Physical Registration: Configuring Electronic Directories using Handheld Devices

John Barton (john_barton@hp.com) Tim Kindberg (timothy@hpl.hp.com) Shreyas Sadalgi (sadalgi@cs.rutgers.edu)
Internet & Mobile Systems Lab Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94304, USA
http://www.hpl.hp.com

Abstract

An important problem in pervasive computing is how to create and maintain manageable collections of pervasive resources appropriate for disparate and typically nomadic users. This paper describes *physical registration*, a new, human-centric mechanism for constructing collections of resources in the form of electronic directories. Users identify physical objects using sensor-equipped, wirelessly networked handheld devices to build directories of virtual "entities" associated with those objects. The entities can be web resources accessible over a network, like printers, projectors, picture frames, or storage devices. They can also be web resources correlated with objects like books or telephones that are not themselves accessible over a network. Our paper describes and motivates the method, then compares two implementations that we have constructed.

1 Introduction

In a "pervasive" or "ubiquitous" computing environment, microprocessors and thus software components are embedded in many everyday objects such as digital pens and picture frames. Also users carry portable devices such as personal digital assistants (PDAs). The users' devices and the resources in the environment are networked, typically wirelessly, so that users can access those resources. Moreover, even simple, non-electronic objects such as books, museum exhibits and supermarket products will be associated with resources such as web pages about them, resources that can be accessed by physically identifying the objects themselves. In short, over time, the average number of networked resources per cubic meter is rising and those resources are becoming accessible in hitherto unexploited places and situations.

A fundamental problem is that of how to manage interactions between these myriad handheld and environmental resources so that they fall under a scope that is appropriate for users and their applications. That scope should also be of reasonable size so as not to overload our systems. In particular, those are issues for *nomadic* users, who want to set up associations between their portable devices and the resources in their current environment. We need mechanisms for constructing associations that are appropriate for nomadic users, as opposed to those who have their computational "home" in those environments. We need to avoid what might otherwise be a combinatorial explosion of attempts to discover partners for interaction. Simple physical proximity between two resources does not automatically imply they should be associated. The same is true of network proximity. Networks – wireless networks, in particular – do not always respect territorial or administrative boundaries.

This paper tackles the problem of organizing associations in nomadic computing environments: pervasive computing environments that offer local resources as well as remote resources to users who travel within and between them carrying wirelessly connected portable devices. Our system enables those who administer localities to construct directories of the resources that they offer to nomadic users, selecting just those resources that are appropriate for nomadic users out of all the resources that they might otherwise offer.

Electronic or online directories provide a systematic resource for online users. Web users are familiar with global directories of the World-Wide Web, such as Google[1] or LookSmart[2], and more specialized directories like Yahoo[3] or SlashDot[4]. Enterprise computing users are familiar with LDAP directories for phone numbers and addresses[5]. By collecting and possibly cataloging or providing searches for resources, these directories make resources more available to users.

Directories are especially important for nomadic users. Off-line nomadic users consult yellow pages of phone books, study maps, and seek out information kiosks when they enter unfamiliar territory. As wireless networking begins to pervade, we expect these nomadic users to be online and to seek information from online directories [6,7]. Because nomadic users must acquire information in unfamiliar places, supporting their needs will drive the development of easy to use directories. This in turn will make such directories more broadly attractive to users.

Every directory needs a solution for placing entries in the directory (registration). Unlike directories on the Web that list public virtual entities like documents, e.g. Yahoo, or directories inside a business that list private physical resources like network printers, e.g. LDAP, directories for nomadic users will list public, physically accessible resources. Consequently electronic directories for nomadic users will need good solutions for registering physically accessible resources.

This paper describes how to build electronic directories of real world physical entities using physical manipulation of handheld devices. That is, we have designed a system using the same devices and network infrastructure developed for nomadic end users but employed them in the administrative task of configuring directories. We call the process “physical registration”.

The entities in such a directory [6] can be web resources accessible over a network, like printers, projectors, picture frames, or storage devices or they can be resources like books or telephones that are not accessible over a network. The physical entities without a network presence are represented by a web resource that may be descriptive information (a web page) or may be active sources of dynamic content. The physical entities can also be people as represented by their web home page. Finally these physical entities can be places: physical areas represented by a web directory describing the place and collecting reference to entities in the place [8,9]. This collection of references, specifically a collection of URLs, creates a directory for the place. The electronic directory of these entities allows users with information appliances, laptops, smart watches, PDAs, for example, to discover these entities electronically, learn about them and their relationship, control them or modify information about them. Our paper describes a method of constructing such a directory using network-connected handheld devices that can sense identifiers attached to the entities.

For example, imagine that Acme Software Inc. sets up a stand at the Wireless Web World exhibition. That stand is a place with a physical extent and a purpose. Acme has to define its own virtual presence corresponding to that place within the exhibition, which it uses to provide information, printing, purchasing and other services to users who walk up to the stand. Visitors must be able to discover the Acme presence when they walk up, distinguish it from that of neighboring exhibits, and, through their devices, these visitors must be able to browse and access the resources that Acme provides. Our goal is to explore the application of handheld devices with sensors and network infrastructure to support the Acme exhibitors as they configure the online directory that represents their exhibit so that the directory appropriately supports the visitors.

