Implementing Physical Hyperlinks Using Ubiquitous Identifier Resolution

Tim Kindberg
Mobile Systems and Services Laboratory
HP Laboratories Palo Alto
HPL-2001-95 (R.1)
March 12th , 2002*

E-mail: timothy@hpl.hp.com

physical hyperlinks, identifier resolution, ubiquitous computing, nomadic computing

Identifier resolution is presented as a way to link the physical world with virtual Web resources. In this paradigm, designed to support nomadic users, the user employs a handheld, wirelessly connected, sensor-equipped device to read identifiers associated with physical entities. The identifiers are resolved into virtual resources or actions related to the physical entities - as though the user 'clicked on a physical hyperlink'. We have integrated identifier resolution with the Web so that it can be deployed as ubiquitously as the Web, in the infrastructure and on wirelessly connected handheld devices. We enable users to capture resolution services and applications as Web resources in their local context. We use the Web to invoke resolution services, with a model of 'physical' Web form-filling. We propose a scheme for binding identifiers to resources, to promote services and applications linking the physical and virtual worlds.

Implementing Physical Hyperlinks Using Ubiquitous Identifier Resolution

Tim Kindberg

Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94304, USA
+1 (650) 857 5609
timothy@hpl.hp.com

ABSTRACT

Identifier resolution is presented as a way to link the physical world with virtual Web resources. In this paradigm, designed to support nomadic users, the user employs a handheld, wirelessly connected, sensor-equipped device to read identifiers associated with physical entities. The identifiers are resolved into virtual resources or actions related to the physical entities — as though the user 'clicked on a physical hyperlink'. We have integrated identifier resolution with the Web so that it can be deployed as ubiquitously as the Web, in the infrastructure and on wirelessly connected handheld devices. We enable users to capture resolution services and applications as Web resources in their local context. We use the Web to invoke resolution services, with a model of 'physical' Web form-filling. We propose a scheme for binding identifiers to resources, to promote services and applications linking the physical and virtual worlds.

Categories and Subject Descriptors

H.3.5 [**Online Information Services**]: Web-based services. H.4.3 [**Communications Applications**]: Information browsers. C.2.4 [**Distributed Systems**]: Client-server, distributed applications.

General Terms

Algorithms, Management, Design.

Keywords

Physical hyperlinks, identifier resolution, ubiquitous computing, nomadic computing, mobile computing.

1. INTRODUCTION

This paper describes the design and implementation of a system that provides *physical hyperlinks* from the physical world to virtual resources on the Web. In this paradigm, which is designed to support nomadic users, the user employs a handheld, typically wirelessly connected, sensor-equipped device to read identifiers extracted from, attached to, or near physical entities. Those identifiers undergo *resolution* into Web resources related to the physical entity. Thus physical entities are bound to URLs referenced when the user senses their identifiers with a reader such as a barcode scanner.

Compared to hyperlinks in HTML, physical hyperlinks appear as tagged physical objects instead of marked text or images; the 'mouse' is a reader with which they scan instead of 'clicking'; the result appears on a browser in their hand. We shall speak of 'scanning an entity' and 'clicking on a physical hyperlink' as the same operation, whatever the underlying identification technology.

When the user clicks on a physical hyperlink the Web resource they obtain may be information or a service provided to the client, or an action in the environment. Users can select the type of effect that occurs when clicking on a physical hyperlink. We have built a variety of applications as Web services that induce application-specific hyperlinks onto the physical world. Those applications appear to the user as Web pages on the screen of their mobile device in the normal way. However, the applications' Web pages encompass the induced physical hyperlinks as well as their own conventional hyperlinks. Our examples include the following:

Physical browsing. The user obtains information pages about items that they find and scan. For example, while browsing in a book shop, they are offered 'one-click' purchasing for books when they scan them. While browsing in a supermarket, they find information about the products. If a user finds no entry for a particular object, the system may invite them to add one.

Physical registration. To register (or deregister) entities such as printers and furniture as belonging to a place such as a meeting room — and thus visible in the place's Web pages — an administrator scans them at the place's administration page [2].

Light control. The user scans the 'lights on' barcode placed by the entrance to their work area. That action causes the corresponding lights (which, in our workplace, can be activated from the Web) to be toggled on or off.

Virtual Graffiti. When the user scans an object, they see a bulletin board page associated with it, one that is shared with their user group. Note that these bulletin boards are not, in general, publicly accessible. For example, two families who scan a 'noticeboard' barcode on the wall of the same café will see different, private sets of messages by virtue of scanning them at different Web message board pages.
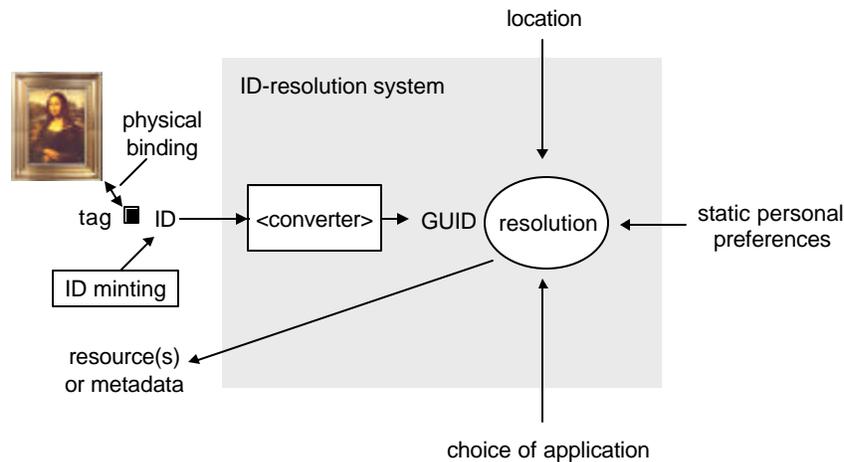
Figure 1. The elements of an ID-resolution system.

