

People, Places, Things: Web Presence for the Real World

Tim Kindberg, John Barton, Jeff Morgan, Gene Becker, Debbie Caswell, Philippe Debaty, Gita Gopal, Marcos Frid, Venky Krishnan, Howard Morris, John Schettino, Bill Serra, Mirjana Spasojevic
Internet and Mobile Systems Laboratory
Hewlett-Packard Laboratories
1501 Page Mill Road, Palo Alto, CA 94304, USA.
timothy@hpl.hp.com

Abstract

The convergence of Web technology, wireless networks, and portable client devices provides new design opportunities for computer/communications systems. In the HP Labs' "Cooltown" project we have been exploring these opportunities through an infrastructure to support "web presence" for people, places and things. We put web servers into things like printers and put information into web servers about things like artwork; we group physically related things into places embodied in web servers. Using URLs for addressing, physical URL beaconing and sensing of URLs for discovery, and localized web servers for directories, we can create a location-aware but ubiquitous system to support nomadic users. On top of this infrastructure we can leverage Internet connectivity to support communications services. Web presence bridges the World Wide Web and the physical world we inhabit, providing a model for supporting nomadic users without a central control point.

1. Introduction

People are nomadic. They move around to work, to shop, or to play. Increasingly the places they enter are computerized. Their workplaces are filled with computers; the shops they browse are computerized; the toys they buy are computerized. And communications networks reach into every corner: web access continues to soar; cell phones abound, new wireless technologies are emerging. The convergence of these increasingly pervasive computers and communications networks offers new opportunities and challenges for systems designers [27].

At HP Labs we have been exploring this opportunity and taking on these challenges through adaptations of web infrastructure to support nomadic users [7]. We have been pushing web technology in to digital "appliances" or

"things" like printers, radios, and automobiles [10]. We have been organizing physically related appliances into web "places" [5,17]. We have been exploring ways for people to use new digital communications devices to interact with these places and use the things they find there. Our early results, which we outline here, have been very promising. We believe that the web model of decentralized but standardized interaction will provide the basis for computing networks to support nomadic users.

Most of our work has focused on extending web technology, wireless networks, and portable devices to create a virtual bridge between mobile users and physical entities and electronic services. We think the physical world and the virtual world would both be richer if they were more closely linked. Currently the Web is largely a *virtual* space: a space of web "sites", online "malls", and chat "rooms". These virtual locations have little correspondence with physical spaces. While much of the information on the Web describes the world we physically inhabit, there are few systematic linkages to real world entities. This is unfortunate, because most of our activities concern physical objects other than computers.

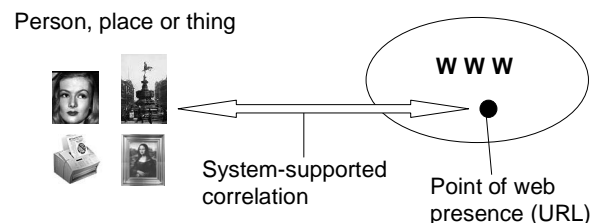


Figure 1. Points of web presence

In this paper we describe how we have systematically integrated web services to enhance communications with mobile people, to provide location-specific services in the places that they visit, and to provide interaction with the things that they encounter. To help users and developers formulate a common model of how these systems behave

we describe them as supporting "web presence". Web presence extends the "home page" concept to include all physical entities and to include automatic system supported correlation of the home page or *point of web presence* with the physical entity (see Figure 1).

We divide up physical entities into three categories: people, places, and things. This is the set of categories that the designers of Taligent followed, although to different ends [32]. As we go about our daily lives we move between *places* (the home, the office, shopping malls etc.), we meet with and talk to *people*, and we find and use *things*. People are, of course, the users of things and the occupants or visitors in places. Places have a special role as the venue or container for people and things. Our goal is to expand our access to people, places, and things by bridging them to the virtual world of web content. We want to make people, places, and things *web-present*.

Things become web-present by embedding web-servers in them or by hosting their web-presence within a web server. Places become web present by organizing web things into collections under the management of a web service we call a "PlaceManager". People become web present by offering global web home pages with "WebLink" services to facilitate communications between individuals and by offering information via location-specific PlaceManagers. We describe demonstration applications that show the benefits of web presence, and we describe the infrastructure that supports them.

Overall we aim to show that the Web is the most promising basis for a nomadic computing infrastructure. We shall argue this based on the following:

Ubiquitous access. The Web derives its power from a design that has led to widespread accessibility. The rising quality and reach of the Web increases the probability of Web access in mobile venues. People use the Web at work and at home today; increasingly they will find it available when they shop or travel; eventually they will find adequate access "everywhere" they want electronic connections. The Web offers accessibility that supports mobility in two senses. First, resources on the Web can be accessed from any device that supports the standard HTTP protocol [18]. This includes devices that the user carries: general-purpose devices such as personal digital assistants (PDAs) and laptops, and information "appliances" dedicated to a specific function, such as digital cameras. Equally, it is a simple matter to put HTTP *server* support in devices [10] that nomadic users encounter, such as printers and projectors. Second, the Web supports mobile users by allowing them transparent access to resources outside their current environment. That is, the Web runs over the Internet, the Internet is widely accessible, and the Web runs the same way at all points of the Internet. We shall discuss ways of bridging devices that are *not* IP-capable to the Web.

