ECT: A Toolkit to Support Rapid Construction of Ubicomp Environments

Chris Greenhalgh, Shahram Izadi, James Mathrick, Jan Humble, Ian Taylor
School of Computer Science and IT, University of Nottingham, Jubilee Campus, Wollaton Road, Nottingham, NG8 1BB, UK
{cmg, sxi, jym, jch, imt}@cs.nott.ac.uk

ABSTRACT
We present the initial release of a freely available software toolkit for ubiquitous computing that can reduce the cost – especially in time and effort – of developing, deploying and managing ubicomp installations, applications and environments, and increases the potential involvement of designers and users in this process.

Keywords
Toolkits, deployment, configuration, management, end-user programming.

INTRODUCTION

Over the last three-to-four years we have been involved in a large multi-site initiative to explore the development, deployment and use of ubiquitous computing environments. This has involved the realization of a number of distinct user experiences including the development and placement of technologies within different settings, including the home [1,4]. These experiences have involved the assembly of heterogeneous collections of devices (and platforms), the placement of these devices within a given space and the development of facilities and applications that allow these devices to interconnect in a variety of different ways; this has required considerable effort and diverse expertise. In response to this we have designed and implemented a toolkit – the Equip Component Toolkit (ECT) – that greatly simplifies this process, and that consequently allows greater involvement of non-programmers (e.g. media designers) and end-users.

TOOLKIT MODEL

The toolkit considers a ubicomp application or experience to be supported by a dynamically interconnected and potentially time-varying collection of hardware and software components, distributed across a number of machines. A *component* is the basic unit of deployment, management and coordination in the toolkit, being a specific unit of functionality with a well-defined interface. These components can be dynamically created within the running system, for example when a Phidget is connected to the USB port a corresponding IO component appears.

Components are managed within container applications, and inter-container coordination is supported via a shared dataspace (as used in [4]). The main inter-component coordination mechanism is to slave a property on one component to a property on another component, so that property changes are propagated between the two. The use of a shared dataspace for coordination allows very flexible introspection of the running system, and distributed manipulation of component requests and property links allows active reconfiguration and adaptation of the running system.

On top of this fundamental component management layer, more abstract facilities and views can be developed, for example relating the various components to their physical setting and intended purpose.

IMPLEMENTATION

The current toolkit implementation has two Containers, one in Java (which hosts JavaBeans as Components) and one in C#. Figure 1 shows a sample application that is readily assembled using the current toolkit; this is a tangible media viewer in the style of [7], but assembled graphically from standard re-usable components which can be developed independently of the toolkit itself. The Phidget [3] components are C# objects; the others are Java. The Phidget Host Manager component is an example of a device driver component, and dynamically instantiates the specific Phidget components as they plugged into the USB port. The Simple Association Learner is an application behaviour/glue component that is instantiated on demand for this particular application. As the system is used it 'learns' to associate particular RFID tag codes with particular media file URLs (which are first generated by the File Exporter component when the user publishes a media file). The media associated with the current RFID tag is shown on the display, and can be navigated using the physical slider.

DISCUSSION

A good software toolkit reduces the cost of application development through code reuse, modularity and tool support. For example, the Context Toolkit [6] adopts a widget approach to the programmatic construction of ubicomp applications. The various iRoom activities [5] use a tuple-space to share simple self-describing events to support the assembly of "smart room" systems. Speakeasy [2] proposes a combination of standard interfaces, mobile code and user involvement to support opportunistic interactions between ubicomp devices. Our own approach emphasizes support for legacy and non-toolkit components (such as ordinary JavaBeans), rather than requiring coding specifically for the toolkit as in these other systems. We also emphasise introspection of the running system (both components and interactions) and dynamic and interactive reconfiguration to a greater extent than these approaches. This work builds on [4], but greatly enhances its flexibility, richness of representation and range of facilities and interfaces.

The toolkit is publicly available and we encourage others to use it and contribute components, interfaces and other extensions: http://www.crg.cs.nott.ac.uk/~cmg/Equator/#ect

ACKNOWLEDGMENTS

We thank our colleagues in the UK EPSRC-funded EQUATOR IRC, and especially those who are working with and contributing components to the toolkit.

