Ubiquitous and contextual identifier resolution for the real-world wide web

Tim Kindberg

Internet and Mobile Systems Laboratory
Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94304, USA
timothy@hpl.hp.com

Abstract. Identifier resolution is presented as a way to link the physical world with virtual applications and services. 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. 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 present a new way for users to specify relevant contextual parameters by a combination of sensing and navigation on the Web. In particular, they capture resolution services and applications in their local context. We use the Web to invoke resolution services, with a new model of 'physical' web form-filling. We propose a new scheme for binding identifiers to resources, to promote services and applications linking the physical and virtual worlds.

1 Introduction

This paper describes the design and implementation of a system for *identifier resolution* as a mechanism for linking the physical world with virtual applications and services. In this paradigm, which is designed to support nomadic users, the user employs a handheld, wirelessly connected, sensor-equipped device to read identifiers extracted from, attached to, or near physical entities. Those identifiers undergo *resolution* into virtual resources or actions that are related to the physical entity.

The result of resolving an identifier may be information or a service provided to the client, or an action in the environment. For example, the user could be offered a map of nearby points of interest when they resolve a 'you are here' barcode in a city; or purchasing details for a book whose barcode they scan in a shop; or citations of a barcoded paper that they print from the Web. When they scan the barcode on a can, they could be offered the 'virtual shopping list' service, which invites them to add the product to their list. When they scan a barcode placed where the light switch would normally be on the wall, the lights go on at their preferred setting. We used barcodes in those examples but other identification technologies are also available [18].

This work is based upon the belief that identifier resolution should be a mechanism that is both ubiquitous and contextual. 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, resolution should be available on a wide variety of handheld devices. For resolution to be contextual means that the desired resolution result may be a function of contextual parameters such as the user's location, their personal preferences (e.g. the user's language), their activity, and the device they are using.

If it is to be a ubiquitous and contextual facility, an ID-resolution system raises technical challenges in human factors, in system design, and in such pragmatic considerations as wireless connectivity and the physical tagging of identifiers. It also raises social and administrative challenges, such as management of identifiers and control over the binding of identifiers to virtual resources.
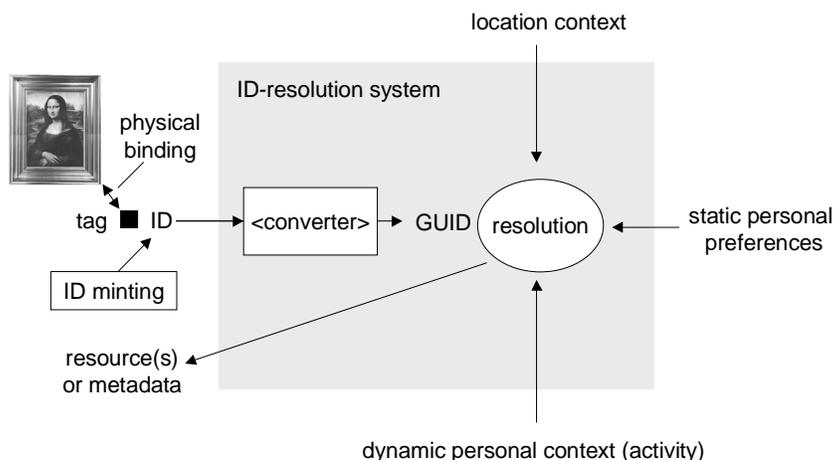
Fig. 1. The elements of an ID-resolution system.

This paper describes the research issues in developing a system for ubiquitous and contextual identifier resolution, and a design and implementation that we have developed for the Internet infrastructure and handheld devices.

1.1 Linking the physical world with the real world using identifiers

This work is part of the CoolTown project [1, 9, 10], which creates linkages between the physical and virtual worlds in the form of the 'real-world wide web': an integration of the Web with physical entities.