Just enough middleware. The diversity of the devices that mobile users carry and encounter argues against specific software based on device type or even application type. It also argues against assuming that all devices will run a form of middleware, such as Java or CORBA, that is heavy on the commitments it imposes at the application, language, or resource level. Rather we believe that the Web standards of HTTP and URLs [3] have proven themselves to support adaptation to diversity. The form of interaction with particular devices and other entities should be encoded using XML [41] documents and MIME types, not binary formats and language-specific interface signatures. Our applications include a variety of digital devices that we can integrate with minimal additional, specialized software.

Local means local. We will demonstrate how to deliver Web services to mobile users without requiring a global wireless connection like a cell-phone or mobile IP [30]. This has the advantage of minimizing how much of the infrastructure needs to be up and running in order for users to interact with local services. For example, a mobile user should be able to print a document at a nearby printer without necessarily having to contact any global services. A local Web server should be enough, and sometimes even a simple peer-to-peer interaction will suffice. We shall also show how users can be in charge of where, when and to whom they make their presence known. This is a critical factor in the extent to which users will accept systems that provide location-aware services.

Section 2 defines "Web presence" and illustrates the usage scenarios that it supports by describing "CoolTown" [7] demonstration applications that utilize Web presence for people, places and things. In Section 3 we sketch the requirements for a system that supports Web presence and describe our infrastructure that meets these requirements. Section 4 relates our work to that of other efforts.

2. Web Presence By Example

Since our approach to services for mobile users differs from others we begin with some examples. These examples are based on the demonstrations running in our "Cooltown" demonstration center at HP Labs [7]. After describing the examples we will explain the infrastructure that supports them.

2.1. Visiting the Cooltown Museum

The Cooltown Museum and Bookstore offers visitors a Web-enhanced experience. As visitors tour the museum, their portable digital assistant (PDA) can receive Web URLs from wireless "beacons". These beacons are small infrared transceivers located close to pictures or

sculptures; the URLs link into a Web of information about the items. Using the PDA's Web browser, visitors can read or hear about the artist or the work and about related art works in the museum. The URLs can also be stored as bookmarks for further study or they can be used to select reproductions of the artwork from the museum's online store. The museum staff uses the same URLs for inventory control as the URLs point to the object's point of Web presence.

As visitor move into the Cooltown Museum bookstore they can use their PDAs to sense the URLs associate with books, calendars, and posters for sale (Figure 2). These URLs will give them book reviews, available inventory of calendars, and other colors of posters for examples. The museum bookstore's Web portal provides services to assist in buying books, including a service to order books that are not available. In addition, the bookstore offers a printing service, for visitors to print out Web pages that they collected during their visit or even order reproductions of paintings to be printed just in time for purchase at the exit.

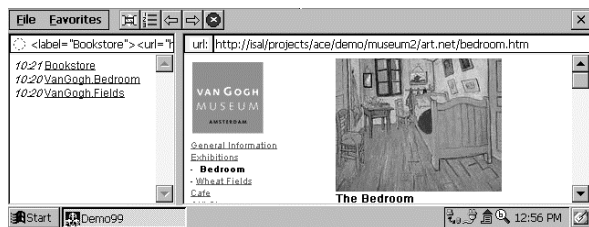


Figure 2. Museum visitor's PDA screen.

These scenarios combine local wireless communications for location-specific addressing with wider range wireless Internet access to obtain the Web pages addressed. Thus these PDAs have 802.11-like connectivity [38], but no location-sensing or tracking technologies are needed to create a location-aware application.

2.2. Working in a Cooltown Conference Room

Every Cooltown conference room supports mobile users. As you enter a Cooltown conference room you can collect the URL for the room from a beacon or tag into your PDA or cell phone. The URL will lead to a Web page for the room giving links to, for example, the room's projector, printer, and electronic whiteboard, as well links to Web-based maps. The Web page is served from a "PlaceManager", a web portal service dedicated to the conference room. This portal is a physically-based local equivalent to web portals like *yahoo.com* and *excite.com*. This meta-service acts as a container for the services of the room and as a portal or gateway to web access for visitors. Its contents can be tailored to the conference room's attendees or the current activities.

Workers in the room can also "eSquirt" URLs at the room's "web appliances" such as web-enabled projectors or printers. An eSquirt is a wireless transfer of a URL over a short-range wireless link, like IR [23] or Bluetooth [4]. Squirting a URL at the projector will project the corresponding web page, creating a shared web browser. Squirting a URL at the printer will fetch the web pages and print them. A worker armed with only a cell phone or even a wristwatch can use the conference room facilities by squirting URLs referencing content available on the Web.

To support visiting guests of an organization the PlaceManager acts as a proxy and firewall. This allows visitors access to room facilities without compromising the surrounding facility. Visitors inside the conference room—and therefore inside the host's firewall—also need to traverse that firewall to access web content and content within their company's Intranet. For this users may employ a secure "Web tunnel" that encrypts traffic and authenticates users all over HTTP links.

