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 and Bill Serra

Internet and Mobile Systems Laboratory

Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94304, USA
{timothy, john_barton, morgan}@hpl.hp.com
www.cooltown.hp.com

Abstract

The convergence of Web technology, wireless networks and portable client devices provide new opportunities for computer communications systems designs. At HP Labs we have been exploring these opportunities through an infrastructure to support "web presence" for people, places and things. Our goal is a bridge between the World Wide Web and the physical world we inhabit. This bridge includes the ability to interact with devices such as printers from a browser using standard HTTP communication. It also includes the ability to provide people, places and things – electronic or otherwise – with a web resource that is used to store information about them and which is automatically correlated with their physical presence. We aim to provide users, particularly mobile users, with support for their everyday activities, which mostly concern physical objects other than PC's.

In this paper we characterize the functionality that we formulate as web presence for people, places, and things. We describe our infrastructure for web presence, and applications that demonstrate the infrastructure's viability.

Keywords

World Wide Web; Web presence; people, places, things; location-aware computing; smart environments; ubiquitous computing.

1 Introduction

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.

We aspire to enhance physical entities with virtual services, whether or not the entities themselves have electronic functionality. Databases and web pages contain petabytes of information about the physical world, and yet this information is *physically* dissociated from it. A user stands before a monument and is not sure exactly what it is. Another picks up a document and doesn't know where it came from or what is supposed to happen to it next. A potential wealth of electronically stored information exists about these things. And services exist that could relate the artifacts to others and to the processes that affect them. Yet these can only be accessed somewhere else – sitting at a computer.

We think the physical world and the virtual world would both be richer if they were more closely linked. More and more of the physical world is becoming "smart", and users need a convenient framework in which they can benefit from the smart artifacts around them. Computing power is being embedded in devices that don't have a usable interface of their own, because they are too small or because they are inconveniently placed. Computing power is in our car engines and may soon be in our smallest domestic appliances. So the interface needs to be away from the device and somewhere at hand – on a PC or a personal digital assistant, for example.

The facility to interact in these ways amounts to a virtual bridge between users and physical entities. This bridge must be dynamic and contextual. There is an increasing requirement to give users the ability to interact with the physical world wherever they happen to be. Users are nomadic: they move around and as they do so they require interaction with the resources in their current situation, while still needing contact with people and services elsewhere.

The convergence of Web technology, wireless networks and portable devices provide new opportunities for computer communications systems designs. At HP Labs we have been exploring these opportunities through an infrastructure to support "web presence" for people, places and things [7]. Our goal is a seamless marriage of the World Wide Web with the physical world we inhabit.

In this paper we describe our infrastructure and the demonstration projects that prove its viability as a basis for web presence. Our presentation explores the concept of web presence. The demonstration projects implement part of the system and they verify our progress. However, in a sense the projects themselves are more fundamental: through our experimentation with Web technology and wireless portable devices we came to understand the possibilities that we now formulate as web presence for people, places, and things.

We start in section 2 with a definition of "web presence" and the user-model that it supports. We will refer to concrete scenarios as much as possible. We believe that web presence will allow these scenarios to be implemented, but we don't claim to have proven that as yet. In section 3 we sketch the requirements for a system that supports web presence and give our infrastructure that meets those requirements. Also in section 3 we describe our demonstration projects that show the infrastructure is usable. While we will highlight our view and the projects we have completed, we believe that many web-centric appliance efforts are converging as hardware and networking trends become clear. Section 4 relates our work to that of other efforts.

2 People, places and things

We will 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 [30]. 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*.

Web presence is the representation of people, places and things on the web. Presence is "The state of being … within sight or call, or at hand." Web presence then is the state of being within reach on the web. Web presence brings the power of the Web information system to the physical world and it brings more of the physical world on to the Web.

The Web derives its power from a design that has lead to widespread accessibility. The web offers accessibility that is widespread in two senses. First, resources on the web can be accessed from any device that supports the standard HTTP protocol [18]. This includes general-purpose devices: personal digital assistants (PDAs), laptops and PCs. It also includes information "appliances" dedicated to a specific function, such as digital cameras and printers. Second, the web allows us access to resources from afar transparently. 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. Web presence tries to leverage the Web's broad accessibility, using both its potential for the standard access and its ubiquity of networking.