1.1 Contribution

We describe the deficiencies in current registration schemes (Section 2) and explain a model for resource discovery by nomadic users (Section 3). We thus motivate physical registration as a model for nomadic computing environments (Section 4) that enables administrative users to construct directories appropriately, applying their knowledge of social, administrative and other constraints that pertain in real-world environments. We describe and compare two implementations of a prototype that demonstrates physical registration. The usage scenario we envision has an administrative user with authority for a place directory—a “registrar”—taking a handheld device equipped with a barcode reader into a place to be provisioned for mobile users. The registrar selects the directory to be updated by sensing a barcode attached to a wall in the place. The registrar then begins to select entities like printers, scanners, projectors, even tables and books, scanning their bar-codes and thereby entering them in the place’s directory. In Section 5, we discuss our approach.

2 Related Work

An inventory assessment system has some of the same elements as those described above: it has users with devices for sensing tags on objects, a mapping from the tag identifier to information about the object, and an inventory of objects. An “inventory” is a “directory”.

However, inventory systems have restricted goals and a closed architectural approach. Typical inventory control systems construct the inventory to assess stock on hand. A retail store that placed its inventory online for customers in its store to browse would come close to our use for these technologies. However we have in mind that this technology would be broadly available. As to architectural approach, inventory systems tend not to use technologies that can be combined in other applications. In a typical inventory control application the sensor would be dedicated

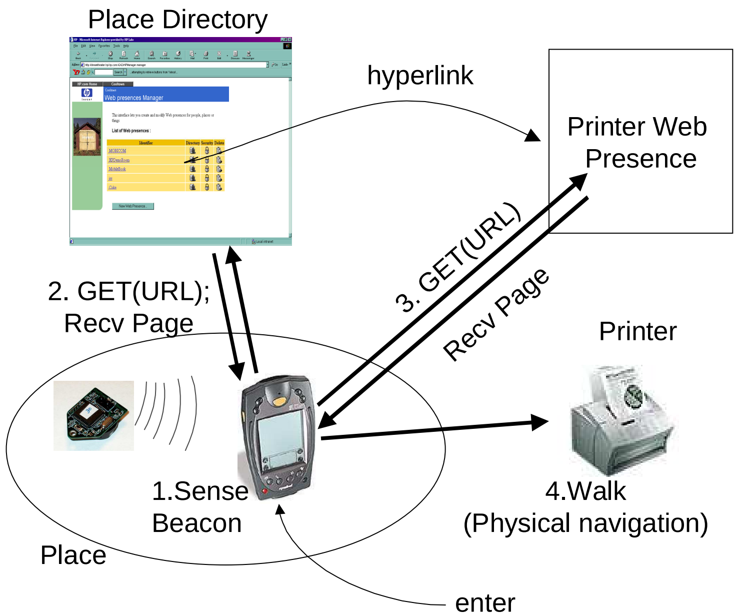

Figure 1. Physical discovery of a printer.

to inventory control. Typical inventory control systems would use private proprietary communications, concentrate on database operations, and only support specialized clients. For nomadic computing systems, we seek to build our application by combining technologies, each of which participate in other applications (in many nomadic computing environments). Thus we require that the sensing technology in our system be used for location sensing and for physical discovery, as well as for physical registration. We also require continuance of our web-based approach to each step[7]. Thus we use open communications standards, Web URLs for addresses, and support web clients.

Text entry and multicast network search are two other mechanisms for registration that are designed for more open electronic directory systems. However, they are not ideal for configuring directories for mobile users. The "text entry" method of registration employs a registrar, a person skilled in the configuration of the infrastructure. The registrar creates the directory entries one at a time, either directly in a text file [5] or by responding to prompts from an infrastructure management program. This approach requires the registrar to know all details about the entity to be registered as well as the possibly intricate process for directory update. Furthermore this process is vulnerable to typographical errors during the text entry.

The other approach to this problem relies on network multicast [10,11,12,13]. Entities connected to the network listen for particular network packets that signal discovery. These packets have queries for kinds of entities required by the user's application. The listening entities reply if they can satisfy the query. The directory is therefore distributed among the entities. A special resource can act as a cache or "directory" by multicasting generic queries, storing the results, then responding to client queries itself.

