Using CoolBase to Build Ubiquitous Computing Applications

Wesley Chan
Mobile Systems and Services Laboratory
HP Laboratories Palo Alto

HPL-2001-215
September 6th , 2001*

E-mail: wesley_chan@hp.com

ubiquitous computing, pervasive computing, CoolTown, CoolBase, mobile, embedded

The CoolBase platform from HP Labs facilitates the creation of a wide variety of ubiquitous computing scenarios–from delivering personalized museum exhibit services on handheld PDAs to "smart" printers that output documents from links stored on your mobile phone. The platform consists of several hardware and software modules that implement an architecture that links web presences to people, places, and things. Since the architecture relies on widely used open standards, the platform helps to overcome many of the challenges inherent in the design and creation of ubiquitous computing applications.

* Internal Accession Date Only Approved for External Publication?
To be presented at the UbiTools '01 Workshop, September 30, 2001, Atlanta, GA.

Using CoolBase to Build Ubiquitous Computing Applications

Wesley Chan (wesley_chan@hp.com)
Mobile Systems and Services Laboratory
Hewlett-Packard Company
Palo Alto, CA 94304

The CoolBase platform from HP Labs facilitates the creation of a wide variety of ubiquitous computing scenarios—from delivering personalized museum exhibit services on handheld PDAs [1] to "smart" printers that output documents from links stored on your mobile phone. The platform consists of several hardware and software modules that implement an architecture that links web presences to people, places, and things. Since the architecture relies on widely used open standards, the platform helps to overcome many of the challenges inherent in the design and creation of ubiquitous computing applications.

Ubiquitous Computing Scenarios

The "virtual guide" scenario is one familiar ubiquitous computing application. The virtual guide aims to deliver personalized services to a user based on some context (for example, location or time). A museum visitor, for example, may want to map out which exhibits she wants to see on her wirelessly networked mobile phone or PDA. As she visits each exhibit, her mobile device automatically displays the web page that is tied to each exhibit. A person organizing a meeting, on the other hand, may need to book a last minute conference room on her virtual guide. The guide senses what building the organizer is in, and automatically displays a listing of available meeting rooms at the specified meeting time.

Another familiar scenario involves "smart" appliances—devices that are able to connect and communicate with other (and often different) devices, in order to provide some useful set of services. A PDA, for example, discovers a nearby Internet radio and automatically downloads the interface to it, so that you can control it remotely. A "smart" printer automatically outputs a document you queued online after identifying yourself by swiping your employee badge on an attached RFID reader. A computer projector automatically displays your sales presentation after your mobile phone beams a web address into the projector.

Challenges

Two widespread challenges in building ubiquitous applications are *connectivity* and *linkage*. Connectivity problems arise due to the rampant diversity of devices and networks [2]. This leads to complicated and proprietary interfaces between two unlike devices. Most mobile phones today cannot send documents to your printer because the phone cannot understand your printer's interface. Likewise, your PDA cannot control your DVD player—again, because your PDA does not understand your DVD player's interface, and vice versa. Often, the only way to address the connectivity challenge involves costly point solutions or custom programs that enable inter-device communications.

Designers of ubiquitous computing applications must also create an architecture to *link* virtual services with physical entities. This is especially apparent in the virtual guide scenario. A museum guide has to retrieve the data linked to a particular exhibit—an audio file associated with a painting or an interactive game tied to an exhibit on tropical fish. A supermarket shopping guide needs to display recipes and coupons linked to food items that a shopper scans into the system.

Often, linkage architectures are designed specifically for each point solution. A museum guide, for example, may store all information locally on the client—so when a user approaches an exhibit, an infrared beacon triggers the playback of content on the device. On the other hand, a supermarket shopping guide may choose to store all of the services online in a proprietary database—so when a user scans in a product, custom software on the client queries the database to display information to the shopper. Each custom architecture requires costly integration work that often prevents its reuse in other ubiquitous computing applications.

The CoolBase Platform

The CoolBase platform addresses some of the challenges associated with building ubiquitous computing applications. At the heart of the platform is the concept that everything has a *web presence*. A web presence is a collection of services and information that can be accessed through the web. It can be as simple as the home page for a user, or as complex as a directory of applications, services, and relationships encapsulated in an XML document.

CoolBase implements an architecture that links web presences with people, places, and things, as described in [3] and [4]. The architecture is based on widely used web standards—HTML and XML over HTTP. This ensures that the CoolBase is open, extensible, and can interface with other web-based systems and services.

