AUTOMATIC DESIGN OF PERSONAL UNIVERSAL CONTROLLERS

Proposal to Intel For Funding
Applications, Interface and Media;
System Infrastructure

March 28, 2001

Brad A. Myers

Human Computer Interaction Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3891
(412) 268-5150
FAX: (412) 268-1266
bam+@cs.cmu.edu
http://www.cs.cmu.edu/~bam

Proposed amount:
$221,397 over 3 years

1. IDENTIFICATION

Please provide the following identification information:

Title of proposal:	Automatic Design of Personal Universal Controllers
Name of principal investigator:	Brad A. Myers
Name of university or institution:	Carnegie Mellon University
Department:	Human Computer Interaction Institute
Date of submission:	March 30, 2001
Technical committee:	AIM

2. RESEARCH OBJECTIVE

2.1 PROBLEM

We propose to investigate how various kinds of handheld devices can be used to control all kinds of home, office and factory equipment. Increasingly, home and office appliances, including televisions, VCRs, stereo equipment, refrigerators, washing machines, thermostats, light switches, clocks, telephones, copiers, and factory equipment, have embedded computers, and often come with remote controls. However, the trend has been that as appliances get more computerized with more features, their user interfaces get harder to use [Brouwer-Janse 1992]. Some appliances need thirty or more buttons to cover all of their functions. This complexity can make even relatively simple tasks, like setting the clock on a VCR, so difficult that people avoid them. Even simple appliances can be hard to use if the interface is unfamiliar to the user; when traveling, many people are stumped by the user interface for setting the alarm on the clocks in hotel rooms, so they arrange for a wake-up-call instead.

Meanwhile, another trend is that people are increasingly carrying computerized devices that can communicate. People have cell-phones, pagers, personal digital assistants (PDAs) such as the Palm Pilot or PocketPC, and even watches that can communicate using various wireless technologies including infrared (IR) and radio. The advent of the BlueTooth short-distance radio network [Haartsen 1998] is expected to enable many devices to communicate with other devices in the vicinity.

Our proposal is that when users point their own handheld at an alarm clock in a hotel room, at a photocopier in an office, at a machine tool in a factory, at a VCR at home, at a piece of test equipment in the field, or at almost any other kind of appliance, a description of control parameters will be sent to the handheld. The handheld will use this information to create an appropriate control panel, taking into account the properties of the controls that are needed, the properties of the handheld (the display type and input techniques available), and the properties of the user (what formats are familiar for various tasks, what language is preferred, how big the buttons should be based on whether the user prefers using a finger or a stylus). The user can then control the appliance using the handheld. The appliance will not need to dedicate as much in displays and buttons for the user interface, since it will only need to contain a description of its capabilities and storage for the current settings, along with hardware for wireless communication. The handheld software will use intelligent techniques to create useful and appropriate interfaces that are customized for each user.

The approach we propose will help ensure that the resulting user interfaces on the handheld will be easy to use. First, we are creating control panels by hand for various types of handhelds and various appliances. We will then perform user studies to validate and improve these designs. A preliminary study suggests that this has the opportunity to be spectacularly successful. For example, our prototype of control panels for a stereo and an office telephone enabled users to complete complex tasks in *one-half* the time and with *one-fifth the errors* compared to using the manufacturer's interfaces [Nichols 2001]. The hand-designed user interfaces will guide the design of a specification language that will describe the functionality of the appliance at a high level. The next phase of the research will be to create software that will automatically design interfaces from the specification language for different appliances. Throughout, we will be performing user studies to evaluate and improve the usability of the automatically generated interfaces.

We call our approach the "Personal Universal Controller" (PUC). A key feature is that, unlike so-called "universal remotes" that are available today such as the Phillips Pronto [Philips 2001], our proposed controller will be *self-programming*. Today's remotes are either pre-programmed in the factory with a subset of the features of some specific appliances, or else the user is required to laboriously hand-program the remote with each desired function of each appliance. In contrast, our proposed remote will engage in a two-way exchange with the appliance, first uploading a description of the appliance's functions, then creating a control panel automatically, and finally sending appropriate control signals to the appliance as the user operates the control panel.