The twin advantages of a broad deployment and global remote access can be especially valuable in a world of wireless, portable networked devices. These devices provide a means for people to interact with and communicate through the Web as they move from place to place. Consequently these wireless, portable devices provide the means for people to interact with web-places reflecting the real-world place they are in. This includes the electronic manipulation of real-world things and the electronic augmentation of real-world meetings with people.

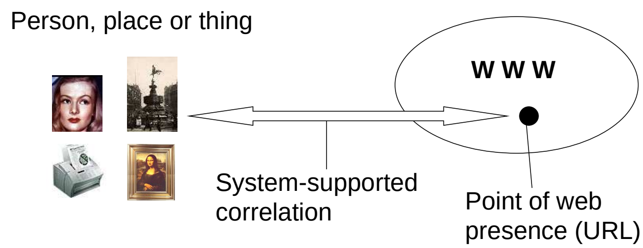

Figure 1. Correlation with a point of web presence.

Notice that the definition of presence is expressed in terms of the human scale – “within sight or call” or “at hand”. While the Web is a global-scale information system for which physical location is transparent, we often want to access resources that are conveniently near our physical location. For example, we tend to prefer restaurants and printers that are nearby. This will surface in our need to show location-dependent views of services and in the need for new security for web-systems.

2.1 The Meaning of Web Presence

We define *web presence* for any person, place, or thing to be accessibility via the Web through a system-supported correlation between a web resource and the physical entity concerned. Web presence means that the entity is bound to a resource that has a URL [3] and is accessible by the standard HTTP protocol. We shall call this resource the entity’s *point of web presence* (see Figure 1). This means, in turn, that a web-present entity has a web page.

However, web presence means more than having a web page. We frequently access web pages describing physical entities. For example, consider a user Veronica who is visiting San Francisco. Veronica can read a web page about the San Francisco, another about Harry her host while she visits, and a page about Coit Tower in San Francisco that she may visit. This is all useful to Veronica, but web presence means more than these web pages for three reasons. First, a point of web presence is supported by network systems. Second, we may be able to manipulate the point of presence to control the physical entity. Third, the point of web presence can be manipulated by other web services. Let’s examine these in more detail.

1. System-supported correlation: Places for Web Things and People in Web Places. In the example given above, Veronica has somehow learned the URLs of web pages about San Francisco, her host Harry, and Coit Tower. But there is no system-supported relationship between the URLs and the corresponding entities. By contrast, a web-present entity’s URL is correlated systematically to the physical object.

Consider an alternative version to the scenario above. On Veronica’s arrival in San Francisco, her PDA electronically picks up URLs for pages about the city itself and the places within it that she travels through – the railway station, a shopping district, and a café – and it presents these to her as web links. These places, in other words, have web presence. The URLs are targeted at the physical places. The URLs are distributed within them through wireless broadcasts of restricted range. Veronica’s PDA senses these links and she may select them as she travels.

Suppose Veronica needs to communicate with Harry. At the time she decides this, she is sitting a café, and she does not know where her host Harry might be located. Through a wireless link provided by the café, she calls up a web page for Harry and clicks on a link marked “communicate”. Harry gave Veronica the URL of his web page, as the address of his point of web presence. However, the service at the end of that “communicate” link has correlated Harry’s physical presence in his current location with a web resource giving a convenient means to communicate with him. He happens to be near a telephone in a colleague’s office. When Veronica clicks on the “communicate” link, that telephone rings and a telephone application pops up on her PDA. If, for example, Harry had instead been most conveniently reachable by email, an email program would have popped up instead, with an email address for Harry supplied by the system. Harry has Web presence and Veronica can use it to communicate with him.

After talking with Harry to arrange their schedule, Veronica decides that she has time to visit Coit Tower. While there, she finds artifacts on display with small electronic tags next to them. Veronica's PDA is equipped with a small tag-sensor. When she places the sensor on or near the tag, her PDA displays a web page about the artifact, providing her with information about it. These Coit Tower artifacts have Web presence.

2. Web Things and Web Places: Control and interrogation of devices. An even stronger type of correlation is the ability to control a web-present device through its point of web presence. To continue our example, Veronica has to print a document when she arrives at Harry's organization. She points the infra-red port on her PDA at a nearby printer, which causes a web page about the printer to be displayed on the PDA. This page contains fields to choose the printer settings and select the file to print. The document is transferred for printing over the infra-red link. The printer has a point of web presence that users can manipulate.