Finally a visiting worker may need to pull in colleagues from remote sites to join the discussions in Cooltown. "Weblink" supports multi-modal Web-facilitated communications in a way that is completely controlled by its users. The visiting workers will register themselves with their Weblink homepages; this registration can optionally include the URL for the publicly accessible Web services of the conference room. Colleagues at their desks remote from the conference room can then, for example, post files to the room's printer without the usual file-transfer, print driver difficulties we face in current systems.

This conference room scenario extends the URL addressing and discovery illustrated by the museum scenario to include directories (PlaceManager) and controllable services (projector). Note that the assumptions made by the infrastructure are quite low: these mechanisms are all straightforward extensions of Web technologies and they are all based on resident services with standard clients. No centralized global service needs to be deployed for these nomadic users.

2.3. The common user experience in Cooltown

The three examples above explore various aspects of web presence but they also show a common theme. The typical user experience we seek with web presence is that of collecting links to points of web presence as they encounter them in the physical world. One could even think of the physical world as a web page, with links at certain physical points. Of course the links are URLs presented on the user's client device and the result of clicking on such a link will be a real web page, delivered to the user's screen. The user can then travel through the virtual space of the web page with normal Web

techniques or the user can travel through the physical space to find a different point of web presence. The user understands URLs as pointers to meta-information about objects at the same level that users of Web browsers now understand URLs as bookmarks for Web content.

3. A web presence infrastructure

We have presented a vision of web presence for the benefit of users – particularly nomadic (mobile) users – and the organizations that host them. Realizing web presence requires an infrastructure with some challenging properties. This section begins by describing those requirements and then describes the infrastructure itself.

3.1. General infrastructure requirements

Services everywhere. The infrastructure should give users and devices convenient and widespread access to an open set of services supplied by, or based around, people, places and things.

Scalability. The number of people, places and things in the world suggests support for trillions of web present entities. A system to provide web presence has to perform well and remain manageable in the face of such a potentially large scale.

Simple model of configuration by users. The overheads for users of setting up a new web-present place, or giving a person or thing web presence must be low. A "place master" in charge of configuring a PlaceManager, for example, should not need to be more sophisticated than a web master.

Layered infrastructure. The infrastructure should be layered so that we minimize the dependencies on the infrastructure of individual devices that do not require higher-level services. For example, web presence in the home or the car should be possible without requiring the sophisticated resources provided by a large organization.

In the rest of this section we outline an infrastructure that meets these requirements and we describe the implementation work we have done to demonstrate the potential of such an infrastructure.

3.2. Infrastructure layers for web presence

To meet the requirements described in the preceding section we argue specifically for "web presence". The alternative would be to provide, for example, "Java presence" (see Jini [24]), or "CORBA presence" [31]. The advantages of the Web over other systems include its much wider deployment and the fact that it comes with a user-interface, the web browser, with which users are familiar. The Web benefits from a simple system model - standard-format content exchange via HTTP and URLs. Web solutions exist for higher-level content-provision

services, including security and payment. Web standards are language-independent, and have been implemented on many OS and hardware platforms. The user can employ a browser – on a PC or on a PDA, wherever the user happens to be – to exchange content with people, places and things.

By using web technology we increase our chances of having a ubiquitous, scalable system and Web systems have proven to be simple enough for many users to configure. The central task of a Web presence infrastructure then is to add new layers to support presence while preserving the desirable properties of the Web.

Figure 3 shows layers of a web-presence infrastructure for which we give an overview in the following subsections. At the bottom layer are mechanisms for obtaining the points of web presence of people, places and things, through discovery systems and by sensing URLs and identifiers. In the middle layer are services for exchanging content with web-present entities and coordinating the processing of that content to provide application-level functionality – all through the HTTP standard. At the top layer is infrastructure to provide services related to places and nomadic users.

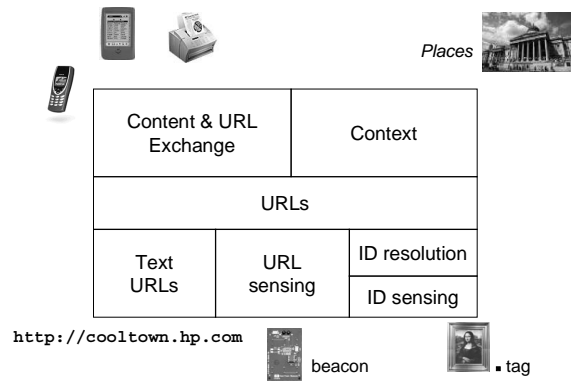


Figure 3. Infrastructure layers

In the remaining discussion we will personalize the descriptions by replacing "a visitor" with "Veronica" and, when we discuss communications we'll call Veronica's host "Harry". Veronica travels with a portable web browser, visits places that may or may not know her, and she is able to use things she finds there when appropriate without pre-arranged or pre-paid admission.

3.3. Finding Points of Web Presence

The lowest layer of our infrastructure contains the means for addressing points of web presence – that is, obtaining their URLs. One way a user can find a person, place, or thing is by keying in a URL they read on the device or on the wall of a place. However to make web presence easy to use our infrastructure must also support

sensing and automatic discovery of web-present entities as described in section 2.