The system is aimed at users carrying handheld devices equipped with sensors and, usually but not necessarily, wireless connectivity. The real-world web adds to the conventional web in that users can find web links by sensing the physical world, as well as by browsing web pages. 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. 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 the form of a barcode, an RFID tag, an iButton, an infrared beacon, or the GPS coordinates of the entity that interests them. 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.

1.2 Ubiquity and context

This project's goal is a resolution system that is ubiquitous in its availability and which can be used in a variety of ways by different communities of users in various contexts. To achieve ubiquity we require a widely available service delivery platform on top of widespread wireless networking. We utilise the Web to deliver resolution services. We do so, as with earlier CoolTown work, 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.

An important characteristic of resolution is how the user's context is taken into account in the choice of resource(s) offered to them 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 scan 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 (GM) 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.

1.3 Contribution

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 new way for users to specify relevant contextual parameters, by a combination of sensing and navigation on the Web. They thus select, from a range of possible resolution services, one that is appropriate for their context. In particular, they capture resolution services and applications in their local context. We use the Web to invoke resolution services, with a new variant of web form-filling. We propose a new 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 real-world objects, and how those identifiers are bound into multiple naming contexts. Section 5 describes our design and implementation of web-based resolution. Section 6 concludes with a discussion.

2 Subtasks in an ID-resolution system

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 real-world 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 an object that associates an identifier with a virtual resource or metadata about the resource, in particular its address.

ID Capture. Identifiers are captured from real-world entities. An identifier may be derived from the entity's image, attached on or near it with a tag, or it may be an extrinsic individuating factor such as its position.

Context capture. Contextual parameters are also captured: the local environmental context (e.g. supermarket, home), the static personal preferences of the user, and the dynamic personal context of the user (e.g. their activity, their current device's capabilities).

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 a resource or a list of links to resources, which is sent to the client.

Thus *identifiers* are minted, captured and converted; and they are bound to resources. *Bindings* are created and looked up. *Context* is captured and fed, with the identifier, into *resolution*.

3 Related work

In the original CoolTown work [10] (Section 1.1), infrared beacons emit the URL of an entity's web presence. The service whose URL is emitted by a beacon may provide personalised content; but the user cannot choose the service for a given entity.

Several other projects have investigated tagging (identifier and sensing) technologies [18] and identifier resolution systems. Applications include information services [6, 16, 17], leaving virtual notes on physical objects [7], and content-transfer [11]. 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 for Uniform Resource Names (URNs) [12], the URN *u* is looked up to select a resolver for that URN, *R*(*u*); then *u* is resolved by *R*. The handle system [5] for Digital Object Identifiers (DOIs) [4] uses a similar two-tier scheme. A name-authority prefix within a DOI is used to select a resolver. Both resolution schemes assume that there is an 'authoritative' binding or set of bindings for a given identifier, one that can be located starting from knowledge of only the identifier itself. In some circumstances, that is a useful property. But no matter where one presents an identifier to the system, one always obtains the same answer, contrary to our approach.

4 Identifiers and bindings

An ID-resolution system 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. Global uniqueness over space and time is valuable because it guarantees that one identifier cannot conflict with another, even if entities that do not use one another's identifiers now come to share them in the future.

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 barcodes, EAN barcodes, 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 being able to include information in the name, such as the name of the party that created the identifier.

Convenience of minting identifiers. If individuals and small organisations such as shops are to be able to participate by attaching identifiers to their 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 [2] include URNs (Uniform Resource Names) [14] and URLs. URIs are, in general, variable-length strings intended for use by humans as well as software; they are infinitely extensible.

URNs are globally unique over space and time, by design. URLs can be considered as uninterpreted unique strings rather than as locators. As such, they are identifiers that are globally unique over space but not necessarily over time: two principals that are assigned the same domain name at different times might use the same URL to identify different entities. And URLs that are intended as pure identifiers are still liable to be used (erroneously) as locators.

Thus, we have proposed a new type of URI, a *tag* [8]. Tags are unique over space and time. 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 a uniquely assigned name and using that as a prefix.