Later, Veronica realizes that she has not switched on any lights at her home, which will be empty that night. She clicks on a link to her home, which provides her with web pages showing the state of all her domestic appliances. Using a web form she turns on a selection of lights. Veronica's home is a Web place that contains Web things, things representing her appliances and lights.

3. Web Things Talking to Web Things: Access by devices. Things that possess their own computational resources can themselves exchange web content with other, web-present entities, without a human around to click on a web link. For example, a "home management" device in Veronica's empty home could respond to the absence of people as night falls by automatically switching on the lights and television. It detects the absence of people by interrogating a security monitoring service on the home PC linked to the alarm system. The particular forms of web content exchanged in this case describe the state of the appliances.

2.2 Modes of web presence

The correlation between an entity and its point of web presence can be supported internally or externally:

Internal Support. A *web present device* (or *web device*, for short) is a device whose physical state is electronically readable and/or settable via HTTP operations on its point of web presence. A web present device supports correlation with its web presence directly through its internal computer. We gave examples above of how devices such as printers and home appliances could have their state read and manipulated by humans and by one another. The correlation is achieved through a web server embedded in the device or the device can be controlled remotely by a web server acting as an HTTP gateway to the device.

External Support. Non-electronic entities cannot have an embedded web server. In this case we need a way to correlate a person, place or thing with its point of web presence. This can happen manually by having users read URLs attached to or nearby objects, but a useful web presence system should support electronic sensing. We saw how Veronica was electronically supplied with the URL of a place when she physically visited it, and she viewed web pages about artifacts whose tags she electronically read. Physical proximity to the entity may not be the only physical circumstance we want to correlate with a point of web presence. For example, an action, such as removing or placing a container on a shelf, can be used to automatically update a web page about that container [38].

Clearly, people and places are only ever externally correlated with web pages, through sensing technologies. On the other hand, things may be externally correlated with web pages, or they may be web present devices, or both.

2.3 Web presence for things

The typical user experience we seek with web presence for things is that of collecting links to the things' points of web presence as he or she encounters them in the physical world. One could even think of the physical world as a web page, with links at certain physical points. But of course the links are 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. While this physical/virtual browsing is a typical scenario, we

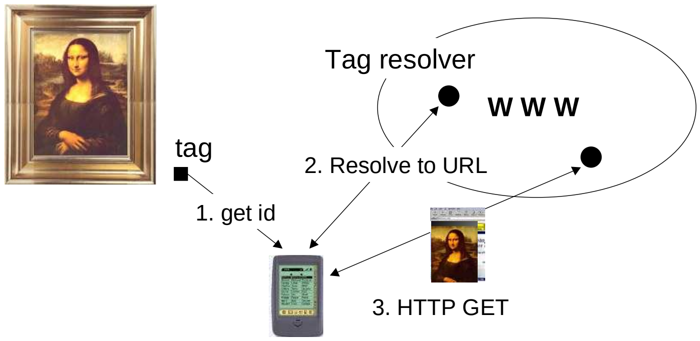

Figure 2. The web presence of a painting.

imagine other usage scenarios for web present things, in the same way that the Web supports many usage scenarios in addition to Web browsing.

How does the user collect the points of web presence? The user's electronic client device collects these URLs by discovering them via a network system or by 'sensing' them electronically.

For network system discovery of URLs, the client device must access the network and participate in a network protocol for discovery. Web present devices can participate in this same network protocol. The protocol provides the client device with the web-present device's URL. For example, Veronica's rental car could be running an embedded web server able to present information about its current engine state as web pages. She could use her PDA to calm her fears about the unfamiliar clunking noises that it makes. If the noise indicates a problem, a local mechanic can obtain authoritative parts information using the same web discovery service. His work would be recorded in the car's web device for future reference by the rental-car company. We discuss the discovery protocol further in Section 3.2.

Sensing is another way for the client device to find URLs. The result of sensing can be direct – the sensor obtains a URL – or it can be indirect – the sensor obtains a datum such as an identifier that has to be looked up to obtain a URL. Things with direct sensing give URLs: we can access the web point of presence directly. Figure 2 shows indirect sensing. A tag next to a painting is sensed and its identifier resolved to obtain the gallery's web page about that painting.