2.2 MOTIVATION

An important motivation for the proposed research is the increasing complexity of consumer and business appliances. Most appliances have many unused features since consumers find it difficult to master the basic functions of some of these appliances, never mind the sophisticated features. One reason is that appliances must economize on buttons and displays, and therefore they often reuse the same buttons for multiple functions. Often, pressing and holding a button will perform a different operation than a quick tap, but usually there is no indication of this on the button's label. Many appliances have invisible temporal modes that change the meaning of buttons. Indicators of appliance state can be confusing. In addition, there is little standardization of the labeling, placement or behavior of the controls on consumer appliances. The increasing computerization of equipment can only make this problem worse, as it becomes cheaper to embed sophisticated computing in even the smallest appliances. Today enormous processing power can be very cheaply embedded in devices as simple as a light-switch, but providing a good user interface for that processor is still a significant expense. Many appliances today come with custom remote controls, which themselves may be poorly designed. Furthermore, the consumer is faced with the familiar problem of having dozens of incompatible remote controls to deal with.

A CHI'92 panel stated:

> User interfaces for consumer products are notoriously bad. … [The causes include] designs that are modeled after the user interface of computer systems, … [and a] rigid system of constraints on display size, memory, cpu power, input devices, conditions of use, component price, mechanical compatibility, manufacturability and serviceability. [Brouwer-Janse 1992]

Most appliances probably do not get the extensive usability testing [Nielsen 1993] that might help identify usability problems early in the design phase. Hugo Strubbe from Philips says:

> Human factors techniques have been tried successfully on TV interface prototypes. However, such work typically has little effect on products. Designs made by human factors people are often more expensive than those made by engineers. Cost is an important purchase criterion for consumers. They rarely evaluate ease-of-use in the shop. Therefore, one cannot charge extra for it. Consumers who are unable to use the product at home accept this as their fault and do not return the product. The consumer has to be taught to insist on ease-of-use, and our human factors work does not directly contribute to this [Brouwer-Janse 1992, p. 289].

2.3 MOTIVATION FOR AUTOMATIC DESIGN OF PANELS

The most interesting and challenging part of the proposed research is to *automatically* create panels of controls for appliances. Most remote controls today are pre-programmed in the factory for one or two appliances. For example, most VCR remotes can also control TVs. This approach is clearly not scaleable to many appliances, and the remotes often only cover the basic functions and omit useful features the consumer might want (e.g., my newest VCR remote does not have a "mute" button for the TV). The other approach used today for multi-device "universal" remotes requires the consumer to laboriously hand-program each function of the remote. For example, with the Phillips Pronto "universal" remote control [Philips 2001], and Pacific Neo-Tek's "Omni-Remote" software [Pacific Neo-Tek 2000] for the Palm, the user must have the old remote for the appliance to be controlled. The old remote is aimed at the controller, and, for each operation to be provided on the controller, the user must specify a button on the controller and then push the corresponding button on the old remote. In contrast, the proposed PUC will be able to create a control panel for new appliances without requiring any programming by the user.

One approach might be to have the appliance send a fully-designed control panel to the remote control. This is similar to the philosophy of Jini [Sun 2000], which sends a complete implementation of the user interface for the requested service. For the appliance control envisioned here, sending a pre-defined panel would not work because of the wide range of handheld mobile devices, with vastly different input and output capabilities. We want to support cell-phones with 8-line displays and 14 buttons, monochrome and color Palm OS devices with 160x160 pixels and a touch screen, color Pocket PC devices with 240 x 320 pixels, etc. In the Jini model, the appliance would need to provide many different control panels, each tuned for a different kind of handheld. Another problem is that each type of handheld may have its own conventions for how interactions should be performed, and the downloaded interface would not be able to take these into account. Furthermore, the appliances we want to control are expected to last for many years whereas the features of handhelds change every year. For example, we would want the same appliance that can be controlled by a PDA to work with a future controller built into a watch, which might have a new kind of round display and only four buttons. Finally, this approach does not allow the designs to be customized to be familiar to the user, since each appliance will have their own independent design.

Therefore, we propose to have the appliance send to the controller a high-level specification of the parameters and operations of the device, and have the controller automatically create a user interface based on this specification. This will enable the same specification to be used by controllers with significantly different characteristics. The automatic generation of the user interface will take into account the input and output capabilities of the controller along with user preferences and other information. Users might specify, for example, that they want bigger buttons and bigger labels because they want to press them with a finger, or alternatively that they prefer smaller buttons so more will fit on the screen at the same time. Since the display on the handheld is likely to be bigger than on the appliance, longer names and even help text can appear on the controller. If automatic machine language translation is available, the user might be able to request a French or Japanese interface, and have the textual labels translated by the controller.