My music, your place. The user wishes to hear music they own, which can be accessed on the Web, at a place they are visiting such as a friend's house or a kiosk in an airport. They need to select the piece of music and the device to play it on (their friend's music system or the kiosk). The application invites them to scan the device and to scan the music from a CD case or a booklet of barcodes for all the music they possess.

1.1 Ubiquity

This project's goal is a system for physical hyperlinks that is ubiquitous in its availability and which can be used in a variety of ways by different communities of users in various contexts. Ubiquity means that users should be able to pick up identifiers and, as long as they are connected to a wireless network, have them resolved wherever they happen to be — for example, in the home, the workplace or a museum. Moreover, the desired resolution result may be a function of contextual parameters such as the user's location, their personal preferences (e.g. the types of answer they seek) and the device they are using.

To achieve ubiquity we require a widely available service delivery platform on top of widespread wireless networking. We utilise the Web to deliver identifier resolution services. We do so because of its large base of browser and server implementations on many different devices — wireless and wired, handheld and otherwise — and servers. Moreover, the Web's HTTP and URI standards have proved to be flexible and adaptable to new types of service delivery.

We take ubiquity to include universality with respect to users as well as locations. An important characteristic of resolution is how to determine the choice of Web resource(s) offered to the user when they sense an identifier. For example, consider two shoppers and a supermarket employee who all pick up identical cans of food in the supermarket. Suppose that they all use the physical browsing application we described above, scanning the barcode on the can with a wirelessly connected device to look at Web pages about the food.

The chances are that those three users would want to see different results. One shopper wants to avoid genetically modified foods, and wants to see appropriate links to check the food's status. The other shopper suffers from diabetes, so wants to see diabetic links about the food. The employee wants to do a supermarket price or stock check. All may want to see a link to the supermarket page, in addition to the other specialised pages mentioned.

And if the shoppers were back in their homes instead of the supermarket, then they might want to see yet another set of pages when they scan the can's barcode: for example, a page enabling them to put that item on their Web shopping list.

1.2 Contribution

This paper describes the research issues in developing a system of physical hyperlinks, and a design and implementation that we have developed for the Internet infrastructure and handheld devices. We describe how we have integrated identifier resolution with the Web so that it can be deployed as ubiquitously as the Web, in the infrastructure and on wirelessly connected handheld devices. We put forward a way for users to select, by a combination of sensing and navigation on the Web, a resolution service that is appropriate for them and their context. We use the Web to invoke resolution services, with a new variant of Web form-filling. We propose a scheme for binding identifiers to resources, aimed at promoting many powerful services and applications linking the physical and virtual worlds.

Section 2 identifies the subtasks needed in a system for identifier resolution. Section 3 discusses related work on tagging and resolution. Section 4 describes how we propose to manage the identifiers that are tagged on physical-world objects, and how those identifiers are bound into multiple naming contexts. Section 5 describes the design and implementation of Web-based resolution. Section 6 concludes with a discussion.

2. ID-RESOLUTION SUBTASKS

An identifier (ID)-resolution system (Figure 1) involves the following subtasks:

ID creation. Resource identifiers are created (we shall sometimes refer to this as *minting* identifiers).

Binding. Binding is the activity of associating an identifier and one or more resources. Sometimes binding is physical: the identifier is physically tagged to a physical entity. Sometimes binding is virtual: a table entry is created to map the identifier onto a resource or metadata (not shown in the figure). A binding is data specifying an association between an identifier and a virtual resource or metadata about it, in particular its address.

ID Capture. Identifiers are captured from physical entities. An identifier may be derived from the entity's image, it may be in the form of a tag attached to it or near the entity, or it may be an extrinsic individuating factor such as its position.

Conversion. Raw identifiers may require conversion into a unique or canonical form for processing in the system. Uniqueness refers to space and time. We shall refer to a unique identifier in this document as a GUID (globally unique identifier).

Resolution. This is the processing of the GUID and relevant contextual factors to produce bindings or the bound resources, which are sent to the client. Resolution is application-specific. It may involve taking into account parameters such as the user's current location and their static personal preferences.

Thus identifiers are minted, captured and converted; and they are bound to resources. Bindings are created and looked up. Resolution is the process of looking up bindings from identifiers and returning the bindings or the resources to which they refer.

3. RELATED WORK

This work is part of the Cooltown project [1, 11, 12], which creates physical hyperlinks between the physical and virtual worlds so as to form the 'Real-world Wide Web': an integration of the Web with physical entities. It utilises the same HTTP and URI standards as the conventional Web.