Distribution of the URL or other datum to a sensor can be active – pushing information – or passive – awaiting a sensor to request information. *Beacons* are devices that push information to any sensor listening within its range over a wireless network. For example, an infra-red beacon can be configured to emit a URL to any sensor that comes within its range. On the other hand, electronic tags like RFID tags [31] or iButtons [22] are passive. These are small devices that supply a unique identification string (like a bar code) or a URL to a sensor placed near them or in contact with them. (A UPC bar-code could be used instead of an electronic tag, but we believe that electronic tags will come to predominate for many of the purposes we intend.)

Finally, sensing technologies may or may not require deliberate action on the part of the user, and this distinction leads to different usage patterns. Broadcast beacons, which advertise their value to all devices in range regardless of their orientation, are useful where users are grateful for whatever information they can receive without effort. Technologies such as tags and infra-red beacons, by contrast, require the user to position his or her sensor deliberately to obtain the URL. This is useful when users are more selective about the points of web presence that they wish to collect.

2.4 Web presence for places

Web presence for places extends the idea of collecting links as the user encounters entities in the physical world, to include the potential for location-specific *web portals*. The Web presence for a place is a web site devoted to information about that place, a physically-based, local equivalent to World-Wide Web portals like Yahoo [37] and

Excite [11]. As a point of web presence, the web portal is systematically correlated to the physical place. The correlation consists of a combination of the following:

- The place contains a beacon or a tag that provides the URL of the place's portal. Any device that is in range of the beacon or the tag can obtain this URL. This means that people who physically visit a place can find out about it from the web portal, using the link to the place that has been provided automatically to them.
- Some of the contents of the web portal are provided automatically by services within the place. This includes the web presence of people and things that are physically within the place. Services create pages in the web portal that aggregate links to these points of web presence. Thus the system creates a virtual place to represent the physical place, semi-automatically.

For example, consider Veronica, who is visiting Harry in a meeting room at his organization. There is a tag on the wall of the meeting room, at which she can point her PDA to obtain the URL of the room's web portal. The portal contains a link to a web page with a map showing the printers in the organization that is available for guests to use. This page was produced by a service available in the place. The map shows where Veronica is located in the building. When Veronica clicks on a nearby printer, she is presented with the web page that is the printer's web presence. She uses this page to set the printer options and print her document.

A place is a context for service provision based on a physical domain permeated by one or more networks. We are interested in utilizing wireless networks; but much of what we shall say relates also to places with wired networks. In a place, people and devices access the services presented by the place. Many, but not necessarily all, of those services are based around the people and things in the place – those that are physically present or nearby. For example, one service could provide attendance lists for the people in the place; another could provide a printing service based around printers in the place. Places may also provide global services for accessing resources other than those physically inside them. In particular, users may be granted access to the rest of the Internet via a wired connection.

Note that a single physical space can correspond to several web-present places. For example, the same venue could have one web portal reflecting its use as a rock concert, and another when it is hosting a computer technology exhibition. Harrison and Dourish [17] describe this distinction between space and place thus: "space is the opportunity, place is the understood reality."

2.5 Web presence for people

A person's web presence provides information about them and a way of communicating with them, wherever they are. The same person may have several points of web presence. These include a global presence that presents information that is relevant everywhere and place-specific points of web presence.

A person's place-specific point of web-presence appears as a link in the web portal for that place. It is created and in some cases updated automatically, although with the user's consent and involvement. Various types of information are relevant, depending on the circumstances:

- The person's identifying attributes. These are sometimes place-specific. For example, the name we choose for ourselves may depend on the degree of informality. In some places we may choose to be anonymous.
- The history and current state of the user. For example, are they physically present now? What services have they used?

It is essential that the user can maintain control over which information is made accessible to others. The user may decide to have no web presence in a particular place. For example, Veronica in our example may choose not to create any web presence in the Coit Tower web portal when she visits there. On the other hand, she may feel that being web present in Harry's building will enable someone in Harry's organization that knows her to realize that she is there.

Moreover, Veronica may wish to be accessible by colleagues and friends outside Harry's organization while she is visiting him. For this she needs a global point of web presence in the form of a web page that directs attempts to communicate with her wherever she is currently, without necessarily disclosing her whereabouts [27].