The CoolBase Platform consists of several software and hardware modules that facilitates the creation of a variety of ubiquitous computing applications. The platform provides the tools needed to build *ecosystems* that enable users to interact with a wide variety of services on an ever larger selection of "smart" devices. Some of the components of the CoolBase platform include:

Esquirt – This module provides an application programming interface that encapsulates a device interaction model that enable smart devices to access the web presences of other devices and physical entities by-reference. For example, Esquirt enables a mobile phone to send (or "squirt") the reference (e.g. a URL) of a document to a nearby printer. When the printer receives the reference, it retrieves the document and prints it for the user. Likewise, Esquirt also enables the mobile phone to act as a "universal remote control" for a wide variety of smart devices. When the phone connects to an Internet radio, for example, it retrieves a reference and subsequently displays the device's web presence on the screen. A person can operate the radio using the reflected user interface displayed on the phone.

Web Presence Manager (WPM) – This module facilitates the creation and hosting of web presences for people, places, and things. Additionally, the WPM manages the contextual relationships between various web presences. For example, a conference room, a computer projector, and a printer may each have their own web presences. But if both devices are located in the conference room, then the WPM can generate a relationship where the web presence of the conference room encapsulates the web presences of both devices.

CoolTown Beacons and Taggys – Beacons are small, battery-powered hardware devices that broadcast references (e.g. URLs) in specific locations. Beacons are used at HP Labs to broadcast a conference room's web presence to nearby mobile devices [3]. Taggys are small personal devices (about the size of a key fob) that can store references acquired from one or more beacons or Taggys. Additionally, Taggys can broadcast (or "squirt") references when a user activates this mode. Both the beacons and taggys are currently implemented using Infrared communications, but could be redesigned to support wireless radio networking as well.

CoolBase Appliance Server – This module is a simple web application server designed specifically for use in embedded systems—like printers and computer projectors. The module consists of an embedded web server that can host web services and serve dynamically generated web pages. The appliance server is especially useful in creating and hosting web presences directly on embedded devices.

Building Applications with CoolBase

At HP Labs, we have used the CoolBase platform to simplify the creation of both virtual guides and smart appliances. We used several CoolBase components to prototype a science museum tour guide [1]. Visitors borrowed mobile devices (HP Jornada PDAs) that were pre-loaded with an Esquirt web browser. We used Web Presence Manager to create and host the web presences from each of the different science exhibits. We attached beacons to different exhibits which broadcasted the URL of the web presence to mobile devices within range of the beacons. Consequently when users moved within range of a beacon, their PDA would receive the URL of the exhibit's web presence. The PDA would then retrieve the web presence and display a dynamically generated web page.

We have also used CoolBase to build an assortment of "smart" appliances. We used the CoolBase Appliance Server module to build the CoolTown Internet radio [5]. The radio is essentially a web application, and can be controlled from any web-enabled device. You simply need to query the URL of the radio to access its features—from playing music to creating a play list of your favorite music files. Moreover, any Esquirt enabled device can connect to the radio's infrared port. The radio broadcasts its URL to any device within its range, and "squirts" the URL via an infrared connection. When a Palm PDA, for example, with the Esquirt client installed receives the URL, it queries it, and receives the user interface for the radio as an HTML file. The client can then render the web page, allowing the user to "remotely" control the radio from their PDA.

CoolBase thus reduces several difficulties encountered when building ubiquitous computing applications. The platform enabled us to rapidly prototype various applications, including a science museum guide and an Internet radio. In creating these applications, we did not have to define custom or proprietary interfaces, nor did we have to spend time creating architecture for linking virtual services—web presences—to physical entities. CoolBase thus simplifies the development process for ubiquitous computing applications, enabling developers to concentrate their effort on creating services and content—instead of creating and designing different interfaces and architectures for each point solution.

The CoolBase platform is available for download at http://www.cooltown.com/dev/.

Acknowledgements

This paper describes work with contributions from the entire CoolTown project team at Hewlett-Packard Laboratories (http://www.cooltown.com/dev).

References

[1] Spasojevic, M and Kindberg, T. "A Study of an Augmented Museum Visit." Hewlett Packard Laboratories Technical Report. July 19, 2001.
http://www.hpl.hp.com/techreports/2001/HPL-2001-178.html

[2] "CoolTown Beliefs." Hewlett-Packard Laboratories.
http://www.cooltown.com/dev/beliefs.asp

[3] Kindberg, T, et. Al. "People, Places, and Things: Web Presence for the Real World."

[4] Debaty, P. and Caswell, D. "Uniform Web Presence Architecture for People, Places, and Things." Hewlett-Packard Laboratories Technical Report. June, 2000.
http://www.hpl.hp.com/techreports/2000/HPL-2000-67.html.

[5] Krishnan, V. "Baseboard, a CoolTown Reference Platform."
http://www.cooltown.com/dev/reference/coolbase/baseboard/RefPlatform.asp