Figure 2 shows a screenshot, from our demonstrator implementation of the Cooltown Museum described in section 2.1, of the PDA Veronica carries with her as she tours an art gallery. The PDA is equipped with sensors for tags and beacons, but otherwise is equipped with standard software, including WaveLAN connectivity. On the left of the screenshot are links to three web-present entities that Veronica has visited since she turned on the device's sensor. The first is a place, the bookstore in the gallery. The next two (less recently visited) are paintings. The device has remembered the links to these web-present entities (and will do so as long as Veronica desires). As she visits places and senses things, the PDA records all such links by default, creating a series of bookmarks corresponding to Veronica interests.

Discovery. When Veronica enters an area of the Cooltown Museum with a wireless network compatible with her PDA or when she attaches her computer to a public access point on a wired network, her device can send network packets to a "service discovery" multicast address [2,8,15,16,35]. This well-known, preconfigured address routes the packets to all network services that have joined the group. If Veronica requests printing services, printers in the multicast group can respond. Special servers that have grouped printer services can also respond. In both cases, the response is a URL pointing to the printer's web presence. In our example, the gallery bookstore's printer, which is publicly available, will respond. We discuss discovery more in Section 3.2.3, on infrastructure for web places.

This multicast network based discovery works well in some scenarios but it suffers from two problems based on its reliance on network topology. First the range of the discovery may include too few resources, preventing users from locating electronic services they need and that are near by, but that don't happen to be within range of the multicast packets. While newer discovery schemes seek to avoid this problem (see [8]), they can't address the second problem: the range of discovery may include too many resources. For these reasons we have been investigating "physical discovery" mechanisms.

Direct sensing. In our "art gallery" implementation, we laid out a room with pictures on the walls and situated the "bookstore" nearby. We implemented web presence for these entities using active, URL-emitting devices (beacons). We call this form of discovery *direct sensing*. Next to each painting and in the bookstore we placed infra-red beacons that supply PDAs with the URL of the corresponding point of web presence. The paintings' URLs are those of pages supplied by the Van Gogh museum. The URL of the bookstore resolves to a "place

manager" server (discussed in Section 3.2.3), which provides the bookstore's web portal and which runs service discovery and registration services.

Indirect sensing. In the case of indirect sensing, the user's client device obtains a value that it has to look up to obtain a URL. There are many types of value that can be 'sensed' as the user travels through the physical world. For example, the device could obtain latitude and longitude coordinates from a GPS sensor, which could then be translated into a zip code and hence into the URL of a place's web portal provided by Yahoo, for example. The device could even perform image recognition against a database of tourist sites to provide the URL of a corresponding web page.

We have been experimenting with indirect sensing where the sensed value is obtained from a *tag*. There are electronic tags like RFID tags [33] and iButtons [22] – small devices that supply a unique identification string to a sensor placed near them or in contact with them. Barcodes can be used instead of an electronic tag.

Figure 4 shows a different art gallery demonstrator in which a PDA client senses a painting's tag. A correlated web page is resolved from the tag identifier and the client presents it to the user.

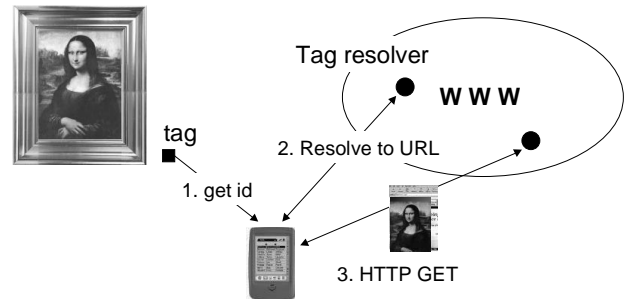


Figure 4. Indirect sensing: the web presence for a painting.

We have produced a prototype implementation for sensing tagged things and displaying a corresponding context-dependent web page. The sensing device knows the URL of a *resolver*. When the client senses a tag, it sends the identifier to the resolver. If the resolver has an entry for the identifier, it sends back the corresponding URL. The user may choose a resolver that reflects his or her personal requirements, or those of a group of users with interests or tasks in common. For example, visitors to an art gallery could be offered resolvers that provide the URLs of pages in their native language, or pages provided by the art school at which they study.

We have begun to address two issues with this scenario. The first is that of how to resolve tags for which no resolver is known *a priori*. We are experimenting with resolvers that provide a location-dependent context, but

which can be linked in a hierarchy so as to provide pages from wider contexts, in addition. The second issue is that of how to enable users to create a new point of web presence for an entity. We are experimenting with template managers, which provide the user with an input form for the entity, formatted according to a standard template that is shared amongst the members of their user community.

3.4. Infrastructure for things; ChaiServer and eSquirt

Given a URL from one of the lower-layer services, we know a point of web presence for a person, place or thing. Users and devices that have discovered a web-present thing utilize it by a combination of retrieving content from it and sending content to it.

Web devices provide services to users based on their electronically accessible functionality. For example, Veronica sends a document to a web-present printer; Veronica's home management service switches on the lights in her home when it gets dark, by accessing the web-present lighting controller. These operations take the form of the exchange of content with points of web presence. Much of the content exchanged with things is everyday content that we use widely today (images, PostScript files, etc.); but some of this content takes the form of formatted data that describes the state of things.