This approach need not require a registrar, but it is limited to entities that can communicate via IP multicast. Moreover, it relies on subnetwork topology: while subnets are correlated with physical areas, they are not identical. For example, consider a large workplace or a small business or home. The network topology for these places will be set by Internet access support organizations that lay down wires and wireless networks and allocate addresses as demand grows. The IP subnets will roughly reflect geography, but without extraordinary effort the resources on a given subnet will only approximate a given users needs. In addition, geographical areas are often used for multiple

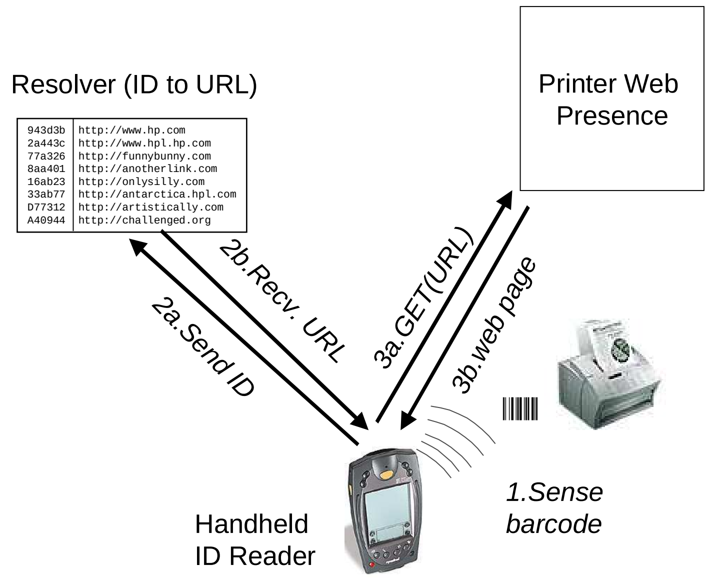

Figure 2. Physical sensing of a printer's web presence.

purposes or by multiple sets of users with different needs. Multicast works well when the networks are installed and maintained for the exclusive use of a coherent user community. For example, when its users deploy a small network, the subnet they obtain will naturally reflect their work location.

Another drawback to the network multicast approach—as with all fully automated registration processes—is that descriptions of resources suitable for machine processing are inevitably limited and inflexible. Many of the high-level subtleties of human concerns such as local usage conventions that come into play in deciding whether, for example, a particular printer should be registered, are best handled by a user.

3 Physical Discovery

We start with a description of physical discovery using the same terminology that we will use for registration. This has two purposes: 1) we introduce the basic ideas and technologies before considering registration-specific issues and 2) by showing end-user scenarios for discovery we show that the technologies we need for registration will already be in place rather than deployed specifically for physical registration.

We shall cover two scenarios. In the first a user with a handheld web browser discovers a printer near their current location. In the second, the user finds a source of printer supplies for the printer they previously discovered.

Discovering the physical printer. For the first scenario we imagine visitors to our laboratory who desire hard-copies of some local documents that they looked at during their visit, for example presentation slides, demo information, or this paper. They are in a conference room where they do not have any information or access for a local printer. To locate a printer, they select an infrared beacon in our conference room that emits a URL pointing to the room's "web presence" – the web resource correlated with it. That resource is the directory for our room presented as a web page (see Fig. 1, step 1). Their handheld web browser obtains the directory web page and they can see a link to a nearby public printer (Fig 1, step 2). They select a link on the printer's page that gives a map to the printer (Fig. 1, step 3); they use the map to walk to the printer (Fig. 1, step 4). Once there, they can transfer the URL to print the document [6].

Discovering the printer's web presence. The second discovery scenario runs from a physical printer to the printer's web presence. The usage scenario could be a user that finds a printer requiring ink. The location of ink cartridges or order forms for ink can be specified on a web page. To obtain this page, our user senses a barcode attached to the printer as shown in Fig. 2, step 1. The barcode is read as an identifier and this identifier is resolved to

URL by lookup in a database managed by a server we call a "resolver" (step 2). The URL that returns gives the web page for the printer; it can have links to forms for ordering supplies locally. We call this kind of discovery physical sensing of an entity's web presence.

This second scenario used an identifying tag and a network-embedded resolver service to get a URL, while the first scenario used a direct URL beacon. An in depth discussion of the use of tags and resolvers is given in [14]. Through out this paper we will assume that the resolvers exist and that the have been configured with appropriate URLs. Such configuration is itself an important problem (see also Section 5).

We can relate the two discovery scenarios as shown in Fig. 3. In the first scenario, our user discovered the physical printer using web navigation. First they discovered the room's directory (maintained by a server called a "PlaceManager") by physically selecting its corresponding beacon (Fig. 3, step 1.1). Next they discovered the printer by traversing a hyperlink on the place page (step 1.2) and reading a map that came from a page about the printer (step 1.3). Finally they walk to the printer (step 1.4). In the second discovery scenario we proceed from the physical printer to its web presence (Fig. 3, step 2.1).

4 Systems for physical registration

Physical registration sets up a directory to point to a set of related entities by using handheld devices that can sense object identifiers. The directory may correspond to any type of selection that the registrar finds appropriate for the expected clients of the directory. It could, for example, be a directory of interesting artwork, supposing that the *objets d'art* were tagged with identifiers and thus had been given web presences. Or the directory could be that of the (heterogeneous) resources in a single place that are to be made publicly available to visitors. The type of content in the directory is immaterial to our registration approach. The critical element is electronic support for a human regisitrar who makes the selection, physically navigating to and designating the corresponding physical objects.