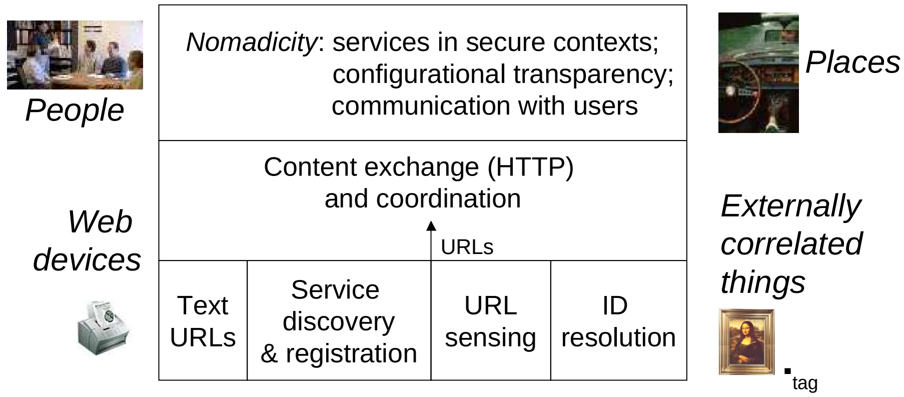

Figure 3. Infrastructure layers

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”, 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 the 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” (Jini [24]), or “CORBA presence” [29]. The principal advantages of the web over other systems are 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.

Figure 4. Museum visitor's PDA screen.

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. The middle layer contains the web itself, which implements content exchange through the HTTP standard. The rest of the infrastructure consists of adjuncts to the web. 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. At the top layer is infrastructure to provide services related to places and nomadic users.

3.2.1 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 our infrastructure must also support sensing and automatic discovery of web-present entities as described in section 2.2, to make web presence easy to use.

Figure 4 shows a screenshot, from our demonstrator implementation, 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. However, not all links persist. Veronica may find that a link ceases to work after she has left the museum.

The paintings' points of web presence provide information about the paintings themselves, and links to related pages. The museum bookstore's web portal provides services to assist Veronica in buying books and in gathering materials according to her interests. These include a search service, so that Veronica can look up titles that she cannot find, and it includes a service to order books that are not available. Both these facilities relieve Veronica from having to wait for assistance from the store clerk. In addition, the bookstore offers a printing service, for visitors to print out web pages that they collected as they toured.

Discovery

When Veronica enters an area of the gallery with wireless networks compatible with her PDA or when she attaches her computer to a public access point on a wired network, then her device can send network packets via multicast routing to a "service discovery" address. This well-known, preconfigured address is the meeting place for network services. 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

Figure 5. A web presence for *Mona Lisa*.

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.

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, 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 when visitors point their infra-red ports at it. The URLs that they emit are configured by hand. In this case 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 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 an electronic tag. Figure 2, above, 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.

We have produced a prototype implementation for sensing things tagged with iButtons (which emit a unique identifier) and displaying a corresponding context-dependent web page (Figure 5). The sensing device is configured with the URL of a *tag resolver*. When the client senses a tag, it sends the tag identifier to the resolver. If the resolver has an entry for the identifier, it sends back the corresponding URL. Otherwise, it sends back a response indicating the absence of an entry. The user chooses 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.

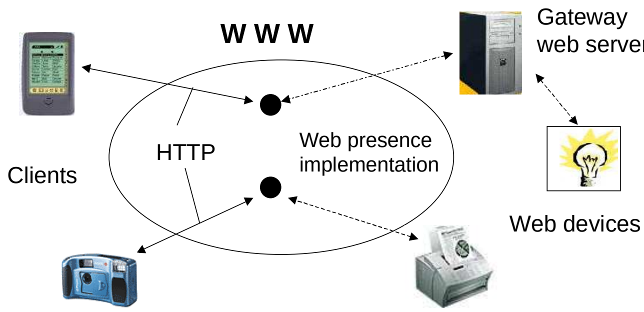

Figure 6. Web devices (printer and light), with and without embedded web servers.