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

4.3 Binding: an analysis

Two types of binding are very familiar and influential: (1) the physical binding of a barcoded identifier to a product and (2) the binding of names to IP numbers in DNS.

Acme Beanz minted the UPC for their cans of beans, and they assert the binding from the UPC to their beans. They alone control the physical binding. 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.

As mentioned in Section 3, the community developing URNs and systems for resolving them have a similar notion: that there is such a thing as the 'authoritative' binding for any given URN, and that the holder of that binding can be worked out from the name itself. They conceive of a single URN-resolution system that takes a URN and produces the authoritative resource bound to that URN.

However, consider again the example of the supermarket shoppers (Section 1.2). 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 [3]. (This use of the word '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.

Which of the bindings is 'authoritative'? Our answer is: they are all equally authoritative. Bindings derive from 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: No, not *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. The UPC Council have 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's 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 [15]. 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 or different resources. We can spell out the steps in our minting and binding model as follows:

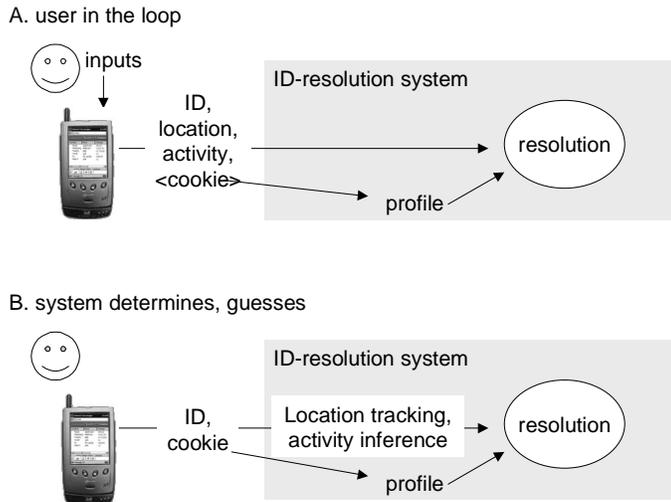1. a principal mints a new globally unique identifier (URI);

Fig. 2. User-centric and system-centric ways of gathering contextual parameters (logical view).

2. that principal creates an 'original binding' of that identifier onto some physical or virtual resource and publishes the binding. 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 Burlingame Intermediate 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 according to an XML schema. Such an 'Xbinding' object is based on Xlink [20] 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. the textual name and the URL of the home page of the principal that asserts the binding from item 1 to item 2;
5. a digital signature, by the principal identified in item 4, affirming items 1-4.

5 A web-based resolution system

This section deals with how captured identifiers are resolved to bindings or to the resources addressed by URLs in bindings. Resolution proceeds by taking into account both the identifier and the contextual parameters (Figure 1). The sub-task *resolution*(id, c_1, c_2, .., c_n), where c_1, c_2, .., c_n are the contextual parameters such as the user's location and activity, returns one or more bindings to which the given identifier and contextual parameters map.

Both parameter-capture and *resolution*(id, c_1, c_2, .., c_n) can be implemented in various ways with differing implications for deployment. We shall describe design options and the implementation we have built.

Context, and the ways in which context can and should be taken into account when performing ID-resolution, is a complex issue. There are the questions of which contextual parameters are relevant, which can be captured and how they should be captured.

Deciding what is relevant is particularly difficult. For example, when a user picks up a book in a bookstore, sometimes they may wish to see that store's pricing information; at other times they may wish to

see a review of the book. It is not clear what information the system could capture as a basis for offering one rather than the other. Providing all possible links, search-engine style, seems unsatisfactory unless they happen to be few.

Where the relevant parameters are clear, we can sometimes capture them, in various ways. Figure 2 shows two basic models of context capture, although there are several models in between. One, the user-centric model, is to have the user opt in to contextual specification, supplying such parameters as their activity (what type of answer they require) and location, and optionally using a cookie so that the system can keep a personal profile. The other, system-centric model is to take any such load off the user and have the system pick a result based upon their personal profile, location-tracking and activity-inference, etc.