Another important advantage of automatic generation is that the controller can impose consistency across all the appliances that are controlled. For example, both a phone and a clock radio have a volume control, so panels for both should use the same widget for volume and put it in the same place. A controller on a Pocket PC will use the standard widget for setting the volume (a slider next to a speaker icon), but on a Palm, the volume usually is represented as a pop-up with a few choices. Similarly, when the controller is pointed at a clock from any manufacturer, the same control panel should appear on the controller.

Dynamically created user interfaces to networked appliances, such as telephones, enable new kinds of services to be automatically incorporated into the user interface even after the appliances have been deployed. For example, if the phone system starts offering a new service, this might automatically appear as a new, labeled button on the PUC, rather than requiring the user to remember to press some arbitrary three-digit code.

2.4 RESEARCH HYPOTHESIS

We hypothesize that automatically created control panels on a personal universal controller for appliances can be easier to use than the manufacturer's interfaces, possibly approaching the quality of carefully hand-designed user interfaces.

2.5 TESTING AND VERIFICATION

An important component of the proposed research will be to perform user tests throughout the development. We are currently performing user tests on hand-designed user interfaces.

First, we did a paper-prototype study to see if we were on the right track. We compared the manufacturer's interface to a paper-prototype [Rettig 1994] of our Palm interfaces for the AT&T telephone and the Aiwa stereo. (More details of the experiment are published elsewhere [Nichols 2001].) Figure 1 shows four screens used in the study. We chose these two appliances because both are common and combine several functions into a single appliance. The Palm interfaces cover all of the features of these appliances, as described in their user manuals. Subjects were asked to work through one list of tasks for the stereo and another list for the phone. We recorded the number of missteps and the number of times external help was required while the tasks were performed.

The results of the study indicate that subjects made fewer missteps and asked for help less using the prototype handheld interfaces than using the actual appliances. All of the results are statistically significant at the $p < 0.001$ level using a unpaired t-test. On average, the Palm user interface subjects made about 1/5 the errors (missteps). This indicates that the prototype handheld interfaces were more intuitive to use than the actual interfaces. Further, the average time to complete the tasks was about twice as long with the real appliance interface as with the Palm prototypes.

Encouraged by these results, we are now performing a comparison of hand-designed control panels for the same devices running on a Compaq iPaq handheld running the Microsoft PocketPC operating system. These handhelds have bigger screens, and our implementation (in Embedded Visual Basic) is revealing some interesting issues about the feedback on remote controls.

Next, we will be working on the software to automatically create interfaces. We will continually test and refine the automatically generated interfaces in tests similar to those above. In addition to the stereo and telephone, we propose to test a copier and other appliances. We hope our sponsors will help us decide on the appropriate devices to study.

Figure 1. The actual devices (left) and some hand-drawn control panels for the Palm for the AT&T phone (top) and AIWA stereo (bottom).

2.6 POTENTIAL FOR INDUSTRY

Having a working Personal Universal Controller will have significant impacts on industry, and may significantly increase the use of Intel processors in appliances. Manufacturers will be motivated to make smarter appliances, since users will be more able to use the advanced features. People will get more benefit out of their appliances and will be likely to upgrade to models supporting controls by PUCs. Sales of handhelds that can be used as PUCs would also expand significantly.

If the proposed research is successful, it will have benefits beyond just remote control devices for appliances. The proposed research will help further the cause of separating the user interface from the application code, which has been a basic goal of user interface software research from the beginning. Using a high-level specification of the functionality to guide the automatic design of the user interface may be useful for many other kinds of applications, besides remote controllers. Future mobile and handheld devices will have increasingly diverse user interface characteristics, and yet can run the same software (for example, if the software is written in a portable language such as Java). Therefore, it will be increasingly important for the user interface to automatically adapt to whatever platform it is on. The techniques developed as part of the proposed research may identify ways to create a more general user interface generator for many portable computerized applications.

3. RELATIONSHIP TO OTHER RESEARCH & PRACTICE

3.1 RELATED WORK

There are many different classes of systems that are related to the proposed work.

A number of research groups are working on controlling appliances from handheld devices. Hodes proposes a similar idea to our PUC, which is called a “universal interactor” that can adapt itself to control many appliances [Hodes 1997]. However, their research seems to have focused on the system and infrastructure issues rather than how to create the user interfaces. Their later paper describes the “rvic” system [Hodes 1999] that allows a Palm pilot or laptop to remotely control the audio/video equipment in a meeting room, but the control panels are hand-designed and hard-coded into the Palm program. The Stanford iRoom project [Fox 2000] also supports using the PDA as a remote control, and they tried two designs: one with the remote control hand-coded on the Palm, and the other using Web forms displayed by a standard Web browser on the handheld. In both cases, the programmer designed the control panels in advance. An IBM project [Eustice 1999] describes a “Universal Information Appliance” (UIA) that might be implemented on a PDA. The UIA uses an XML-based language called MoDAL from which it creates a user interface panel for accessing information. However, the MoDAL processor apparently only handles simple layouts and its only type of input control is text strings.

