Personal Universal Controllers: Controlling Complex Appliances With GUIs and Speech

Jeffrey Nichols[*], Brad A. Myers[*], Michael Higgins[†], Joseph Hughes[†], Thomas K. Harris[*], Roni Rosenfeld[*], Kevin Litwack[*]

[*]School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
{jeffreyn, bam, tkharris, roni, klitwack}@cs.cmu.edu
http://www.cs.cmu.edu/~pebbles/puc/

[†]MAYA Design, Inc.
Suite 702
2100 Wharton Street
Pittsburgh, PA 15203
{higgins, hughes}@maya.com

ABSTRACT
We envision a future where each person will carry with them a *personal universal controller* (PUC), a portable computerized device that allows the user to control any appliance within their environment. The PUC has a two-way communication channel with each appliance. It downloads a specification of the appliance's features and then automatically generates an interface for controlling that appliance (graphical, speech, or both). In this demonstration we present a working PUC system that automatically generates graphical and speech interfaces, and controls real appliances, including a shelf stereo and a Sony camcorder.

Keywords: Handheld computers, remote controls, appliances, Pebbles, Universal Speech Interface (USI), personal universal controller (PUC)

INTRODUCTION
Common home and office appliances, such as copiers, camcorders, thermostats, and VCRs, are becoming increasingly complex. As these devices become more computerized with more functions, their user interfaces become more difficult to use [1]. At the same time, people are increasingly carrying portable computerized devices, such as mobile phones and personal digital assistants (PDAs) such as the Palm and PocketPC. Speech recognition is also becoming practical and successful. We are investigating how handheld devices and speech interfaces can be used to improve the interfaces to home and office appliances with an approach we call the *personal universal controller* (PUC).

A PUC provides an alternate interface with which the user interacts to control an appliance. Instead of being preprogrammed with a set of controls at the factory or requiring laborious manual programming by the user (like today's "universal remotes," such as the Philips Pronto), the PUC engages in two-way communication with the appliance to download a specification of the appliance's functions. This specification, written in an XML-based language that we have developed, is used by the PUC to *automatically generate* user interfaces for the complete functionality of the appliance. The user can then use the interface to control the appliance and receive feedback on the state of the appliance. Our preliminary studies suggest that users may be able to use the generated interfaces on handhelds *half the time* with *half the errors* for simple and complex tasks compared to the manufacturers' interfaces [5].

Our demonstration of the PUC system will show the automatic generation of graphical and speech interfaces, demonstrate how our interface generators use our appliance specification language to create interfaces, and show the interfaces controlling real, commercial appliances.

RELATED WORK
There are several related projects, but they differ from ours in key ways. The V2 standard group has demonstrated AAIML, a language that specifies the functions of an appliance, and interpreters that build working interfaces from this language [9]. Our system is more novel than V2's because we have built PUCs that can generate both speech and graphical interfaces from the same abstract appliance specification language. The graphical interfaces generated by the PUC also have more intuitive and interesting layouts than those generated by V2. Furthermore, the PUC is able to control a variety of actual complex appliances whereas V2 only demonstrated control of X10 and prototypes simulated on a desktop computer.

Other related systems are ICrafter [6] and the Total Access System [8]. ICrafter is capable of automatically generating interfaces on a limited level, but this was not the focus of their research. The Total Access System is similar, but focuses on creating interfaces for disabled persons.

ARCHITECTURE
The PUC system has four parts: appliances, controllers, a communication protocol that allows the appliance and controller to talk, and a specification language with which the

CHI 2003, April 5-10, 2003, Ft. Lauderdale, Florida, USA.
ACM 1-58113-637-4/03/0004

appliance describes its functions to the controller. An important goal of our project is control real appliances, which may not support our PUC communication protocol. To solve this problem, our architecture includes "adaptors": hardware and software that translate from an appliance's built-in communication protocol to our PUC protocol.

AUTOMATIC INTERFACE GENERATORS

We have built automatic interface generators that can create graphical interfaces on a PocketPC and speech interfaces using Universal Speech Interface (USI) techniques [7]. The graphical interface generator is written in C# using Microsoft's .NET Compact Framework and runs on both desktop computers and PocketPCs. The speech interface generator uses the Sphinx speech recognizer and is designed to run on a desktop computer. The algorithms used for generating these interfaces have been described elsewhere [3]. Both of these generators work from descriptions written in our appliance specification language.