These interactions require the following HTTP infrastructure:

Client software. This includes standard browsers on users' client devices – PCs and PDAs. Client devices that access web-present entities under programmatic control require a runtime that provides standard HTTP-client *POST* and *GET* operations.

Web servers. Small, embedded web servers are needed to run on devices that have sufficient capabilities to support them, such as a printer; associated HTTP operation handlers are required to read and manipulate the devices' physical state. Gateway web servers running on conventional server machines are needed for devices that cannot run a web server, such as a simple domestic lighting unit. These utilize HTTP operation handlers that read and manipulate the physical state of remote things.

Figure 5 shows interactions between two clients and their respective web devices. It shows a digital camera that interacts with a printer via HTTP operations on the printer's point of web presence. The URL for the printer resolves to its network address. The camera first interrogates the printer to find out the formats it accepts. The printer can supply this as an HTML page or as an XML document that describes it. Then the camera sends the image as, for example, JPEG encoded data. Figure 5 also shows lights connected to a home PC by a controller unit. The home PC acts as a gateway to the lighting unit.

A remote user issues HTTP operations via a web browser on a PDA; the programs on the PC that handle HTTP operations directed to the light's URL manipulate the lighting unit accordingly.

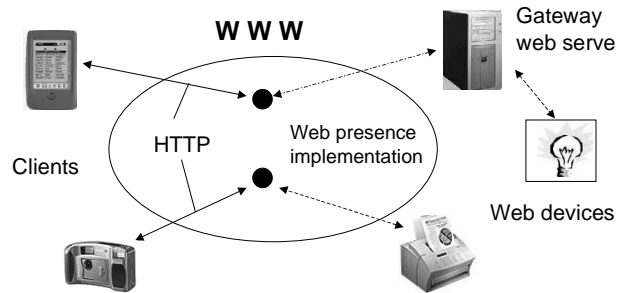


Figure 5. Devices (printer, light), with and without embedded web servers.

To explore the potential for web-present devices, we developed the ChaiServer web server [6]. It has been engineered to have a small footprint (200 kbyte). A ChaiServer executes objects called *chaillets*, which handle HTTP requests. Chaillets can be dynamically loaded and unloaded to suit different environments. A Chailet implements a service with a service type name. We assume that service type names are "well known", that is that clients will know the strings that correspond to the service they need. Standards will emerge including names such as "printing", "logging", "camera", due to efforts such as the Universal Plug and Play Forum [36]. A service also has a list of named *entry points*, which correspond to chaillet method names.

As an example, consider a GET request on the URL: `http://55.55.55.55/laser15.print?statusCheck=text/html`. This arrives at the device with IP number 55.55.55.55. ChaiServer dispatches it to the chaillet instance named *laser15* which implements the service name *print*. The chaillet method being invoked is *statusCheck*, and the client has requested it to return the response as HTML.

In addition to Web server functionality, a ChaiServer can be configured to support device discovery. This allows Veronica to "walk-up and print" at a public printer in a bookstore. She should be able to print a document from her portable device without this having to involve any nearby servers. This can be done as simply as having the client device use HTTP POST to deliver the document to the printer using a URL. Veronica's portable device obtains this URL by sending requests for printer service over a multicast IP or IrDA [23] link to the printer. The printer's device discovery protocol responds with a URL. The client device can then POST the document to that URL.

The *eSquirt* technique described in section 2.2 provides a different approach to printing. Suppose

Veronica is in the art gallery bookstore and decides to print a reminder of the Van Gogh she just saw. She may not like the result she can get when she prints the image from her PDA screen. The PDA, with its limited resources, has been supplied with a reduced fidelity version of the page, in which the pictures have relatively low resolution. With eSquirt, Veronica can walk up to the printer and "squirt" a *URL* for the image she has selected. The PDA sends the URL of the page to the printer, and not the page content that it has stored. On receipt of the URL, the printing service obtains a full-fidelity version of the web page, renders it and sends the result to the printer.

3.5. Infrastructure for places; PlaceManager

We said that a place is a context for service provision, based on an underlying physical domain permeated by one or more networks. For nomadic users, wireless networks are much more convenient. Examples are a railway station covered by WaveLAN connectivity [38], a café covered by Bluetooth [4], and a home covered by HomeRF [19].

To deliver the convenience potential available with wireless networks, it should be possible to connect devices to networks spontaneously, and have them function with little or no human intervention. This implies more than just automatic IP address allocation. It implies automatic service discovery and registration: the ability for devices to discover appropriate services, and the ability for devices (and other service providers) to register their availability without human intervention. In section 3.2.1 we discussed ways clients can discover local services. The reciprocal problem is the configuration of a collection of services to create a "place". Several discovery and registration services have been designed [2, 8, 15, 16, 35]. These services work well only when the area of service provisioning matches the area covered by a sub-network.

Real places don't always match networks topology: they can physically overlap, and they may have fuzzy boundaries. For example, we want to be sure that we access the printer in our hotel room, and not the one next door which is connected on the same 802.11 network. But what is to be included in a place is also a matter of policy. Policies determine which of the services that exist in a place's physical domain should be made available to users within the place. Some policies concern the place as a whole. In particular, users are offered services that are relevant to the place's purpose. For example, the same hall may be a disco tonight and a religious venue tomorrow. Other policies affect individual users. In particular, users and devices acting on their behalf should not be offered services for which they do not have access privileges. For example, compared to an employee, a

guest accessing a place's web portal may see a restricted set of web pages and see less information on those web pages.