The Xweb [Olsen Jr. 2000] project is working to create technologies that can create customized interfaces that are appropriate to the interests of the user. The goal is to separate the functionality of the appliance from the device upon which it is displayed. Xweb defines an XML language from which user interfaces can be created. Another XML language for user interface design is UIML [Abrams 1999], from which user interfaces can also be created.

Other projects have looked at the general issues with PDAs and stationary devices working together, including the original Xerox ParcTab [Want 1995] system, Rekimoto’s many systems [Rekimoto 1997][Rekimoto 1998][Rekimoto 1999], and our Pebbles system [Myers 1998][Myers 2000a][Myers 2000b][Myers 2001]. In these, the user interfaces for the PDA have been hand-designed.

With respect to automatic design of user interfaces, the WML language for WAP phones is relevant, since it leaves some aspects of the user interface for the phone to decide. However, in practice most of the design must be included in the WML specification. There were a number of research systems that looked at automatic design of user interfaces for conventional computers. These sometimes went under the name of “model-based” systems [Szekely 1993]. Here, the programmer provides a specification (“model”) of the properties of the application, along with specifications of the user and the display. This approach was moderately successful at automatically creating dialog boxes [Kim 1993] [Vander Zanden 1990] and creating complete interfaces in a limited range [Olsen Jr. 1989] [Frank 1993] [Szekely 1993]. The ITS system from IBM was used to create all the screens for the information kiosks at the EXPO’92 worlds fair [Wiecha 1989][Wiecha 1990]. Of particular note is the layout algorithm in the DON system that achieved a pleasing, compact, and logical placement of the controls [Kim 1993]. Other systems focused on the initial creation assuming a designer would edit the resulting user interface [Foley 1988] [Singh 1989]. We plan to extend these results to create panels of controls on handhelds.

One problem with many of these model-based systems was that the automatically created user interface was often not as good as a person could create, and therefore required intervention to fix the resulting interface or else fix the rules with special cases. Another problem is that the specification became quite large with much extra information needed for the layout algorithm to do a good job. Writing this specification became quite burdensome to the designer, often more difficult than just programming the interface by hand. We feel that our proposed system will avoid these problems because we will start off with high-quality user interfaces from which the specification language features will be designed, and we can use the restrictions of the control panels to simplify the specifications. Controller interfaces are simpler than other kinds of interfaces since they are constructed mainly of buttons and selection lists, grouped appropriately on the screen. Understanding the groupings solves much of the layout problem, which is a key issue that will be addressed in our specification language.

3.2 LONG TERM RESEARCH

The problem of automatically creating user interfaces has clearly been studied for many years, with only minor successes, and has had little impact on commercial approaches. This long-standing unsolved problem will be the focus of the proposed research.

3.3 OUR RESEARCH STRENGTHS

We feel we can make headway on this difficult research problem due to our significant contributions to many areas of user interface software and intelligent user interfaces. The research proposed here will be performed as part of the Pebbles project [Myers 2001] (http://www.pebbles.hcii.cmu.edu), which is investigating the many ways that handheld devices can be used at the same time as other computerized devices. As part of the Pebbles project, we have looked at multiple PDAs connected to a PC to support meetings [Myers 1998]. In other research, we are investigating how a PDA can be equally useful for an individual to augment the Windows user interface for desktop applications [Myers 2000b]. We performed a study that showed, for example, that the PDA could be used very effectively as a scrolling device for desktop applications [Myers 2000a]. The software we have developed as part of the Pebbles project is mostly available for free download from our web site. One Pebbles application, the SlideShow Commander, was recently licensed to a company for commercial sale (http://www.slideshowcommander.com) and has sold thousands of copies already. We anticipate that the technology developed as part of the proposed research will similarly be widely distributed for free or commercially.

We have also worked in the area of User Interface Software. As part of the Garnet system developed in the late 1980s, Brad Vander Zanden and I investigated automatic creation of user interfaces [Vander Zanden 1990] which is of particular relevance to the proposed research.