REFERENCES

1. Crabtree, A., Rodden, T., Hemmings, T. and Benford, S. (2003) "Finding a place for UbiComp in the home", Proceedings of the 5th Annual Conference on Ubiquitous Computing, October 12th 15th, Seattle:Springer.
2. W. Keith Edwards, Mark W. Newman, Jana Z. Sedivy, Trevor F Smith, and Shahram Izadi. Challenge: Recombinant Computing and the Speakeasy Approach. Proceedings of the Eighth ACM International Conference on Mobile Computing and Networking (MobiCom 2002). Atlanta, GA. September 23-28, 2002.
3. Greenberg, S. and Fitchett, C. (2001) Phidgets: Easy Development of Physical Interfaces through Physical Widgets. Proceedings of the UIST 2001 14th Annual ACM Symposium on User Interface Software and Technology, November 11-14, Orlando, Florida, p209-218, ACM Press
4. Humble, J., Crabtree, A., Hemmings, T., Åkesson, K-P., Koleva, B., Rodden, T., Hansson, P., "Playing with the Bits - User-configuration of Ubiquitous Domestic Environments", Proceedings of the Fifth Annual Conference on Ubiquitous Computing, UbiComp2003, Seattle, Washington, USA, 12-15 October 2003
5. Brad Johanson, Armando Fox, Terry Winograd. The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms. IEEE Pervasive Computing Magazine 1(2), April-June 2002.
6. Daniel Salber, Anind K. Dey and Gregory D. Abowd. The Context Toolkit: Aiding the Development of Context-Enabled Applications. In the Proceedings of the 1999 Conference on Human Factors in Computing Systems (CHI '99), Pittsburgh, PA, May 15-20, 1999. pp. 434-441.
7. Ullmer, B., a nd Ishii, H., mediaBlocks: Tangible Interfaces for Online Media (video), in Extended Abstracts of Conference on Human Factors in Computing Systems (CHI '99), ACM Press, pp. 31-32.

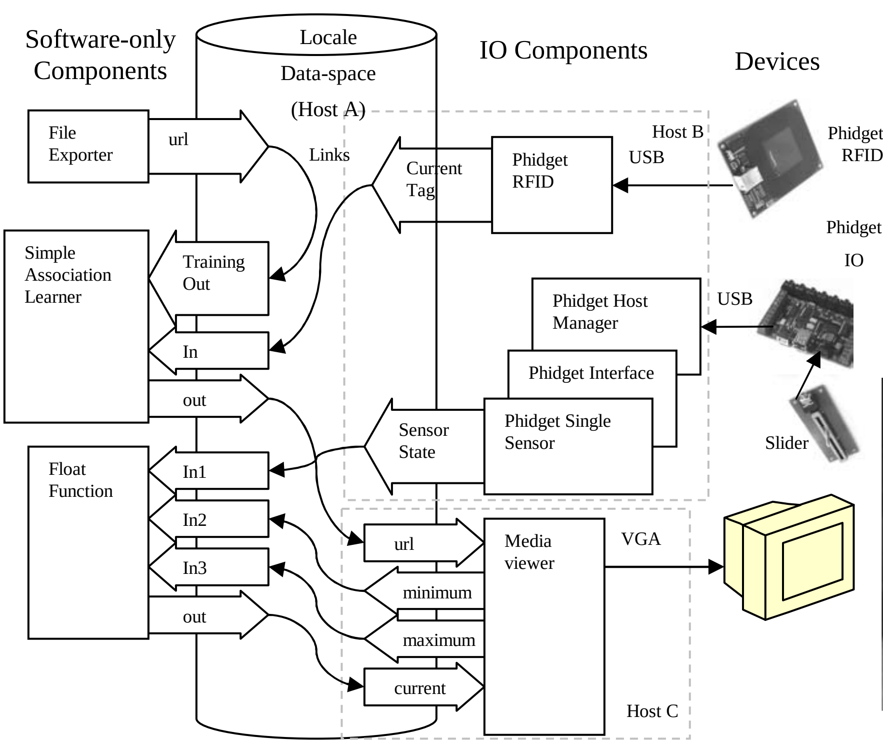

Figure 1. Sample tangible media application