We have begun to address two problems with this scenario. The first is the question 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. The second problem we are addressing is that of how to resolve tags for which no resolver is known *a priori*. For example, suppose that all manufacturers tag their goods and store tag-to-web-page mappings on a web site of their choice. A store identification system needs to locate the manufacturer-supplied web presence of an arbitrary item it sells, with no information other than the tag identifier. A resolution system with properties similar to the DNS seems to be required [12]. This is the subject of ongoing research.

3.2.2 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 6 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 a JPEG image. Figure 6 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.

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 *chailets*, which handle HTTP requests. Chailets 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, out of efforts such as the Universal Plug and Play Forum [34]. A service also has a list of named *entry points*, which distinguish between functional units of the service and correspond to chailet method names. Chailets have methods for setting and reading *properties* – attributes that conveniently correspond to the state of the underlying device.

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 chailet instance named *laser15* which implements the service name *print*. The chailet 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 stops in. She should be able to print a document from her portable device without this having to involve any nearby servers. This could 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 and then the client device can deliver the document.

We have built an interesting alternative version of the walk-up and print scenario that we call *eSquirt*. 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 sends the image from her PDA. 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. In that case, 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 itself obtains a full-fidelity version of the web page over the Web and prints that.

3.2.3 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. The cases that particularly interest us involve wireless networks, because of the convenience of the interconnectivity they afford. Examples are a railway station covered by WaveLAN connectivity [36], a café covered by Bluetooth [4], and a home covered by Bluetooth or HomeRF [19].

People and things underlie many of the services provided in the place through their web presence. But the services available in a place are usually not known *a priori*. Users can *discover* services, by browsing the place's web portal. Users can also access devices indirectly by requesting services by name. If, for example, Veronica decides to print a document in her hotel room and the room only has one printer, her client device will attempt to "discover" a printer, and finding only one, print her document there.

Inside a place 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. Several discovery and registration services have been designed to scale to large numbers of clients and servers, and at the same time to operate in simple environments such as the home. These include the Service Location Protocol [16], the Secure Discovery Service [8], the Simple Service Discovery Protocol [15], Jini's discovery service [2] and Salutation [33]. Of these protocols, only the Simple Service Discovery Protocol leverages web standards where it can.

Even things that are externally correlated with their points of web presence can be established as existing in a particular place. Consider a user carrying a device that knows it is within a particular place – because of a beacon nearby, say. When the user uses that same device to sense a tag attached to the thing, the system can automatically infer the thing's presence in that place, and update the thing's point of web presence and that of the place automatically. This is an example of a general principle of transitivity which is often useful: if an entity x is in place P and if y is close to x, then we count y as being in P also.

Places can physically overlap and they may have fuzzy boundaries, so it is usually necessary to disambiguate which services are available in which places. This is in part a physical question. 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 wireless 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 which underlies the web portal. It is responsible for providing secure views of the set of entities that are present in the place and the services based around them. What is deemed to be "in" the place is a matter of policy, which the place manager implements. In general, the content that the place manager provides to a particular user is also policy-driven. The content depends not only on the client's security principal, the user; it also depends on that principal's preferences, and on the client device's functional capabilities.

Our PlaceManager employs the SLP service discovery protocol to track the web devices in the place; and it employs ChaiServer's standard web-based authentication features in order to provide user-dependent views. We provide a web page generation tool, which automates many of the place master's tasks in specifying the web portal's pages and their views. More details are available in [5].

3.3 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 around the Global Positioning System (GPS)*: 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 to things 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.

SecureWebTunnel

HP's SecureWebTunnel work addresses the following example. Suppose Veronica is in the Coit Tower cafe and wants to fetch a business plan before meeting Harry. SecureWebTunnel provides a secure communication channel through a pair of proxies, one in Veronica's PDA, and one in the firewall of Veronica's company intranet. The client