The need for physical and contextual organization of place information creates extra burden on configuration of web presence for places. We have begun to explore some of these issues with an implementation of a prototype *PlaceManager* layered on top of ChaiServer. The PlaceManager [5] is a component that underlies the web portal. It is responsible for providing secure views of the set of resources that are present in the place and the services based around them. In general, the content that the place manager provides to a particular user depends on the client's security principal, the user; on that principal's preferences; and on the client device's functional capabilities.

3.6. Infrastructure for nomadicity

So far we have described services that are relevant to particular places. But users require higher-level services that meet their needs to be mobile and to work in different places at different times. These services include:

Services for remote access to things: for example, to fetch documents from other places, for printing in the current place.

Services for remote access to people: for example, for Veronica to reach Harry wherever he might be and through whatever means suited the two of them.

Location-aware services based on physical coordinates such as latitude/longitude. For example, while Harry waits for Veronica at the bus stop he should be able to get details about her bus's current position.

There have been many efforts on each of these fronts. Access to "enterprise data" and global email services are examples of the first two kinds of service that have been widely deployed. At HP we have explored examples of these services based on web infrastructure. For secure remote access we developed *SecureWebTunnel*, and for access to people, *WebLink*. For location-aware services we have explored several web-present places, including the art gallery and its bookstore described above. Another example is the *WebBus*. The WebBus example, like the art gallery and bookstore, is an exploration of how to build context-dependent services for a particular scenario.

Secure web tunnel. HP's SecureWebTunnel work addresses the following example. Suppose Veronica is working in Harry's conference room and she needs to fetch a business plan from her home office. Unfortunately, she is of course outside the firewall of her company's intranet. SecureWebTunnel enables Veronica to fetch documents via the Web from her company intranet securely, even though she is outside the firewall. Moreover, apart from having to prove her identity, the

fact that she is outside the firewall is transparent. She can continue to use the standard browser on her PDA without reconfiguring it when she travels outside the organization. And she continues to use the same URLs outside the organization, as she would use inside it.

SecureWebTunnel provides a secure communication channel through a pair of web proxies, one in Veronica's PDA, and one in the firewall of Veronica's company intranet. The proxy in her PDA is location-aware. When the PDA's proxy is given a URL for a location that is inside the intranet, it behaves differently according to whether it is currently inside or outside the intranet's firewall. In the latter case, it contacts the SecureWebTunnel proxy in the firewall rather than the proxy normally used inside the intranet. The PDA proxy makes a secure SSL connection [13] to the firewall proxy. Veronica is challenged, and has to type in a password (which is transmitted securely over the SSL connection). The SecureWebTunnel proxy in the firewall verifies the password. For a limited time, Veronica can now issue requests to any web resource that has been cleared for secure export in this way. The secrecy and integrity of the data is guaranteed by the SSL connection

The SecureWebTunnel work is similar to that of Abadi et al. [1]. SecureWebTunnel has a significant advantage: the use of a proxy on the client device means that the firewall proxy does not have to re-write web pages on the fly in order to provide URL transparency outside the firewall. The Satchel project also addresses document access from outside firewalls [14]. An outstanding research problem is how to implement policies and manage more selective access control for this type of export of services from firewall-protected places. Aspects of this problem have been tackled for virtual workspaces for collaborative activities [25].

WebLink. HP's WebLink uses a person's point of web presence as a level of indirection for electronic communications. A user such as Harry, with whom Veronica needs to communicate, places a link to a WebLink redirector service in his conventional personal web page. The WebLink redirector service runs on a globally accessible web server. When Veronica clicks on the link in Harry's page, an invocation is made to the WebLink redirector service to request a resource for communicating with Harry. If the web redirection service possesses a URL for communicating with Harry in the place where he currently resides, it returns that URL to Veronica's PDA. However, if Harry's location cannot be determined (i.e. he is offline) then the redirector returns a web page that includes a service to leave a message for Harry. This information will be delivered to Harry when he comes on-line.

The web redirection service is updated with the URLs of suitable communication services as Harry moves

around, as long as he wishes to be reachable. There are various ways of implementing the update mechanism, reflecting different choices of responsibility between the user's PDA and web servers in the places that host the user. We have implemented an update service that runs in each Web-present place. When Harry enters a place, he identifies himself to the PlaceManager through his PDA; if he wishes to be contacted there, his PDA also supplies the URL of his redirection service. The place-level service identifies the URL of a suitable communication service for Harry in the place. It registers this choice of communication channel with Harry's redirection service, which will then offer it via the link in Harry's web page.

With WebLink, the communications channel that Harry and Veronica use can depend upon their locations and preferences without the usual trial-and-error used in electronic communications currently. There are other efforts similar to WebLink, including the mobile people architecture [28] and various commercial "universal inbox" systems. Weblink's distinguishing features include its integration with other aspects of web presence. Since the location of the user can be sensed electronically the appropriate communication channel can automatically be offered. Standard web security techniques can be used to control access to Harry's entry in the WebLink redirection service. Harry can password-protect his entry, so that only those he trusts can communicate with him through it.