Fig. 4 shows how the user experience of registration is complementary to discovery, for the case of a place's directory. The registrar selects the place to be configured (step 1), walks over to a physical resource like a printer that should be in the place's directory (step 2), senses the entity's web presence (step 3), and adds the address for that web presence into the place's directory (step 4).

We developed two implementations of physical registration. The first one, the "device-centered approach", relies on a specialized application written for the handheld device. The second one, the "infrastructure approach", uses only a modified web browser in the PDA and puts more logic into services. We shall describe these approaches in the next section.

4.1 Device-Centric Physical Registration

The device-centered approach achieves physical registration by coordinating services through an application running on the registrar's handheld device. The device-centric model drives the identifier sensing, lookup and directory-update from the device indicated in Fig. 5. First we sense the barcode to obtain an identifier (ID). Next the application on the device sends the ID to the resolver (step 2a) and obtains a URL (step 2b). The protocol here can be quite simple: the send can be an HTTP GET to the resolver URL with the path of the URL being the digits from the barcode. We use the resolver detailed in [14].

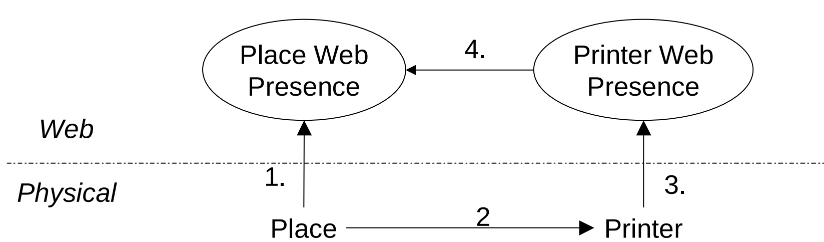

Figure 4. User experience in physical registration. The numbers correspond to steps described in the text.

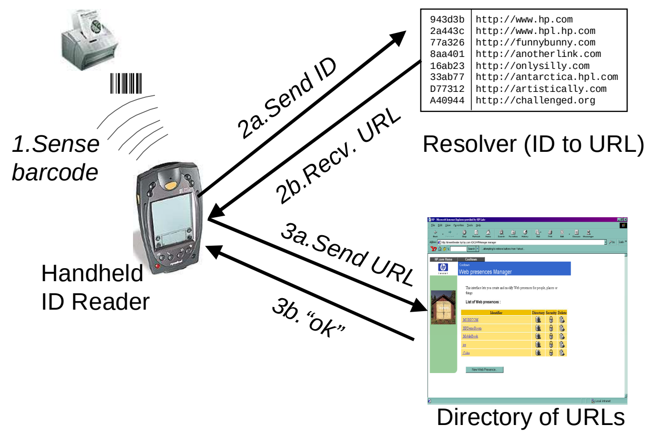

Figure 5. Physical Registration in the device-centric model.

At this point the device has a URL for the resource to be entered to the Place directory. Next the device application sends the resource URL, again as an HTTP GET request to the web server hosting the directory (step 3a); the directory replies with success or failure (step 3b) to complete directory update. In our system the directory is the Web Presence Manager described in [9].

Our prototype uses the Symbol-1740 PDA running Palm-OS, equipped with a laser barcode scanner, a Wireless NIC (802.11) and an IR port. The device is able to communicate over the wireless interface using the TCP/IP protocol through a Spectrum24 Access Point[15]. Our implementation of the device-centric approach is built upon the existing E-squirt [16] implementation used in CoolTown[17]. E-squirt is capable of receiving the URLs of entities' web presences from beacons over infrared. The URLs are stored and displayed as a list, as shown in Fig 6.

We added extra functionality into the E-squirt application to maintain a place context (seen as the "UR in EUREKA" notice in Fig. 6), to detect barcode scan events, and send the barcoded identifier across to the resolver over the wireless interface. The resolver responds with a XML string that encodes the URL corresponding to the identifier encoded in the barcode. This information is interpreted, saved in the list of URLs found and displayed to the user with highlight to indicate that it is pre-selected. The user (registrar) can then press the "AddTo" button to submit the URL to the PlaceManager as an added entity in the place.

The device-centered approach requires two addresses in the device, one corresponding to the resolver and one corresponding to the target directory. These addresses can be loaded into the device in a variety of ways. Three examples include manual entry by the registrar, beacon receive, and tag resolution. Manual entry is tedious and it requires an out of band, authoritative information source well known to the registrar.

In the second approach, the registrar places the infrared receiver of the handheld device in front of a beacon that corresponds to the directory to be updated. This approach would be a natural one if users of the directory for the room typically used a beacon to discover that directory. The only addition needed to allow physical registration would be for the text emitted by the beacon to include an address for the resolver as well as the directory (PlaceManager) address. The in-place nature of this solution avoids the uncertainty of finding the addresses by some out-of-band means.