4. RESEARCH POTENTIAL IMPACT

The exciting vision of the Personal Universal Controller proposed by this research can clearly not happen until all appliance manufacturers adopt a consistent standard for communicating with the PUC, and agree on the specification language for the appliances. It is not expected that this would happen during this research project. However, the goal of this project is to investigate the format and content of such a specification and algorithms that might be used to generate the interfaces. We hope that this research will help motivate the manufacturers to adopt such a standard, and we would be willing to help with the standardization effort.

In the short term, there will clearly not be appliances that are built to communicate with our Personal Universal Controller. We will be able to test and evaluate our research progress using simulations of the specifications that the appliances would return, and then simulations of the control of the appliances. However, it will be much more interesting and exciting to be able to demonstrate the actual control of some appliances. Increasingly, we expect this to be possible. For example, a 3Com Corporation project has created a Palm remote controller for an Internet phone [Dalgic 2000] which we might be able to use. With BlueTooth, more and more devices should be able to participate in two-way communication with a PUC. We hope our sponsors will help us find devices that we can control.

5. RESEARCH PLAN

5.1 DELIVERABLES

Deliverables from this effort will include:

- Hand-designed screens for various devices, along with prototype implementations of the screens for testing purposes.
- The general specification language for device parameters, which might serve as the beginnings of a standard.
- A software application running on multiple platforms (probably Palm, PocketPC, and maybe WAP phone) that takes the specification language and produces high-quality user interfaces.
- Simulated and actual control of devices using the generated control panels.
- Papers and reports describing the research results.

5.2 SCHEDULE

In the first six months of the research, we propose to further study hand-designed control panels for various kinds of PUC devices and for various platforms. From this, we will spend six months developing a specification language that can adequately represent the complexities of the control panels needed for real appliances. In the second year, we will begin the design and implementation of the automatic layout algorithms. At the same time, we will begin working with some real appliances that can be remote controlled, and working on having them describe their features using our specification language. In the third year, we plan to port the automatic generation algorithm to multiple platforms with different characteristics, and to demonstrate remote control of real devices. Throughout, we will be performing user studies of the interfaces to evaluate and refine the designs.

5.3 TECHNOLOGY TRANSFER

Our research group has a long history of producing systems that are made generally available and are widely used. The Amulet toolkit (http://www.cs.cmu.edu/~amulet) was used by hundreds of research and commercial projects and is still being downloaded over 100 times a month. Our current Pebbles project has produced software that is being downloaded about 200 times a week, and one application has been licensed for commercial sale. We expect that the results of the proposed work to also be widely distributed. We would work with manufactures and our sponsors to identify ways to get more appliances to support a specification like those we develop to enable PUCs.

5.4 PERSONNEL AND BUDGET

The key personnel involved will be Brad Myers and PhD student Jeff Nichols.

Budget for Intel Proposal		**Year One** 6/1/2001 5/31/2002	**Year Two** 6/1/2002 5/31/2003	**Year Three** 6/1/2003 5/31/2004	**Project Totals**
Personnel	**Term**				
Graduate Student (SCS)	Academic Year	$40,257	$42,273	$44,388	$126,918
Graduate Student (SCS)	Summer	$4,950	$5,199	$5,457	$15,606
Myers, Brad at 7.5%	Calendar Year	$8,843	$9,154	$9,477	$27,474
Total Personnel		**$54,050**	**$56,626**	**$59,322**	**$169,998**
FT Benefits		$2,157	$2,237	$2,343	$6,737
Total Personnel and Fringe		**$56,207**	**$58,863**	**$61,665**	**$176,735**
Operating Expenses					
Copying and Publishing 0.20%		$114	$114	$126	$354
Long Distance Calls 1.00%		$563	$585	$615	$1,763
Postage 0.20%		$114	$114	$126	$354
Computing Facilities 6.90%		$3,877	$4,063	$4,251	$12,191
Total Operating Expenses		**$4,668**	**$4,876**	**$5,118**	**$14,662**
Overhead 56.00%	*not charged*	--	--	--	--
HARDWARE		**$10,000**	**$10,000**	**$10,000**	**$30,000**
TOTAL SPONSOR COSTS		**$70,875**	**$73,739**	**$76,783**	**$221,397**

6. ADMINISTRATIVE MATTERS

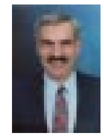
All prototypes, papers and results of the proposed research will be made available to Intel for their internal evaluation. The intellectual property resulting from this research will be owned by Carnegie Mellon University, and we will negotiate with our sponsors about licensing.

