output interaction. Also, the user may choose his/her cellular phones as their input interaction devices, and television displays as his/her output interaction devices. The user may control appliances by his/her gesture by navigating augmented real world generated by wearable devices.

The second characteristic is that our approach enables us to choose suitable input/output interaction devices according to a user's preference. Also, these interaction devices are dynamically changed according to the user's current situation. For example, a user who controls an appliance by his/her cellular phone as an input interaction device will change the interaction device to a voice

Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02)
1063-6927/02 $17.00 © 2002 IEEE

input system because his both hands are busy for other work currently.

The third characteristic is that any applications executed in appliances can use the any user interface systems if the user interface systems speak the universal interaction protocol. In our approach, we currently adopt keyboard/mouse events as universal input events and bitmap images as universal output events. The approach enables us to use traditional graphical user interface toolkits such as Java AWT, GTK+, and Qt for interfacing with any interaction devices. In fact, most of standard specifications for consumer electronics like to recently adopt Java AWT for their GUI standards. Thus, our approach will allow us to control various future consumer electronics from various interaction devices without modifying their application programs. The characteristic is very desirable because it is very difficult to change existing GUI standards.

2.2 System Architecture

Our system uses the thin-client system to transfer bitmap images to draw graphical user interface, and to process mouse/keyboard events for inputs. The usual thin-client system consists of a viewer and a server. The server is executed on a machine where an application is running. The application implements graphical user interface by using a traditional user interface system such as the X window system. The bitmap images generated by the user interface system are transmitted to a viewer that are usually executed on another machine. On the other hand, mouse and keyboard events captured by the viewer are forwarded to the server. The protocol between the viewer and the server are specified as a standard protocol. In the paper, we call the protocol the *universal interaction protocol*. The system is usually used to move a user's desktop according to the location of a user[2], or shows multiple desktops on the same display, for instance, both MS-Windows and the X Window system.

In our system, we replace the viewer of a thin-client system to the UniInt(Universal Interaction) proxy that forwards bitmap images received from a UniInt server to an output device. In our approach, a server of any thin-client systems can be used as the UniInt server. Also, UniInt proxy forwards input events received from an input interaction device to the UniInt server.

Our system consists of the following four components. In the following paragraphs, we explain these components in details.

- Home Appliance Application
- UniInt Server
- UniInt Proxy
- Input/Output Devices

Home appliance applications generate a control panel for currently available appliances to control them. For example, if TV is currently available, the application generates user interface for the TV. On the other hand, the application generates the composed GUI for TV and VCR if both TV and VCR are currently available.

The UniInt server transmits bitmap images generated by a window system using the universal interaction protocol to a UniInt proxy. Also, it forwards mouse and keyboard events received from a UniInt proxy to the window system. In our current implementation, we need not to modify existing servers of thin-client systems, and any applications running on window systems supporting a UniInt server can be controlled in our system without modifying them.

The UniInt proxy is the most important component in our system. The UniInt proxy converts bitmap images received from a UniInt server according to the characteristics of output devices. Also, it converts events received from input devices to mouse or keyboard events that are compliant to the universal interaction protocol. The UniInt proxy chooses a currently appropriate input and output interaction devices for controlling appliances. Then, the selected input device transmits an input plug-in module, and the selected output device transmits an output plug-in module to the UniInt proxy. The input plug-in module contains a code to translate events received from the input device to mouse or keyboard events. The output plug-in module contains a code to convert bitmap images received from a UniInt server to images that can be displayed on the screen of the target output device.

The last component is input and output interaction devices. An input device supports the interaction with a user. The role of an input device is to deliver commands issued by a user to control home appliances. An output device has a display device to show graphical user interface to control appliances.

In our approach, the UniInt proxy plays a role to deal with the heterogeneity of interaction devices. Also, it can switch interaction devices according to a user's situation or preference. This makes it possible to personalize the interaction between a user and appliances.

3 Conclusion

This paper has described a new user interface system to fill the gap between traditional graphical user interface systems and advanced input/output interaction devices for networked home computing. We have also described the effectiveness of our system by demonstrating our system to control our home computing system.

References

[1] W. Ark, D. Christopher Dryer, D.J. Lu, "The Emotion Mouse", In Proceeding of the HCI International conference, 1999.

[2] Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, Paul Webster, "The Anatomy of a Context-Aware Application", In Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, 1999.

[3] HAVi Consortium, "HAVi Specification: Specification of the Home Audio/Video Interoperability(HAVi) Architecture, http://www.havi.org/

[4] H. Ishii, B.Ullmer, "Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms", In Proceedings of Conference on Human Factors in Computing Systems,1997.

[5] T. Nakajima, "System Software for Audio and Visual Networked Home Appliances on Commodity Operating Systems", In Proceedings of the IFIP/ACM International Conference on Distributed System Platforms - Middleware 2001, 2001.

[6] Mark Weiser, "The Computer for the 21st Century", Scientific American, Vol. 265, No.3, 1991.