The third solution is even more self-consistent in that the same id technology is used for place selection and for entity selection. To get the place address, the registrar scans its tag and resolves it to obtain the URL for the place. If tags alone are to be used for physical registration, some initial resolver is required to convert the place tag identifier into a URL for the place directory. This resolver could be configured into the handheld device manually, by multicast discovery on the wireless network, or by a beacon.

4.2 Infrastructure-based Physical Registration

Our second solution to physical registration places more of the physical registration logic in the service infrastructure and less in the handheld device. The goal is to make the handheld device software a minor extension to a web browser.

To achieve this we create a service for registration that knows, or can be given, the directory to be updated and the resolver for identifiers. This service could be, for example, a component of the PlaceManager[8,9].

We describe this solution in two parts. First we load the handheld device with the web page for the directory-update service. This is illustrated in Figure 7. The directory to be updated is selected by sensing a URL from beacon (step 1, Fig. 7), then an “Edit” button on the corresponding web page obtained by the handheld device (Fig. 7, step 2) is used to connect to the place configuration (edit) service. The edit service returns a web page that supports entity tag resolution and directory update. This page is a web form with a single field that takes an identifier; that identifier is sensed by the handheld device and automatically filled into the form, rather than entered from a keyboard.

With the web form for the service loaded in the handheld we can proceed with second part of the process as shown in Fig. 8. The first step, identifier sensing (Fig. 8, step 1), is the same as for the device-centric solution. However, the resolution (Fig. 8, step 3) and entity entry (Fig. 8, step 4) steps are handled by the registration service acting on behalf of the handheld device. Thus the handheld sends the identifier to the registration service (step 2, Fig. 8) and receives a reply from the service after the entity is registered (step 5, Fig. 8). This is a simplification of our actual implementation: if the resolver knows several web presences for the same physical object then the user is invited to choose one of them.

Filling out the web form at the device involves feeding the sensed identifier automatically into the appropriate field of the form and automatically posting it, so that the user’s only actions are to sense identifiers and check the results that appear as web pages. Our current implementations approximate this type of “physical” form-filling using URL manipulations. One implementation runs on the Symbol 1740 devices, running the EudoraWeb browser and a web form plug-in that we developed for that browser. We have also developed a Windows CE version, based on an in-house browser. In each case, the software needed on top of the basic browser is quite simple. Further details can be

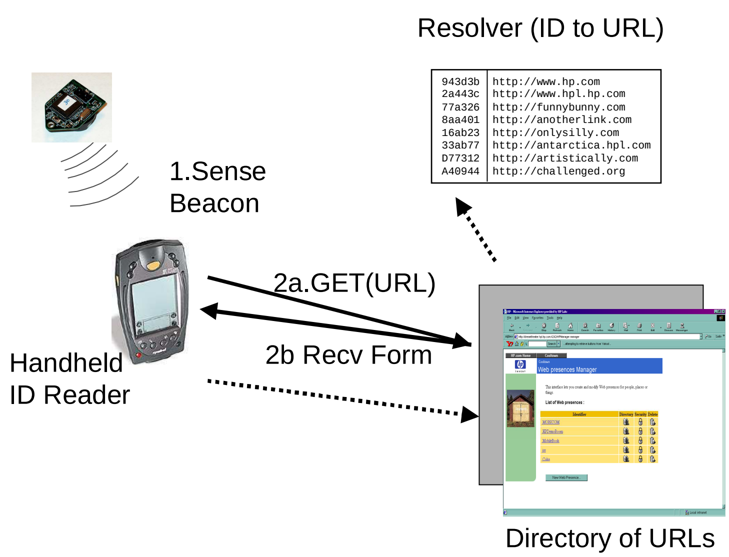

Figure 7. Resolver and directory selection for infrastructure-based physical registration

found in [14].

The association between the directory service and the resolver used for obtaining web presences can be accomplished in two ways. First, the directory service may be permanently configured to use a standard resolver.

For many applications in place management, this will be adequate: the hosting environment for place management would simply have a resolver that lists all entities in that environment. Each place would have links to some subset of these entities. For example, a campus might have a resolver for all entities owned by the university. A classroom would have links to only some of these entities.

Other uses for physical registration might not have such an obvious connection between the directory and a single resolver. For example, your own personal directory might include links to entities at work and at home or links to entities you found interesting while traveling. Our implementation allows the user to specify the resolver in one of two ways, both based on using a secondary web form that appears as a link on the directory editing service's web page. First, as in the device-centric solution, the user may pick up a resolver's URL from a beacon, but this time while their browser is set to the directory service's resolver-selection form. Second, the user may scan a barcode into that same form. Users may keep barcodes for each of the resolvers that they like to use, on a sheet of paper. They scan a barcode to specify the resolver to use. Of course, that assumes that the resolvers themselves are all listed in a root resolver shared between the user and the directory service.