In the original deployment, Cooltown users sense URLs from infrared 'beacons' placed on or near objects such as a printer or a painting. The URL is that of the Web page — the 'Web presence' — of the corresponding entity. Nomadic users can thus view or bookmark pages about the places, things and even people they encounter. The service whose URL is emitted by a beacon may provide personalised content; but the user cannot choose the service for a given entity.

Here, we extend the connection between the physical and Web worlds to include any type of identifier that can be sensed with handheld devices. The identifier could be in many forms, including a barcode, an RFID tag, an iButton, an infrared beacon, or the coordinates of the entity that interests them (e.g. the WebSign system uses a 3D pointing device to designate the object [18]). The choice of identifier technology has an impact on deployment due to costs and physical constraints. But this paper will concentrate on the choice of identifier encoded within that technology.

Several other projects have investigated tagging (identifier and sensing) technologies [22] and identifier resolution systems. Applications include information services [8, 20, 21], leaving virtual notes on physical objects [9], and content-transfer [13]. But all the projects of which we are aware have either produced a single application or a closed system that can be configured for different applications but not in an extensible way. None addresses how a given identifier could yield different results in different contexts.

In the resolution scheme proposed by Mealling [15] for Uniform Resource Names (URNs [17]) and other identifiers, a URN u is iteratively re-written and looked up in the Domain Name System (DNS) to compute a service to resolve that URN, $R(u)$; then u is resolved by R. The Object Naming System [4] is simpler in that it transforms an 'electronic product code' directly into the domain name of the corresponding service needed to obtain a resource for that product. The handle system [7] for Digital Object Identifiers (DOIs) [6] uses a two-tier scheme: the name-authority prefix within a DOI is mapped to a service to resolve it.

These resolution schemes require additional Internet infrastructure or use existing infrastructure in complex ways. Moreover, they all provide only a 'default' or 'well known' binding for a given identifier: one that can be located starting from knowledge of only the identifier itself. That is a useful facility in many circumstances. But no matter who presents an identifier to the system or where they present it, they always obtain the same answer, contrary to our approach.

4. IDENTIFIERS AND BINDINGS

A system of physical hyperlinks depends on identifiers and bindings that satisfy certain requirements, if it is to be widely accepted.

4.1 Identifier Requirements

Identifiers have several requirements associated with them:

Uniqueness. The ability to mint identifiers uniquely over space and time is valuable because individuals and organisations can create identifiers and share them without conflict.

Inexhaustible supply. Identifiers should be practically inexhaustible, so that we may label every conceivable entity of interest to humans or software.

Legacy identifiers. The ID-resolution system should operate with legacy identifiers such as ISBNs, ISSNs, UPC codes, EAN codes, iButton identifiers, MAC addresses, etc. Many of those are already attached to everyday items.

Human tractability. It is valuable for identifiers to be convenient for humans to read, type, etc. The value comes from enabling humans to find ways around errors (e.g., if a barcode does not scan); from enabling humans to communicate about the names they are using; and from enabling them to include information useful for creating names, such as attributes.

Convenience of minting identifiers. If individuals and small organisations such as shops are to be able to participate by attaching identifiers to their entities — not just classes of entities such as product types but individual entities — the cost of minting (creating) new GUIDs should be negligible. There should be zero or negligible registration cost (c.f. domain name registration). No participant, big or small, should have to communicate with others to create identifiers guaranteed to be unique.

4.2 Identifiers: Our Approach

We use Uniform Resource Identifiers (URIs) as GUIDs. URIs [3] include URNs and URLs. URIs are, in general, variable-length strings intended for use by humans as well as software; they are infinitely extensible.

Several existing classes of URI can be minted in such a way as to be unique over space and time. URNs employ a variety of domain-specific registration schemes. UUIDs [14] use names assigned uniquely at any given time such as IP numbers, together with timestamps and random numbers. On the other hand, URLs are globally unique over space but not necessarily over time: two principals that are assigned the same authority (e.g. domain) name at different times might 'mint' the same URL and use it to identify different resources. Another disadvantage is that URLs that are intended as pure identifiers are liable to be used (erroneously) as locators.

Thus, we have proposed a new type of URI, a *tag* [10]. Tags are minted uniquely over space and time in a decentralised way but, unlike UUIDs, they are tractable to humans. They can be minted at no cost by anyone who already holds the registration of a domain name, and even by anyone who possesses an email address. This is achieved by date-stamping an email address or domain name and using that as a prefix.

We leverage the evolving URI framework for legacy identifiers. For example, when a device reads an ISBN on a book, that identifier is converted to a canonical URN form before look-up.

4.3 Binding: an Analysis

Two types of binding are very familiar: (1) the physical binding of an identifier to a product and (2) the binding of names to IP numbers in DNS. In each case, minting and binding authorities coincide.

For example, 'Acme Beanz' minted the UPC for their cans of beans, and they bind the UPC to their product physically and in their product catalogues. They alone control these bindings. Similarly, the holders of the registration for *champignon.net* minted the name *www.champignon.net*, and they assert the binding from that name to 209.157.129.132. They control the binding, which is in their DNS zone file.

The resolution schemes we described in Section 3, such as those for URNs, are also chiefly concerned with looking up precisely the minting authority's binding: the mapping from the GUID to the resource specified by the authority that minted it. The

resolution service for looking up the binding can be determined from the name itself.