Here is the administrative information:

Researcher's Name	Brad A. Myers, Senior Research Scientist
Researcher's Organization:	Carnegie Mellon University, http://www.cmu.edu/
Researcher's Department:	Human Computer Interaction Institute http://www.hcii.cmu.edu/
Researcher's Contact:	Human Computer Interaction Institute School of Computer Science Carnegie Mellon University Pittsburgh, PA 15213-3891 Phone: (412) 268-5150, FAX: (412) 268-1266 Email: bam+@cs.cmu.edu www: http://www.cs.cmu.edu/~bam
University Foundation or Technology Development Office authorized to process the grant/gifts and intellectual property issues: address, phone, email.	Robert Kearns Sr. Research Administrator Office of Sponsored Research Carnegie Mellon University 5000 Forbes Avenue

	Pittsburgh, PA 15213 tel: 412-268-5837, fax: 412-268-5841 email: rk2a@andrew.cmu.edu
Shipping address and contact for possible donation of equipment	*same as Researcher's contact*

7. PERSONAL DATA



Brad A. Myers is a Senior Research Scientist in the Human-Computer Interaction Institute in the School of Computer Science at Carnegie Mellon University, where he is the principal investigator for various research projects including: the Pebbles Handheld Computer Project, Silver Multi-Media Authoring, Natural Programming, User Interface Software, and Demonstrational Interfaces. He is the author or editor of over 200 publications, and he is on the editorial board of five journals. He has been a consultant on user interface design and implementation to over 40 companies, and regularly teaches courses on user interface software. Myers received a PhD in computer science at the University of Toronto where he developed the Peridot UIMS. He received the MS and BSc degrees from the Massachusetts Institute of Technology during which time he was a research intern at Xerox PARC. From 1980 until 1983, he worked at PERQ Systems Corporation. His research interests include handheld computers, user interface development systems, user interfaces, programming by example, programming languages for kids, visual programming, interaction techniques, window management, and programming environments. He belongs to SIGCHI, ACM, IEEE Computer Society, IEEE, and Computer Professionals for Social Responsibility. URL: http://www.cs.cmu.edu/~bam

7.1 TEN RELATED PUBLICATIONS:

Brad Vander Zanden and Brad A. Myers. "Automatic, Look-and-Feel Independent Dialog Creation for Graphical User Interfaces," *Human Factors in Computing Systems*, Proceedings SIGCHI'90. Seattle, WA, Apr, 1990. pp. 27-34.

Brad A. Myers. "Using Multiple Devices Simultaneously for Display and Control." *IEEE Personal Communications*, vol. 7, no. 5, Oct. 2000. pp. 62-65.

Brad A. Myers, Robert C. Miller, Benjamin Bostwick, Franklin Chen, Carl Evankovich, Herb Stiel, Karen Cross, Adrienne Warmack, Chun-Kwok Lee, YuShan Chuang, Marsha Tjandra, Monchu Chen, Kin Pou Lie, Bo-chieh Yang and Zack Rosen. "Using Handheld Devices and PCs Together," *Communications of the ACM*. Accepted for publication. To appear.

Brad A. Myers. "User Interface Software Tools," *ACM Transactions on Computer-Human Interaction*. vol. 2, no. 1, March, 1995. pp. 64-103.

Brad A. Myers, *et. al.*, "The Amulet Environment: New Models for Effective User Interface Software Development," *IEEE Transactions on Software Engineering*, vol. 23, no. 6, June, 1997, pp. 347-365.

Brad A. Myers, ed. *Languages for Developing User Interfaces*. Boston: Jones and Bartlett, 1992.

Brad A. Myers. *Creating User Interfaces by Demonstration*. Boston, MA: Academic Press, 1988.

Brad Myers, Scott E. Hudson, and Randy Pausch, "Past, Present and Future of User Interface Software Tools," *ACM Transactions on Computer Human Interaction*. March, 2000. vol. 7, no. 1. pp. 3-28.

Brad A. Myers, Herb Stiel, and Robert Gargiulo. "Collaboration Using Multiple PDAs Connected to a PC," *Proceedings CSCW'98: ACM Conference on Computer-Supported Cooperative Work*, November 14-18, 1998, Seattle, WA. pp. 285-294.

Brad Myers, Kin Pou ("Leo") Lie and Bo-Chieh ("Jerry") Yang, "Two-Handed Input Using a PDA and a Mouse," *Proceedings CHI'2000: Human Factors in Computing Systems*. April 1-6, 2000. The Hague, The Netherlands. pp. 41-48.