4.5 Comparing the Physical Registration Solutions.

The device-centric solution was our first implementation. It has the advantage that the selection of a resolver and the selection of a directory editing service may be completely decoupled. However, this approach requires the development of PDA software specific to physical registration. The development of software on handheld devices is laborious; its installation, maintenance and upgrade becomes a systemic barrier to deployment.

The infrastructure solution seeks to avoid development of application specific PDA code. By reducing the role of the handheld device to web-client (browsing and form-filling) operations, we have the potential for these web-client operations to be used by many applications. This in turn increases the value of the web-client software, lowering the barrier to system deployment. The infrastructure solution also reduces network traffic between the handheld and the infrastructure services. Since wireless communications is expensive and less reliable, this reduction is advantageous. The primary disadvantage of the infrastructure solution is the configuration coupling it creates between the resolver and directory services.

5 Generalizations

Our implementations were adequate to demonstrate the concept of physical registration. Practical deployment may require different approaches. Therefore we dissect the concept in to its components and the actions that connect them as a way of introducing some of the alternatives. We also discuss some broader issues for physical registration.

5.1 Physical Registration Components

Our description of physical registration uses six components:

Tags. Tags are auxiliary physical objects attached to, within, or near the entity we wish to register for later discovery. A tag encodes an identifier. We mainly use barcodes for our tags but alternatives are passive or active radio frequency tags[18], active contact tags [19], and text read by scanners [20,21].

Identifiers. Identifiers, encoded in tags, provide a way for us to map a physical object into a web presence, whose URL is registered in a directory. We use legacy identifiers such as UPC codes on products and books but we embed those into the Uniform Resource Identifier (URI) namespace. In fact, all identifiers are, or are converted to, URIs in our system. URIs are infinitely extensible, human-tractable strings that can be made unique over space and time.

Handheld Device This component senses barcodes, iButtons or other physical manifestations of the identifier, and converts them into a string of characters. The device also acts as the registrar's user interface for lookup, directory selection, and data-entry. In our prototype we sense barcodes for object identifiers. Barcode scanners are increasingly found integrated into PDAs [15]. Many objects are manufactured with barcodes; where they are not, barcodes can be easily printed and attached to objects. When the user scans a tag such as a barcode, an application on the handheld device handles the sensing event. In response to the event, the handheld device sends messages over the network. The messages ultimately cause the directory to be modified and the entries to be added or deleted. Note that we presume that the device has wireless connectivity to infrastructure services. Such devices are uncommon in 2001 but we developed physical registration for futures systems when this connectivity will be common.

Resolver. This component responds to "lookup" requests containing identifiers. In our system the result is a URL. This URL is then made available to the browser or any other application that can access the services of the resource through the web.

Address. These are the values to be registered (loaded into) the directory to represent the association of the entity with the tag and the entity represented by the directory. We use URLs for addresses.

Directory. The directory is the end-point for registration. It acts as a sink for address registration requests that originate when the registrar selects an entity with the identifier sensor on a handheld device.

5.2 Key Actions in Physical Registration.

Physical registration uses the six components above in five actions:

Identifier Sensing. By identifier sensing we mean the act of the registrar to select an entity by manually approaching the entity and operating a sensor to detect an identifier. For example, the registrar performs identifier

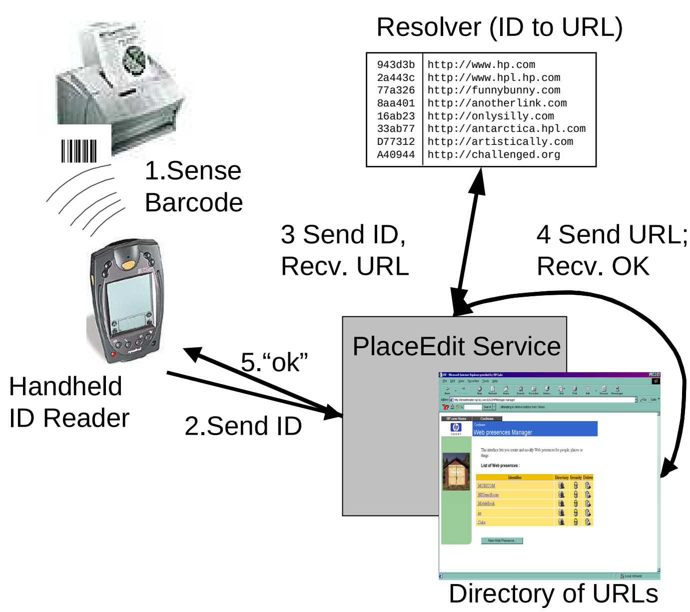

Figure 8. Physical Registration using Infrastructure Services.

sensing in a barcode system by scanning an entity's barcode.

Resolver Selection. Having sensed an identifier we need to convert it to a corresponding URL of a web resource (one of possibly several) of a type that meets our requirements. On scanning a CD's barcode, we may intend a web page about the CD and artist; but we may equally intend an MP3 file of the CD. For this we need to send the identifier to an appropriate resolver. For that we need the address of such a resolver.