However, consider again the example of the supermarket shoppers (Section 1.1). One identifier, the UPC on the can — let's call it upca:78996800002 — has bindings in multiple *naming contexts*: collections of bindings between names and resources or metadata [5]. (This use of 'context' is related to, but not to be confused with the user's context, e.g. location and preferences.) One binding of the can's identifier is in the naming context that stores resources related to genetically modified food; another in the naming context that stores resources related to diabetes; one in the supermarket and one in a shopper's household.

Our view is that none of those bindings is actually less 'authoritative' than the minting authority's binding; they simply derive from different authorities. Bindings derive from individuals and organisations that assert them (Safeway stores, the Genetically Modified Food Information Council, the Finnish Diabetes society, Tim Kindberg's household). Those organisations can digitally sign them to make them literally authoritative.

What about the fact that the name contains 'upca' — does that signify some 'extra' authority for the manufacturer? Our answer is that the manufacturer has certain binding privileges but not exclusive binding authority. There's a basic distinction between the authority to mint identifiers and the authority to bind them to resources, which rarely seems to be made (perhaps because of the prevalence of the DNS model). The UPC Council has devised a mechanism for ensuring that manufacturers can mint identifiers for their products without fear of collision. Those manufacturers then bind those identifiers (a) physically as part of the fabric of the product and (b) virtually, to virtual resources about the products.

Should that mean that the 'Genetically Modified Food Information Council' cannot also bind the UPC code to their own virtual resources, or that, if they do so, their binding has a lesser status? One can understand why Acme Beanz might wish that virtual binding didn't exist or was deprecated, for commercial reasons. But there is no logical objection that they can raise; at least, not as long as we satisfy the requirement that no-one will reasonably be confused about whose binding is whose.

4.4 Binding: Our Approach

Our model is predicated on a decentralised plurality of naming contexts and name spaces, like the Spring naming system [19]. We routinely live with multiple naming contexts for the same set of identifiers. Take any two PCs. Each resolves names such as */usr/bin/perl*. But the answer we get if we present that name to the two PCs may be different (two different implementations of perl). We do not get confused as long as we are clear about the difference between the naming contexts. Similarly, if the user presents an ISBN to *amazon.com* then they expect a different result from that which they would obtain from *barnesandnoble.com*.

In our model, principals (individuals, organisations, communities) mint identifiers and bind them to physical or virtual resources. Other principals, also, may bind the same identifiers to the same

1. User clicks link to select a resolution service

context:http://glim.net/resolve?id=_SLOT_&op=l2Ls

plug-in

2. Browser invokes plug-in, which hands URL to sensor module & browser

http://glim.net/resolve?id = _SLOT_&op=l2Ls

resolve!

4. Sensor module hands *http://glim.net/resolve?id= 9780201619188 &op=l2Ls* to browser

sensor module

3. User scans barcode *9780201619188*
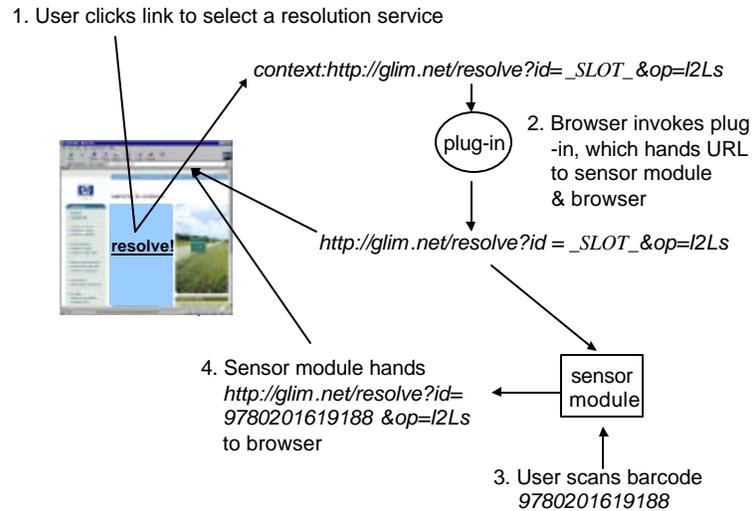
Figure 2. Prototype resolution client (runs on hand-held device).

or different resources. The steps in our minting and binding model are as follows:

1. A principal mints a new globally unique identifier (URI).

2. That principal creates a 'labelling binding' of that identifier onto some physical or virtual resource that belongs to them. For example, a museum attaches (binds) tags to its exhibits; an author allocates an identifier to her document and inserts the bar-coded identifier into the document.

3. They and other principals now bind the same identifiers into whatever naming contexts they like. For example, the exhibit identifier could be bound to entries in the museum guide and also to comments about the exhibits maintained by the students of Gordonbrock Primary School. The document identifier could be bound in a directory of citations and also a directory of critical reviews.

In our model, bindings are first-class objects expressed in XML. Such an 'Xbinding' object is based on Xlink [24] and consists of:

1. A 'URI' attribute — the identifier (URI) that is being bound.

2. An *xlink:href* attribute — the URI (usually, but not necessarily, a URL) of the resource that is bound to item 1.

3. An *xlink:title* — a textual description of the binding.

4. A set of keywords that describe the entity (those can be used in look-ups from Google or other search engines).

5. The textual name and the URL of the home page of the principal that asserts the binding from item 1 to item 2.

6. A digital signature, by the principal identified in item 5, affirming items 1-5.

5. WEB-BASED RESOLUTION

This section deals with how captured identifiers are resolved to bindings or to the resources addressed by URLs in bindings.