8. REFERENCES

[Abrams 1999] Marc Abrams, Constantinos Phanouriou, Alan L. Batongbacal, Stephen M. Williams and Jonathan E. Shuster. “UIML: An Appliance-Independent XML User Interface Language,” The Eighth International World Wide Web Conference, Toronto, Canada, May 11-14, 1999.http://www8.org/ and http://www.uiml.org/

[Brouwer-Janse 1992] Maddy D. Brouwer-Janse, Raymond W. Bennett, Takaya Endo, Floris L. van Nes, Hugo J. Strubbe and Donald R. Gentner. “Interfaces for consumer products: "how to camouflage the computer?"” CHI'1992: Human factors in computing systems, Monterey, CA, May 3 - 7, 1992. pp. 287-290.

[Dalgic 2000] Ismail Dalgic, Michael Borella, Rick Dean, Jacek Grabiec, Jerry Mahler, Guido Schuster and Ikhlaq Sidhu. True Number Portability and Advanced Call Screening in a SIP-Based IP Telephony System. 3Com Corporation. http://www.3com.com/technology/tech_net/white_papers/503054.html. March 14, 2000.

[Eustice 1999] K.F. Eustice, T. J. Lehman, A. Morales, M. C. Munson, S. Edlund and M. Guillen. “A Universal Information Appliance,” IBM Systems Journal. 1999. 38(4). pp. 575-601. http://www.research.ibm.com/journal/sj/384/eustice.html

[Foley 1988] James D. Foley, Christina Gibbs, Won Chul Kim and Srdjan Kovacevic. “A Knowledge-Based User Interface Management System,” Human Factors in Computing Systems, Proceedings SIGCHI'88. Washington, D.C., May, 1988, 1988. pp. 67-72.

[Fox 2000] Armando Fox, Brad Johanson, Pat Hanrahan and Terry Winograd. “Integrating information appliances into an interactive workspace,” IEEE Computer Graphics and Applications. 2000. 20(3). pp. 54-65.

[Frank 1993] Martin R. Frank and James D. Foley. “Model-Based User Interface Design by Example and by Interview,” ACM SIGGRAPH Symposium on User Interface Software and Technology, Proceedings UIST'93. Atlanta, GA, Nov, 1993, 1993. pp. 129-137.

[Haartsen 1998] Jaap Haartsen, Mahmoud Naghshineh, Jon Inouye, Olaf J. Joeressen and Warren Allen. “Bluetooth: Vision, Goals, and Architecture,” ACM Mobile Computing and Communications Review. 1998. 2(4). pp. 38-45. Oct. www.bluetooth.com.

[Hodes 1997] Todd D. Hodes, Randy H. Katz, Edouard Servan-Schreiber and Lawrence Rowe. “Composable ad-hoc mobile services for universal interaction,” Proceedings of the Third annual ACM/IEEE international Conference on Mobile computing and networking (ACM Mobicom'97), Budapest Hungary, September 26 - 30, 1997. pp. 1 - 12.

[Hodes 1999] Todd Hodes, Mark Newman, Steven McCanne, Randy H. Katz and James Landay. “Shared Remote Control of a Video Conferencing Application: Motivation, Design, and Implementation,” Proceedings of SPIE Multimedia Computing and Networking, San Jose, CA, January, 1999. pp. 17-28. http://daedalus.cs.berkeley.edu/publications/mmcn99.ps.gz

[Kim 1993] Won Chul Kim and James D. Foley. “Providing High-level Control and Expert Assistance in the User Interface Presentation Design,” Human Factors in Computing Systems, Proceedings INTERCHI'93. Amsterdam, The Netherlands, Apr, 1993. pp. 430-437.

[Myers 2000a] Brad A. Myers, Kin Pou ("Leo") Lie and Bo-Chieh ("Jerry") Yang. “Two-Handed Input Using a PDA And a Mouse,” Human Factors in Computing Systems, Proceedings CHI'2000. The Hague, The Netherlands, Apr 1-6, 2000a. pp. 41-48.

[Myers 2001] Brad A. Myers, Robert C. Miller, Benjamin Bostwick, Franklin Chen, Carl Evankovich, Herb Stiel, Karen Cross, Adrienne Warmack, Chun-Kwok Lee, YuShan Chuang, Marsha Tjandra, Monchu Chen, Kin Pou Lie, Bo-chieh Yang and Zack Rosen. “Using Hand-Held Devices and PCs Together,” ACM Communications of the ACM. 2001. p. To appear.