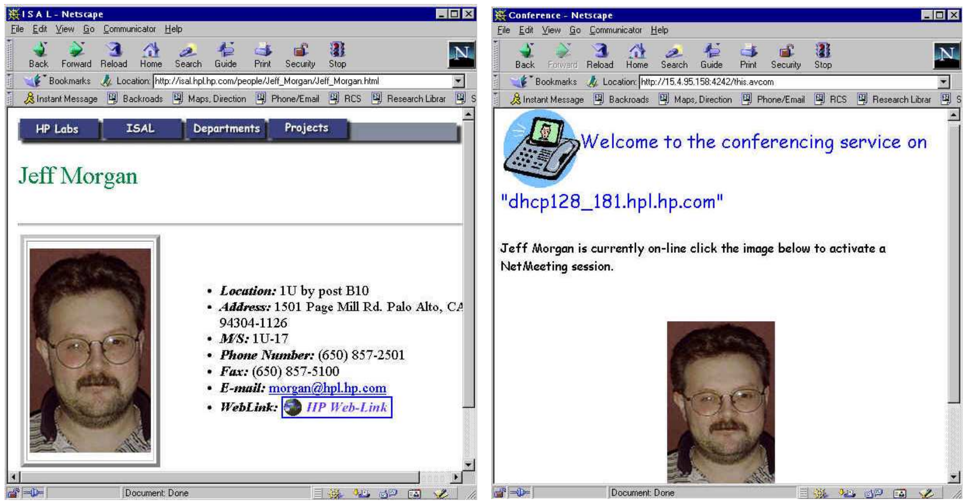

Figure 7. WebLink. The left-hand page shows Jeff Morgan's home page with the "WebLink" link; the right-hand page enables the user to communicate with Jeff by NetMeeting at the machine Jeff currently works at (other forms of communication are available).

proxy in her PDA exists to make the authorization process more convenient for her. When the client proxy is given a URL for a location that is inside the intranet, it behaves differently according to whether the client proxy itself (i.e. Veronica's PDA) 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. A secure SSL [13] connection is made, using the private key or other authentication material obtained through Veronica's PDA. The SecureWebTunnel proxy in the firewall verifies the security information and forwards the URL request if authorization is granted.

The SecureWebTunnel work is similar to that of Abadi et al. [1]. Similar functionality is also addressed by the Satchel project [14]. An outstanding research problem is how to implement polices and manage more selective (for example, time-limited) 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, then 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 now offer it via the link in Harry's web page.

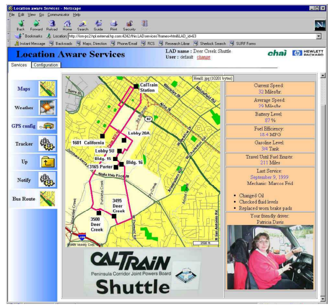

Figure 8. A web-present bus, equipped with a GPS receiver.

With WebLink, the communications channel that Harry and Veronica end up using 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 [27] 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.

Figure 7 shows two of the pages encountered by a user who clicks on a WebLink link to one of the authors. After clicking on the link in the author’s home page (left-hand page in Figure 7), the user is presented with a page presenting communication options (not shown), including email-like facilities. The author is sitting at a machine where he can run Microsoft’s NetMeeting currently, and the user chooses this option to communicate with him (right-hand page in Figure 7).

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 book store.

Figure 8 shows a view from our demonstrator for those waiting for the “HP shuttle” bus. The bus’s web page shows a map of the shuttle’s route on which is marked the bus’s current location. The services that are available from this page include notification, to alert users of the bus’s approach. The user can click on the map to set a “notification point”. When the bus reaches that point, the system alerts the user by email.

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 more connection between the physical and the virtual, 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 too far from interesting 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 [35], 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 [32] 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.

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.

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.

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 [34]. 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 [28] and WebL [26].

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/.

Acknowledgements

The authors would like to thank Ilja Bedner and Celine Pering, who assisted with the CoolTown demonstrator applications.

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. Creating a Web Representation for Places. Submitted to WWW9 conference.

[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. Joe Foley, MIT. Personal communication.

[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. pp 67-76.

[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? To appear, *Proceedings 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] 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.

[28] Merrick, P., and Allen, C. Web Interface Definition Language (WIDL). W3C Note, World Wide Web Consortium, 1997; http://www.w3.org/TR/NOTE-widl.

[29] The Object Management Group home page. http://www.omg.com/.

[30] Potel, M., and Cotter, S. *Inside Taligent Technology* (Addison-Wesley, 1995).

[31] RFID home page. http://www.aimglobal.org/technologies/rfid/.

[32] 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.

[33] Salutation Consortium home page. http://www.salutation.org/.

[34] Universal Plug and Play Forum. http://www.upnp.org/.

[35] 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. pp. 370-377.

[36] WaveLAN home page. http://www.wavelan.com/.

[37] Yahoo home page. http://www.yahoo.com.

[38] 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.