5.1 Virtual and Physical Navigation

As we explained in Section 1, in our paradigm certain Web sites (pages) 'induce' or 'include' physical hyperlinks at entities with identifiers. Any Web site can induce its own physical hyperlinks independently of the others, just as it can have its own conventional hyperlinks.

Re-phrasing that in terms of resolution, different Web sites may employ the same entity in different hyperlinks by resolving its identifier independently. Users navigate on the Web to select a resolution service (a naming context) that provides their desired application. They are equipped with a resolution client that is a hybrid of a Web browser and a sensor, implemented on their handheld device. To select a resolution service, the user navigates to a Web page that gives them the resolution service they require — much as they would navigate conventionally to a Web site that gives them a service for travel, say, or books. In our case, they navigate to a resolution service using bookmarks, physical hyperlinks in their environment, or conventional hyperlinks.

Once the resolving Web site has been selected, resolution takes place by a new type of Web form-filling. Instead of filling out information to identify the object (e.g. a book's author and title, or ISBN) using a mouse and keyboard or stylus, users employ their handheld device to sense the identifier of the entity. For example, they scan the barcode of a book. The sensed identifier is automatically filled into the page's Web form, the form is automatically posted, and the page that that resolution service provides for that object is returned to them. Thus the only physical action needed to obtain the result for a given physical object using a given resolution service is identifier-sensing; by default, no other manipulations of the device are required.
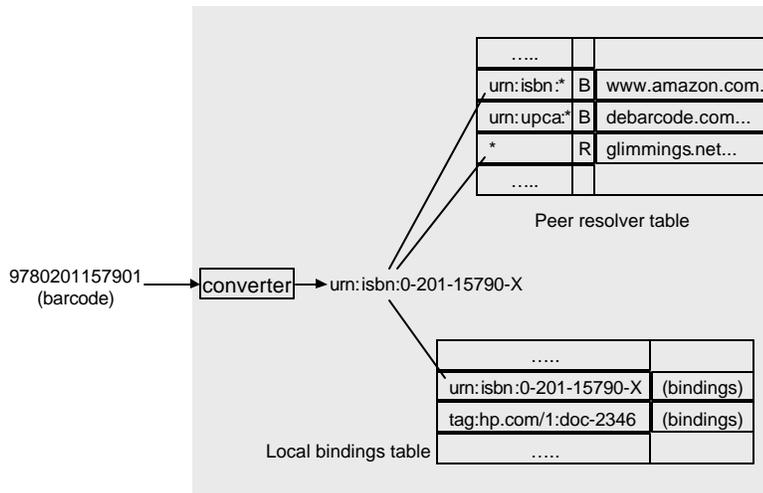
Figure 3. CoolTown resolver components showing entries for a given identifier.

The following are further examples of this Web paradigm as it might be spoken of by users. As in Section 1 we use the word 'scan' intended in the generic (as opposed to barcode-specific) sense of identifier-sensing:

"How do I report a broken printer?" "Go to the 'maintenance' page under *internal.hpl.hp.com* and scan the printer." (On sensing the printer's identifier, the user sees the service history of the particular printer, and can report a fault or see that the fault has been reported.)

"How can I get information in Spanish in the gallery?" "Go to the 'Information in Spanish' page in the eGuide and scan any painting you're interested in." (The user sees pages about the individual paintings.)

"How shall I leave a message for you?" "Scan the café at the family's Web message page." (The user sees their family's postings as though they were left on a notice-board at the café.)

"The nurse scanned my medicine bottle to find the notes that the doctor made when he prescribed the medication." (Clinicians and pharmacists 'attach' their records to medicines and communicate to one another by reading their barcodes in a shared Web context.)

The (imagined) users' dialogue about this system does not contain the word 'identifier': it is the objects themselves that interest them. However, these scenarios are made possible by the existence of identifying tags by the paintings, on the printer, on the walls of the café and on the medicine bottle — or by the ability to sense the location of the (fixed) object using a 3D pointing device. The scenarios assume conventions that the user has to understand such as what tags look like and where they can be found.

The users in these scenarios are 'nomadic'. In general, they scan objects as they find them, not while sitting at a PC. Their client enables them to navigate using conventional and physical hyperlinks to choose services and applications (frequently-used pages would be bookmarked). It also allows them to pick up new applications and services in the places they visit. They can pick up a Web site for their location from an infrared beacon. Inside the place's Web site they can find pages giving local services for resolving identifiers of local objects (i.e. making those objects physical hyperlinks). As an alternative to beacons, places can put up 'you are here' identifiers (e.g. barcodes), for resolution at a well known Web site so as to yield the same set of pages about the place as they would have obtained from a beacon.

The Web resolution paradigm has the advantage for users that the mechanism for selecting the desired service is familiar: the Web. By selecting a Web site for resolution, they specify their application, much as they might have chosen an application such as a spreadsheet or word processor on a PC desktop. The Web forms supplied by resolution services may also enable them to pick up other contextual parameters. For example, they could scan the place they are in to give a location-specific result; they could even scan a personal profile from a list of barcoded identifiers that they carry with them on paper.