Lookup. The lookup action on a resolver converts the identifier into zero or more addresses. In our example, lookup takes a string from a barcode and returns URLs that point to corresponding web presences for the object. Such a URL can be used by a visiting nomadic user to obtain a resource associated with the entity through its tag. The registrar uses the same URLs for physical registration into a directory that that nomadic users can easily discover, without necessarily visiting the object itself.

Directory Selection. Identifier sensing and lookup gives us a pointer to the information we need to record. Directory selection gives us a location to record the information. We might, for example, scan a bar code placed on the wall of a conference room and lookup the URL of its corresponding "place manager" directory service to select the destination of our registration activities.

Address-entry. The address-entry action takes the address of an entity from the lookup step and records it in the selected directory for later retrieval by nomadic users.

Different approaches to physical registration could pick different technologies for the components leading to slightly different interpretations of the actions.

5.3. System Issues

In addition to studying variations in individual components or their interaction we also need to study issues involving the role of physical registration in supporting nomadic users of a pervasive computing infrastructure. We shall illustrate our points with the example in Section 1 in which Acme Software Inc. sets up a stand at the Wireless Web World exhibition

Hierarchy and context overlap. Just as there may be many objects in their surroundings, there may be several place contexts to choose from: the user is nearby an exhibition stand, but there are other stands nearby; all are within the entire exhibition and, in turn, in a city. The web is a natural medium for presenting information to nomadic users because of its hypertext structure. The web pages for a place can include links to nearby places. For example, the Acme exhibition stand's context may contain links to resources that are not local to the stand itself: for example, to pages on the main Acme web site, and to the context of the entire exhibition.

Bootstrapping. To support nomadic computing, the system must take the user from switching on the device to obtaining a link to (i.e. the URL of) an Acme resource, with no user intervention except selection of which of the available places and resources interests them. Therefore, at least the root context for the exhibition appears directly as a URL that can be accessed once the device has fetched its IP configuration from a DHCP server. All other contexts (e.g. the Acme stand), and hence resources, can be discovered automatically from that root.

Scope. We re-emphasize that users are not forced to use the local place context when discovering the web resources associated with an entity found in that place. Consider a user who looks up a book's ISBN at the Acme stand: the Acme Software resolver for ISBN might lead to special offers on Acme books, but a user should be able to use the ISBN with another resolver to see how special that offer really is. The user can choose any context in which the book's identifier will be looked up. The point of a place is to provide locally relevant resources, not necessarily all relevant resources.

Interplay of Tag Technology with System Design. In the simplest case Infrared beacons provide URL strings while tags like bar codes provide opaque strings. However, it is also reasonable to have beacons that emit opaque values and tags that encode text for URLs. Resolvers may utilize both URLs and other types of identifier. They act both as a level of indirection and as a naming context: resolvers allow us to change the mapping of entities to URLs by changing the contents of the resolver or the resolver we choose to consult. (See also [14]).

Update logic. In our simple prototypes we unconditionally append to the directory every time a URL is presented. Other choices would have to be explored through the user studies. For example the directory modification could require confirmation, blocks of URL could be added at once, individual URLs could be removable, permissions could be required for update and so forth.

Creating New Web Presence for Entities. Our description of physical registration assumed that the directory and the resolver for entities already existed. We have experimented with instantiation of new web presences from wireless PDAs. When an identifier fails to resolve to a URL, the resolver returns the URL for the web-presence creation service. This service offers a web form to define a new resource as a web presence for the just-identified entity. The form includes entity type selectors as well as text descriptions. Processing the form creates the new resource and binds the previously unbound identifier in the resolver to the new resource's URL. Subsequent resolution of the identifier would lead back to the newly created representation. When applied to places, this entity creation prepares us for physical registration of entities.

Integrated Web Presence. We described physical registration independent of a particular directory technology. With a directory system like the Web Presence Manager [9] that supports registering URLs in a general way we could create for example bi-directional links. Thus we could link an entity to its place directory at the same time that we register an entity as being in a place.

Locating the Web Presence for a Device. A device such as a printer may have a web presence hosted in the infrastructure. For example, it could be controlled from a PC that runs a web server. The registrar, who knows the URL of that service, can bind it into the place's directory. However, as we stated in the introduction, some devices may host their own web presence – their own web server. That server runs at an IP address that is allocated to the device by DHCP[22] when it is switched on. The registrar needs to locate that web presence conveniently. This problem can be solved by having the device register its URL automatically in a 'service directory' (not to be confused with the directory that the registrar is editing) that the registrar can also find. The device registers itself using a unique identifier that the registrar can also sense from a tag accessible outside the device. The registrar (or, rather, the registration service) can then lookup the same identifier in the service directory to obtain the device's dynamically created web presence. The service directory that we refer to here is a registry as used in standard service discovery protocols such as UPNP[13] and SLP[10] – one that can be discovered by multicast on the local subnet. We argued in Section 2 that those systems are inadequate for constructing most types of directory. Here, we use them as a tool in such a way that those inadequacies do not come into play: our only need the registrar to find the web presence in such a system.