[Myers 2000b] Brad A. Myers, Robert C. Miller, Benjamin Bostwick and Carl Evankovich. “Extending the Windows Desktop Interface With Connected Handheld Computers,” 4th USENIX Windows Systems Symposium, Seattle, WA, August 3-4, 2000b. pp. 79-88.

[Myers 1998] Brad A. Myers, Herb Stiel and Robert Gargiulo. “Collaboration Using Multiple PDAs Connected to a PC,” Proceedings CSCW'98: ACM Conference on Computer-Supported Cooperative Work, Seattle, WA, November 14-18, 1998. pp. 285-294. http://www.cs.cmu.edu/~pebbles

[Nichols 2001] Jeffrey W. Nichols. “Using Handhelds as Controls for Everyday Appliances: A Paper Prototype Study,” ACM CHI'2001 Student Posters, Seattle, WA, March 31-April 5, 2001. p. To appear.

[Nielsen 1993] Jakob Nielsen. Usability Engineering. Boston, Academic Press. 1993.

[Olsen Jr. 1989] Dan R. Olsen Jr. “A Programming Language Basis for User Interface Management,” Human Factors in Computing Systems, Proceedings SIGCHI'89. Austin, TX, Apr, 1989, 1989. pp. 171-176.

[Olsen Jr. 2000] Dan R. Olsen Jr., Sean Jefferies, Travis Nielsen, William Moyes and Paul Fredrickson. “Cross-modal Interaction using Xweb,” Proceedings UIST'00: ACM SIGGRAPH Symposium on User Interface Software and Technology, San Diego, CA, 2000. pp. 191-200.

[Pacific Neo-Tek 2000] Pacific Neo-Tek. OmniRemote. http://www.pacificneotek.com. 2000.

[Philips 2001] Philips. Pronto Intelligent Remote Control. Philips Consumer Electronics. 2001. http://www.pronto.philips.com/

[Rekimoto 1997] Jun Rekimoto. “Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments,” ACM SIGGRAPH Symposium on User Interface Software and Technology, Proceedings UIST'97. Banff, Alberta, Canada, Oct 14-17, 1997. pp. 31-39.

[Rekimoto 1998] Jun Rekimoto. “A Multiple Device Approach for Supporting Whiteboard-based Interactions,” Human Factors in Computing Systems, Proceedings SIGCHI'98. Los Angeles, CA, Apr, 1998. pp. 344-351.

[Rekimoto 1999] Jun Rekimoto and Masanori Saitoh. “Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments,” Human Factors in Computing Systems, Proceedings SIGCHI'99. Pittsburgh, PA, May, 1999. pp. 378-385.

[Rettig 1994] Marc Rettig. “Prototyping for Tiny Fingers,” Comm. ACM. 1994. 37(4). pp. 21-27.

[Singh 1989] Gurminder Singh and Mark Green. “A High-Level User Interface Management System,” Human Factors in Computing Systems, Proceedings SIGCHI'89. Austin, TX, Apr, 1989, 1989. pp. 133-138.

[Sun 2000] Sun. Jini Connection Technology. Sun Microsystems. http://www.sun.com/jini/. 2000.

[Szekely 1993] Pedro Szekely, Ping Luo and Robert Neches. “Beyond Interface Builders: Model-Based Interface Tools,” Human Factors in Computing Systems, Proceedings INTERCHI'93. Amsterdam, The Netherlands, Apr, 1993. pp. 383-390.

[Vander Zanden 1990] Brad Vander Zanden and Brad A. Myers. “Automatic, Look-and-Feel Independent Dialog Creation for Graphical User Interfaces,” Human Factors in Computing Systems, Proceedings SIGCHI'90. Seattle, WA, Apr, 1990. pp. 27-34.

[Want 1995] Roy Want, Bill N. Schilit, Norman Adams, Rich Gold, Karin Petersen, David Goldberg, John R. Ellis and Mark Weiser. “An Overview of the ParcTab Ubiquitous Computing Experiment,” IEEE Personal Communications. 1995. pp. 28-43. December. Also appears as Xerox PARC Technical Report CSL-95-1, March, 1995.

[Wiecha 1989] Charles Wiecha, William Bennet, Stephen Boies and John Gould. “Generating user interfaces to highly interactive applications,” Human Factors in Computing Systems, Proceedings SIGCHI'89. Austin, TX, Apr, 1989, 1989. pp. 277-282.

[Wiecha 1990] Charles Wiecha, William Bennett, Stephen Boies, John Gould and Sharon Greene. “ITS: A Tool for Rapidly Developing Interactive Applications,” ACM Transactions on Information Systems. 1990. 8(3). pp. 204-236.