5.2 A Sensor-Enhanced Browser

We have implemented a resolution client for the Symbol 1740 PalmOS-based device, which has an integrated barcode scanner and wireless connectivity, and runs the EudoraWeb browser. We have also implemented a client for Windows CE which works with an attached iButton reader on an HP Jornada 680 and on a Hitachi ePlate with a compact-flash barcode scanner, each with a PC card for 802.11b connection. We can implement a client for any handheld device running Windows CE that has a slot for an 802.11b networking card and another port that allows a sensor such as a barcode scanner to be attached. The Windows CE clients are built from an in-house Web browser that supports plug-ins.

In building our prototype resolution client, we avoided adapting the browser wherever we could, for pragmatic reasons. The prototype is thus an approximation to the eventual integration with browsers that we envisage. Our client implementations use a browser, a plug-in and a sensing module (see Figure 2). The combination works as follows.

A Web page at which users can scan entities has a URI with the prefix 'context:' followed by a conventional URL for the Web service itself; for example:

context:http://glim.net/resolve?uri=_SLOT_&op=I2Ls.

When the user clicks on a link containing a *context* URI, the plug-in handles that URI:

1. It strips off the URL *u* from *context:u*.

2. It directs the browser to the URL *u*, so that the corresponding page (describing the particular resolution service) is fetched and displayed to the user.

3. It directs the sensing module to use the URL *u* for resolution.

When the sensing module reads an identifier, it 'fills in the Web form':

1. It locates the string '_SLOT_' within the resolution URL, configured by step (3) above, and replaces it with the sensed identifier to obtain a URL *u'*.

2. It directs the browser to the URL *u'*, so that the resolution result page is fetched and displayed to the user.

What has effectively happened is that a form with one slot ('uri=_SLOT_') has been retrieved from the Web and filled in with the sensed identifier; and the result has been returned to the user. In this approximation, no actual form (in the sense that we understand it from HTML or the Xforms work [23]) has appeared or been filled in. But we have generated the URL that would be produced by filling an identifier into a form with one slot, and perhaps some hidden fields, and submitting it.

A true resolution client would be a browser that accepted a new type of form with mark-up text describing slots that can be 'physically filled in' by attached sensors capable of producing values of specified types. It would be possible to have several such slots within a single page and to allow those slots to be filled in by any of a variety of sensors — or by a human with a keyboard or stylus. We are working on a definition of forms as XML entities based on Xforms through which that can be realised, as well as implementations of a sensor-enhanced Web client that can fill in such forms, and services that can supply and process them.

In the meantime, the system we have implemented has proved to be quite powerful. Many options can be built around our approximations of one-slot forms. What would otherwise have been an N-slot form is turned into a chain of 1-slot forms. N is typically no more than 2 or 3. For example, a resolution service can begin by asking 'Where are you?', at which point the user scans the place at a barcode or beacon. Then the service sends a second form for scanning objects in the local context — a form that may be used repeatedly with different entities in that environment.

5.3 Resolvers

Resolution clients fill in and submit Web forms to Web resources called *resolvers* that implement ID-resolution. Anyone may set up a resolver, without registering themselves with any ID-resolution governing body or entering into agreements with others. A user with any resolution client can take advantage of the resolver's services.

From the outside, a resolver is no different from any other Web page or site that accepts input from forms (through a CGI interface). The resolver has a URL. It provides one or more Web pages so that humans can understand the application or service that it provides. Equally, it may be invoked without human intervention, from any HTTP client.

Although resolvers could be implemented using software produced *ad hoc*, this project set about constructing a resolution component, a Cooltown resolver, that generalises to a variety of applications and services. We have built the examples of Section 1 with it.

5.4 Cooltown Resolvers

The following requirements for resolvers emerged from constructing our applications:

1. The ability to maintain a local collection of URI bindings.

2. A relationship with a resource manager.

3. The ability to use the results of other resolvers.

4. Low computational and network load on handheld clients.

5. Scalability: a means of partitioning the resolution process between servers.

Cooltown resolvers (Figure 3), designed to meet requirements 1-5, have the following functionality. In the description, 'resolver' means a Cooltown resolver, unless we state otherwise.

ID conversion. Resolvers take a variety of legacy identifiers (UPC, ISBN, etc.) and convert them to canonical URIs before looking them up. Conversion happens inside the resolver, not the client, to avoid having to update clients as new standards emerge. (Alternatively, conversion could take place at dedicated services at the expense of an extra round-trip.) An outstanding issue is agreement on canonical URI forms for legacy identifiers, and on the heuristics for conversion. The heuristics need to be integrated into existing resolvers as they become available, implying a conversion 'plug-in' architecture.

Resolution services. Resolvers provide the operations specified for URI resolution by Mealling and Daniel [16]: I2L, I2R, I2Ls — where 'I' stands for identifier, '2' for 'to', 'L' for (default) 'link', 'R' for resource and 'Ls' for 'all links'. The I2L and I2Ls operations are provided so that the user (or a software client) can inspect the binding or bindings before deciding to access a resource whose URL is bound to the given identifier. The 'links' referred to are hyperlinks that the resolver returns in HTML form. (The resolver's current implementation also includes bindings as instances of the Xbinding schema inside a comment in the returned page, so that software that requires bindings rather than hyperlinks can 'scrape' the bindings out.)