SPECIFICATION LANGUAGE

Our XML-based specification language abstractly describes the functions of an appliance. We believe that this language contains enough information to build a usable interface for almost any appliance, because we designed it based on an examination of several hand-designed controller interfaces. From this we derived requirements that our language must satisfy in order to generate high-quality interfaces [4].

ADAPTORS

We have created adaptors that allow our PUCs to control many different real appliances. These adaptors consist of software and occasionally custom hardware that translate the appliance's built-in protocol to the PUC protocol. We have also created two classes of adaptors: those specific to a particular appliance, and those that function across a wide-range of appliances that use the same protocol. These are some of the adaptors that we have developed:

Audiophase Shelf Stereo

The Audiophase shelf stereo has no built-in communication protocol, so custom adaptor hardware was needed to translate to the PUC protocol. Our hardware mimics the IR codes of the stereo's remote control to send commands to the stereo, and then interprets the configuration of its LCD front-panel to determine the appliance's state.

Sony Camcorder

The Sony Camcorder has an IEEE 1394 communication port and supports the standard AV/C protocol. We built a software-only adaptor for this device.

HAVi Devices

Home Audio Video Interoperability (HAVi) is an emerging standard for controlling A/V equipment in the home. We have built an adaptor that uses HAVi to control a Mitsubishi VCR, and we are working on a general HAVi adaptor that can use the information that HAVi provides about an appliance to generate a PUC specification. This would allow a PUC to control any appliance that supported HAVi.

Software Media Players

We also have built adaptors to control the WinAmp media player and Windows Media Player, two popular applications for playing MP3 files. Although these applications are just software and do not have physical boxes, we believe it is valuable to control them because they are increasingly being used as components in stereo systems.

CONCLUSIONS

This demonstration shows how a personal universal controller could be used every day by people to control the appliances within their environment. We demonstrate PUCs that automatically generate graphical and speech interfaces, and show that our architecture and specification language are complete enough to control appliances that exist today.

ACKNOWLEDGEMENTS

This work was conducted as a part of the Pebbles [2] project. The speech interface was also conducted as part of the Universal Speech Interfaces project [7]. This work was funded in part by grants from NSF, Microsoft, General Motors and the Pittsburgh Digital Greenhouse, and equipment grants from Mitsubishi Electric Research Laboratories and VividLogic. The National Science Foundation funded this work through a Graduate Research Fellowship for the first author, and under Grant No. IIS-0117658. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation.

REFERENCES

1. Brouwer-Janse, M.D., Bennett, R.W., Endo, T., van Nes, F.L., Strubbe, H.J., and Gentner, D.R. "Interfaces for consumer products: "how to camouflage the computer?"" in *CHI'1992:* Monterey, CA: pp. 287-290.
2. Myers, B.A., "Using Hand-Held Devices and PCs Together." *Communications of the ACM*, 2001. 44(11): pp. 34-41.
3. Nichols, J., Myers, B.A., Higgins, M., Hughes, J., Harris, T.K., Rosenfeld, R., Pignol, M. "Generating Remote Control Interfaces for Complex Appliances," in *UIST 2002.* Paris, France: pp. 161-170.
4. Nichols, J., Myers, B.A., Higgins, M., Hughes, J., Harris, T.K., Rosenfeld, R., Shriver, S. "Requirements for Automatically Generating Multi-Modal Interfaces for Complex Appliances," in *ICMI 2002*. Pittsburgh, PA: pp 377-382
5. Nichols, J.W. "Using Handhelds as Controls for Everyday Appliances: A Paper Prototype Study," in *ACM CHI'2001 Student Posters.* 2001. Seattle, WA: pp. 443-444.
6. Ponnekanti, S.R., Lee, B., Fox, A., Hanrahan, P., and T.Winograd. "ICrafter: A service framework for ubiquitous computing environments," in *UBICOMP 2001.* pp. 56-75.
7. Rosenfeld, R., "Universal Speech Interfaces Web Site," 2002. http://www.cs.cmu.edu/~usi/.
8. Scott, N.G.G., I. "The Total Access System," in *CHI Extended Abstracts.* 2001. pp. 13-14.
9. Zimmermann, G., Vanderheiden, G., Gilman, A. "Prototype Implementations for a Universal Remote Console Specification," in *CHI'2002.* 2002. Minneapolis, MN: pp. 510-511.