WebBus. HP's WebBus is an embedded web server with a GPS transponder and a wireless Internet link. The server travels in the bus. Clients contacting the web server get different experiences depending upon their location. For Veronica in the bus, the WebBus provides a web portal view that depends upon the bus' current location. As the bus approaches Coit Tower, the portal begins to offer information about that attraction. For Harry at the bus stop waiting for Veronica, the WebBus server gives the bus's current location and expected arrival time. Note that the location dependence works twice here. First the bus stop's PlaceManager gives Harry's PDA the link to the appropriate WebBus. Then the WebBus provides up to date information based upon its current GPS reading. In this way Harry can easily decide if waiting for Veronica is prudent or whether he should browse the bus stop's web portal for a local bookstore.

4. Summary and discussion

We have described web presence as a basis for bridging the physical world with the World Wide Web. Through our examples, many of them based on demonstrators that we have implemented, we have illustrated the ways in which web presence for people,

places, and things can support users as they go about their everyday tasks.

The notion of web presence and our demonstrations are built upon earlier work in our laboratory. This includes work on Web-enabled devices [20], work on embedded web server technology [10], and Java implementations of web servers augmented with object models, event models and service discovery [6]. This exploration of the interaction between the Web and the physical world continues with the projects outlined here.

In presenting our work we have made two central arguments: that users can benefit from greater connection between the physical and the virtual worlds, and that the Web is the best technology for building that connection.

The first argument is based upon the observation that users' everyday activities mostly concern physical objects, but most of today's computers are not linked to physical objects. Relatively few physical objects have computers inside them or information systems correlated with them. Portable computers with wireless communications and sensors allow us to systematically increase the connection.

We are by no means alone in this view. The work at Xerox PARC [37], in particular, shows us how to use electronic tags and beacons to enhance physical things by associating their presence with virtual actions, such as the invocation of a program. Work at Georgia Tech [34] has expounded a view of context-enabled applications and an infrastructure to support them. The GUIDE project at Lancaster University [9] has developed a system for presenting guide information to tourists as they visit sites. The Satchel project at Xerox's European Research Centre [14] has produced tools to support the mobile worker with services to fetch remote documents and print them locally. A group at Stanford University [21] is producing a framework for invoking trigger actions according to the user's physical circumstances, for example to remind him to buy milk as he approaches the supermarket. Yanin and Ishii show how an action, such as removing or placing a container on a shelf, can be used to automatically update a web page about that container [40]

Our second argument, that the Web (rather than, say, JINI or CORBA) is the best middleware for building the connection, rests on the Web's wide deployment. It is widely deployed because of the robustness of its standards; it supports evolving standards to embrace new content; it can be implemented on devices with relatively little memory and other resources; and its user interface, the web page, enables users to perform a wide variety of tasks. It supports both human to computer and computer to computer interactions.

We expect that device manufacturers and others will define standards for content in specific domains, and some of these will be accepted and made widespread. This is the project of the Universal Plug and Play

Consortium [36]. New devices will be built to conform to prevailing standards; gateways will be built to encompass legacy devices. Other work on programming over the Web appears in WIDL [29] and WebL [26].

Many similar projects exist, whose aim is to address some aspect of building the connection between the virtual and the physical. Where we differ is in the comprehensiveness of our approach (people, places *and* things), and our commitment to the Web as the key component of the connection.

However, we made it clear that the Web provides only part of the infrastructure for web presence. We need sensing and service discovery technologies to feed the Web with URLs. The Web provides HTTP for content exchange with points of web presence, but the Web as it stands does not support nomadic working: that is where PlaceManager and the other technologies we have described make their contribution.

More information about our demonstration applications is available at our "CoolTown" web site, <http://www.cooltown.hp.com/>.

References

- [1] Abadi, M., Birrell, A., Stata, R., and Wobber, E. Secure Web tunnelling. http://www.research.digital.com/SRC/personal/Martin_Abadi/Papers/tunnel/206.html.
- [2] Arnold, K., Wollrath, A., O'Sullivan, B., Scheifler, R., and Waldo, J. The Jini Specification (The Jini Technology Series). Addison-Wesley Pub Co; ISBN: 0201616343.
- [3] Berners-Lee, T., Masinter, L., and McCahill, M. Uniform Resource Locators (URL). December 1994. Internet RFC 1738. Available at URL <ftp://ftp.nordu.net/rfc/rfc1738.txt>.
- [4] Bluetooth home page. <http://www.bluetooth.com/>.
- [5] Caswell, D. And Debaty, P. Creating a Web Representation for Places. In the proceedings of the 2nd International Symposium on Handheld and Ubiquitous Computing 2000 (HUC 2000) Lecture Notes in Computer Science 1927 pg 114.
- [6] ChaiServer home page. http://www.chai.hp.com/chai_server.html.
- [7] CoolTown home page. <http://www.cooltown.hp.com/>.
- [8] Czerwinski, S.E., Zhao, B.Y., Hodes, T.D., Joseph, A.D., and Katz, R.H. An Architecture for a Secure Service Discovery Service. Fifth Annual International Conference on Mobile Computing and Networks (MobiCom '99), Seattle, WA, August 1999, pp. 24-35.
- [9] Davies, N., Cheverst, K., Mitchell, K., and Friday, A. Caches in the Air: Disseminating Information in the Guide System. Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99), New Orleans, Louisiana, U.S., 25-26 February 1999.
- [10] Embedding Web Access Mechanism in an Appliance for User Interface Functions Including a Web Server and Web Browser. United States Patent no. 5956487, Sep 21, 1999. Inventors: Venkatraman, C., and Morgan, J. Assignee: Hewlett-Packard Company, Palo Alto, Calif
- [11] Excite home page. <http://www.excite.com/>.
- [12] Foley, J. MIT private communications.
- [13] Freier, A.O., Karlton, P., and Kocher, C. The SSL Protocol Version 3.0. <http://search.ietf.org/internet-drafts/draft-ietf->