Managing the collection of bindings. In general, a resolver needs to maintain its own collection of bindings and provide operations to

add, edit and delete them. Cooltown resolvers provide those operations. They can manage more than one binding for a given URI but they may be configured to maintain at most one. One binding is specified as the default binding for that URI (for an I2L or I2R operation).

Relationship with a resource manager. A resolver that does not currently have a binding for a given URI can offer the user a chance to create one. That may be a new binding to an existing resource, such as stored music. But it is sometimes appropriate to create a new resource at that point; for example, a new Virtual Graffiti bulletin board or a consumer's report on a food product. The Cooltown resolver hands off to a resource manager (in the case of Virtual Graffiti, a bulletin board service) through which the user accesses an existing resource or creates a new one, and the resolver binds the result to the URI.

Relationships with other resolvers. The user may require bindings from various sources when, for example, scanning a book in a bookshop while physically browsing. Those sources may be resolvers other than the one at which the user is scanning: either existing Web resolution services such as *amazon.com* or *isbn.nu* or *google.com* (which is supplied with the identifier or the keywords in the Xbinding), or other Cooltown resolvers, e.g., a local one. To support use of other resolvers in a structured way, Cooltown resolvers can be configured with a set of rules for handling URIs, so that one or more other resolvers are chosen through a regular expression match against the URI. For example, there may be a rule to match ISBNs, which produces the URL that will return the page for that book at *acmeBooksellers.com*. Figure 3 shows a table of resolvers that are peers of the resolver shown. Entries marked 'B' map a URI onto a binding that the resolver may return directly, without consulting the peer resolver — for example, the binding of an ISBN onto the page maintained by *acmeBooksellers.com*. Those marked 'R' map a URI onto the URL of a peer resolver, which the current resolver consults to find a list of bindings that that peer currently maintains for the URI. The resolver returns Xbinding objects as we defined them in Section 4, so the ultimate provenance of any particular binding is, or can be made, explicit.

Iterative and recursive operation, and the load on the handheld device. The protocol used between a client and a resolver for the I2R operation can either be iterative or recursive. In the iterative case, the resolver returns an HTTP 302 'relocation' response with the URL bound to the URI and the client then fetches the resource. In the recursive case, the resolver accesses the resource and sends the resultant content as the return value of the client's request.

Iteration is appropriate if the resource or the client is to be authenticated. But, in circumstances where no authentication is required, a resolver could act iteratively anyway, to save itself the workload of recursively fetching the resource. However, there may be a need to limit the computational and network load on the client (requirement 4). This became especially apparent with a client on the Symbol 1740 device, even though it uses a wireless network nominally rated at 2 Mbit/sec. That device incurs a significant latency on each HTTP interaction. We were forced to use recursive interaction wherever possible, despite the load on the resolver.

Multiple servers per resolution service. Our experiments have been small in scale so far but we expect the last requirement, scalability, to become significant eventually. We provide a mechanism whereby a single resolution service can be implemented at multiple servers, each of which maintains some portion of the bindings collection. The collection is physically split according to regular expression matches against URIs — for example, according to URI prefixes. If a URI in a request matches such an expression, the server that handles the request looks up the corresponding URL of a peer server and sends that URL back in an HTTP relocation response, to redirect the client. Unfortunately, this strategy to make resolution scalable tends to increase the load on clients.

6. DISCUSSION

This paper has described ubiquitous identifier resolution as a means of providing physical hyperlinks: links from physical entities to virtual Web resources. We described a model for separating concerns between minting identifiers and binding them, to allow many principals to assign virtual resources independently to identified entities. We argued that the widespread deployment of the Web and the flexibility of the Web's HTTP and URI standards make it a strong choice as a service delivery platform for resolution. We described a 'sensor-enhanced Web browser' client, using which the user can sense and navigate on the Web to any of a multiplicity of resolution services. That client uses a new, sensor-based Web form-filling model to access each resolution service.

Our approach poses several outstanding research issues in human factors and at the system level. It remains to evaluate the usability of the Web-based resolution paradigm: the cognitive load it presents and its efficacy for various activities. One issue is the choice of conventions by which users recognise physical entities of many different types as physical hyperlinks (the equivalent of underlining hypertext links). What types of identifying technology work best in this respect? Another issue is that the paradigm allows for many resolution services but at the expense of potential ambiguity. The user has to answer the question 'What type of result would I like?' and thus select a resolver. For frequently used resolution services, we expect the cognitive load to be relatively low, but we have yet to measure it in a variety of circumstances. An additional issue is input. We are investigating applications such as Virtual Graffiti in which users can add information themselves when they find a broken link or a link to editable information.

Some applications, such as the 'My music, your place' example in Section 1, involve two or more resolution services. Composing resolution services remains a research issue, not just for the user interface but also for the system architecture. Managing the potential ambiguities that arise from a wide variety of identification technologies and namespaces is another issue. For

example, a user may need to distinguish between an object's class identifier (a UPC barcode, say) and instance identifier (another barcode).