5.4 Discussion

Ultimately, user studies are required to understand the true benefits and disadvantages of our approach. However, our position is that physical registration has advantages over the two existing procedures between which it lies: the manual procedure of text entry and the automatic procedure of multicast-based discovery. We conclude by comparing those other techniques with physical registration.

Like text-entry, our approach is manual in the literal sense of using our hands. Rather than referencing entities symbolically in a configuration file as is used in text-entry, physical registration references entities by directly sensing their presence, e.g. by sensing a barcode attached to the entity. The sensed data is transmitted over a network to the directory while the registrar is in the place being configured and near the object being added. The registrar, who is a mobile user, can verify the directory before leaving the place. The registrar's experience is concrete and direct: they are operating a handheld device in the presence of the entities they wish to add to a directory.

Like text-entry, physical registration can be carried out more accurately than the multicast solution. The latter relies on subnet topologies and limited machine-storable resource descriptions that are liable to miss many of the subtleties of usage conventions in a set-up such as the stand in an exhibition hall. With physical registration, the registrar can use their knowledge of the real world to make judgements about their selection of objects and thus web resources.

However, like the multicast solution, we rely on strong network infrastructure to simplify the configuration task. We need existing electronic representations of the entities to be available via URL dereferencing and we need an existing means of mapping entity identifiers into those URLs. The operations in physical registration are manual: it is much more controlled than the multicast registration procedures.

When compared to multicast solutions, physical registration uses similar infrastructure, but it requires direct human action to construct the directory. While these aspects may be regarded as drawbacks in specialized cases, they also distinguish physical registration positively as a human-tractable mechanism for constructing electronic directories in a nomadic computing world.

6. Acknowledgements

We appreciate the special efforts of Philippe Debaty in helping us understand how to use his Web Presence Manager for this project. We also thank John Schettino for help with his software for the PalmOS. Early access to these two components significantly aided our development.

7. References

[1] “Google Search Technology” http://www.google.com/technology/index.html

[2] “LookSmart Help” http://www.looksmart.com/r?l&page=/help/main.html

[3] “What is the Yahoo! directory?” http://help.yahoo.com/help/us/dir/dir-01.html

[4] “About This Site” http://slashdot.org/about.shtml

[5] “Lightweight Directory Access Protocol” http://www.umich.edu/~dirsvcs/ldap/; see also http://www.openldap.org/doc/admin/dbtools.html

[6] People, Places, Things: Web Presence for the Real World, Tim Kindberg, John Barton, Jeff Morgan, Gene Becker, Ilja Bedner, Debbie Caswell, Phillipe Debaty, Gita Gopal, Marcos Frid, Venky Krishnan, Howard Morris, Celine Pering, John Schettino, Bill Serra. Presented at 3rd IEEE Workshop on Mobile Computing Systems and Applications WMCSA 2000 December 7-8, 2000 Monterey Marriott, Monterey, California, USA http://www.cooltown.com/papers/webpres/WebPresence.htm

[7] A Web-Based Nomadic Computing System. Tim Kindberg and John Barton. Computer Networks, 35 (2001) 443-456, Elsevier (http://www.elsevier.com/locate/comnet).

[8] Caswell, D. and Debaty , P. Creating web representations for places. *Proc. Second International Handheld and Ubiquitous Computing Symposium, HUC’2000,* Bristol, England, 2000.

[9] Philippe Debaty and Debbie Caswell. "Uniform Web Presence Architecture for People, Places, and Things", HPL-2000-67, to be published in the IEEE Personal Communications Special Issue on Pervasive Computing

[10] Guttman, E. Service Location Protocol: Automatic Discovery of IP Network Services. *IEEE Internet Computing*, Jul.-Aug. 1999. pp. 71-80.

[11] 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

[12] Jini network technology http://www.sun.com/jini/

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

[14] Ubiquitous and contextual identifier resolution for the real-world wide web (Tim Kindberg). HP Labs Tech. report HPL-2001-95. Submitted to Ubicomp 2001.

[15] “Symbol Wireless Products” http://www.symbol.com/products/wireless/wireless.html

[16] “Key CoolTown Software is Available Free” http://www.hpl.hp.com/news/esquirt.html

[17] CoolTown home page. http://www.cooltown.hp.com/.

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

[19] iButton home page. http://www.ibutton.com/.

[20] Seimans Pocket Reader. Pocket Reader, LLC, 64 Main Street New Canaan, Connecticut 06840 http://www.pocketreader.com/

[21] CPEN, C TechnologiesAB(publ),Ideon Research Park, Scheelevägen 15, SE-223 70 Lund Sweden, http://www.cpen.com/

[22] “Resources for DHCP” http://www.dhcp.org/