tls-ssl-version3-00.txt.

- [14] Flynn, M., Pendlebury, D., Jones, C., Eldridge, M., and Lamming, M. The Satchel System Architecture: Mobile Access to Documents and Services. To appear in *Mobile Networks and Applications*, ACM.
- [15] Goland, Y.Y., Cai, T., Gu, Y., and Albright, S. Simple Service Discovery Protocol/1.0 Operating without an Arbiter. <http://search.ietf.org/internet-drafts/draft-cai-ssdp-v1-03.txt>.
- [16] Guttman, E. Service Location Protocol: Automatic Discovery of IP Network Services. *IEEE Internet Computing*, Jul.-Aug. 1999. pp. 71-80.
- [17] Harrison, S., and Dourish, P. (1996). Re-placing space: The roles of place and space in collaborative systems. In *Proceedings Computer Supported Cooperative Work '96*, Cambridge, MA. New York: ACM.
- [18] HTTP - Hypertext Transfer Protocol. <http://www.w3.org/Protocols/>.
- [19] HomeRF home page. <http://www.homerf.org/>.
- [20] HP device could make Web a device controller. In *Netscape World*, Dec. 1996. <http://www.net-dev.com/netscapeworld/nw-12-1996/nw-12-newsbriefs2.html#HP>.
- [21] Huang, A.C., Ling, B., Ponnekanti, S., and Fox, A. Pervasive Computing: What Is It Good For? In *proc. Workshop on Mobile Data Management*, in conjunction with ACM MobiCom 99.
- [22] iButton home page. <http://www.ibutton.com/>.
- [23] IrDA home page. <http://www.irda.org/>.
- [24] Jini home page. <http://www.jini.org/>.
- [25] Kindberg, T. Security for Network Places. In *proceedings Distributed Systems Security Workshop, ECOOP '98*, Belgium.
- [26] Kistler, T. and Marais, H. WebL, a programming language for the Web. *Computer Networks and ISDN Systems (proc. WWW7)*, April 1998. pp. 259-270. <http://gatekeeper.dec.com/pub/DEC/SRC/technical-notes/SRC-1997-029-html/>.
- [27] Kleinrock, L. (1997) *Nomadcity: anytime, anywhere in a disconnected world. Mobile networks and applications vol 1 no 4*, pp. 351-357.
- [28] Maniatis, P., Roussopoulos, M., Swierk, E., Lai, K., Appenzeller, G., Zhao, X., and Baker, M. The Mobile People Architecture. In *the ACM Mobile Computing and Communications Review*, July 1999.
- [29] Merrick, P., and Allen, C. Web Interface Definition Language (WIDL).
- [30] Mobile IP Web Resources, <http://computer.org/internet/v2n1/mobile.htm>
- [31] The Object Management Group home page. <http://www.omg.com/>.
- [32] Potel, M., and Cotter, S. *Inside Taligent Technology* (Addison-Wesley, 1995).
- [33] RFID home page. <http://www.aimglobal.org/technologies/rfid/>.
- [34] Salber, D., Dey, A.K., and Abowd, G.D. The Context Toolkit: Aiding the Development of Context-Enabled Applications. In *Proceedings of the 1999 Conference on Human Factors in Computing Systems (CHI '99)*, Pittsburgh, PA, May 15-20, 1999. pp. 434-441.
- [35] Salutation Consortium home page. <http://www.salutation.org/>.
- [36] Universal Plug and Play Forum. <http://www.upnp.org/>.
- [37] Want, R., Fishkin, K.P., Gujar, A., and Harrison, B.L. Bridging Physical and Virtual Worlds with Electronic Tags. In *Proceedings of the 1999 Conference on Human Factors in Computing Systems (CHI '99)*, Pittsburgh, PA, May 15-20, 1999.
- [38] WaveLAN home page. <http://www.wavelan.com/>.
- [39] Yahoo home page. <http://www.yahoo.com>.
- [40] Yarin, P., and Ishii, H. TouchCounters: Designing Interactive Electronic Labels for Physical Containers. In *Proceedings of the 1999 Conference on Human Factors in Computing Systems (CHI '99)*, Pittsburgh, PA, May 15-20, 1999. pp. 362-369.
- [41] Extensible Markup Language (XML) 1.0 <http://www.w3.org/TR/REC-xml.html>