Although we put the user in charge of the choice of resolution service, the Web-based paradigm still allows service providers to aggregate resolution services and provide them selectively to users based upon automatic context capture. Thus *acmeResolution.com* could conceivably provide the user automatically with food-related resources according to their personal profile when they are in the supermarket, and the local museum's pages when they are in the British Museum, relieving the user from having to navigate to any other resolution service. We suspect that both automatic and user-controlled selection will be appropriate but in different circumstances.

In the belief that some form of Web-based paradigm will prevail as a mechanism for linking the physical and virtual worlds, we are developing proposals for standards. As stated above, we have proposed a 'tag' URI standard for convenient identifier minting. We believe that a standard counterpart to our Xbindings is required for first-class binding objects. And we are defining prototype standards for sensor input to Web forms — a mechanism which, we believe, goes beyond identifier resolution in its applications. 'Glimmer', a sensor-enhanced Web client that retrieves forms and fills them in with sensed values, is under construction.

7. ACKNOWLEDGMENTS

8. REFERENCES

[1] Barton, J., and Kindberg, T. The Challenges and Opportunities of Integrating the Physical World and Networked Systems. HPL Technical report HPL-2001-18, 2001.

[2] Barton, J., Kindberg, T., and Sadalgi, S. Physical Registration: Configuring Electronic Directories using Handheld Devices. IEEE Wireless Communications, Vol. 9, No. 1, Feb. 2002.

[3] Berners-Lee, T., Fielding, R., and Masinter, L. RFC2396: Uniform Resource Identifiers (URI): Generic Syntax, 1999.

[4] Brock, D.L. The Electronic Product Code (EPC). A Naming Scheme for Physical Objects. MIT Auto-ID Center white paper WH002, Jan. 2001.

[5] Coulouris, G., Dollimore, J., and Kindberg, T. Chapter 9 of Distributed Systems, Concepts and Design, 3rd edition, Addison-Wesley Longman, 2000.

[6] Digital Object Identifiers home page. http://www.doi.org/.

[7] Handle resolution system home page. http://www.handle.net/.

[8] Hecht, D.L. Embedded Data Glyph Technology for Hardcopy Digital Documents, Proc. Society of Photo-Optical Instrumentation Engineers Symp. on Electronic Imaging, Science and Technology, San Jose, Calif., 1994, Vol. 2171, 341-352.

[9] Holmquist, L., Redström, J., and Ljungstrand, P. Token-Based Access to Digital Information. Proceedings HUC 1999, 234-245.

[10] Kindberg, T., and Hawke, S. The 'tag' URI Scheme and URN Namespace. Internet-Draft draft-kindberg-tag-uri-01.txt. http://taguri.org.

[11] Kindberg, T., and Barton, J. A Web-based Nomadic Computing System. Computer Networks, Elsevier, Vol. 35, No. 4, March 2001, 443-456.

[12] Kindberg, T., Barton, J., Morgan, J., Becker, G., Caswell, D., Debaty, P., Gopal, G., Frid, M., Krishnan, V., Morris, H., Schettino, J., Serra, B., and Spasojevic, M. People, Places, Things: Web Presence for the Real World. Proc. 3rd Annual Wireless and Mobile Computer Systems and Applications, Monterey CA, USA, IEEE, Dec. 2000, 19-28.

[13] Kohtake, N. Rekimoto, J., and Anzai, Y. InfoStick: An Interaction Device for Inter-Appliance Computing. Proc. 1st Int'l Symp. on Handheld and Ubiquitous Computing 2000, 246-258.

[14] Leach P., and Salz, R. UUIDs and GUIDs. Internet-Draft Draft-leach-uuids-01, 1997.

[15] Mealling, M. Dynamic Delegation Discovery System (DDDS). Internet-Draft draft-ietf-urn-ddds-toc-02.txt, 2002.

[16] Mealling, M., and Daniel, R. RFC2483: URI Resolution Services Necessary for URN Resolution, 1999.

[17] Moats, M. RFC2141: URN syntax. 1997.

[18] Pradhan, S., Brignone, C., Cui, J.-H., McReynolds, A., and Smith, M. T. Websigns: Hyperlinking Physical Locations to the Web. IEEE Computer, Vol. 34, No. 8, August 2001, 42-48.

[19] Radia, S. Nelson, M., and Powell. M. The Spring Naming System. Tech. report 93-16, Sun Microsystems Laboratories. 1993.

[20] Rekimoto, J., and Ayatsuka, Y. CyberCode: Designing Augmented Reality Environments with Visual Tags, Proc. of Designing Augmented Reality Environments 2000, ACM.

[21] Want, R., Fishkin, K. P., Gujar, A., and Harrison, B. Bridging Physical and Virtual Worlds with Electronic Tags. In Proc. 1999 Conference on Human Factors in Computing Systems (CHI '99), 370-377.

[22] Want, R., and Russell, D. M. Ubiquitous Electronic Tagging. Distributed Systems Online, IEEE. http://www.computer.org/dsonline/articles/ds2wan.htm.

[23] XForms 1.0, W3C. M. Dubinko, J. Dietl, R. Merrick, D. Raggett, T. Raman, L. Bucsay Welsh (eds.). Available from www.w3.org/TR/xforms/.

[24] XML Linking Language (Xlink) Version 1.0, W3C. Steve DeRose, Eve Maler, David Orchard (eds.). Available from www.w3.org/TR/xlink/.