One trade-off between the two models is between the load on the user and the cost of sometimes providing results that are inappropriate, through inaccurate tracking or inference. Another is between the load on the user and the loss of privacy that may come from tracking. The acceptability of the system may depend on privacy implications.

5.1 The web resolution paradigm

Our model assumes that users predominantly want to be in control of whether or not they disclose information about their context. In particular, the user senses (the identifier of) their location; their environment does not, by default, track them.

We also assume that, given well designed mechanisms, users are relatively good at specifying their activity and other contextual parameters, compared to how good systems are at inferring those parameters.

In our implementation, resolution is provided by many services — web sites — that can be independently administered. Users select their context by selecting a resolution service that matches their activity and location — a selection that they can make dynamically. Users 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, infrared and other sensors, or conventional hyperlinks.

Once the resolution service 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, they use 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 real-world object using a given resolution service is identifier-sensing: no other manipulations of the device are required, by default.

The following are examples of this web paradigm for resolution as it might be spoken of by users. We have not found a good generic word for 'sensing the identifier of' an object; we use the word 'scan' intended in that generic (as opposed to barcode-specific) sense:

"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. The scenarios assume conventions that the user has to understand about what those tags look like and where they can be found.
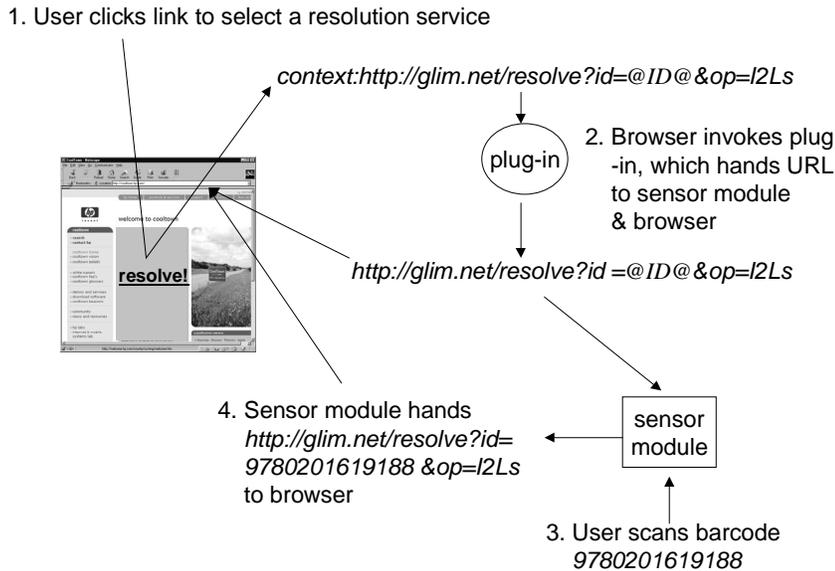
1. User clicks link to select a resolution service

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

plug-in

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

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

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

sensor module

3. User scans barcode *9780201619188*

Fig. 3. Prototype resolution client (runs on hand-held device).

The users in these scenarios are 'nomadic'. In general, they scan objects as they find them, not while sitting at a PC. Their resolver client enables them to navigate using virtual and physical Web links 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. 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 activity, 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 The resolution client

We have implemented a resolution client for the Symbol 1740 PalmOS-based device, which has an integrated barcode scanner and wireless connectivity. The Symbol client is built using the EudoraWeb browser. We have also implemented a client for Windows CE. That client 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 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 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 3). 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=@ID@&op=I2Ls.*

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

1. strips off the URL *u* from *context:u*;
2. 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. directs the sensing module to use the URL *u* for resolution.

When the sensing module reads an identifier, it:

1. locates the string "@ID@" within the resolution URL, configured by step (3) above, and replaces it with the sensed identifier to obtain a URL *u'*;
2. 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 field 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 of true web-based resolution, no form (in the sense that we understand it from HTML or the Xforms work [19]) has actually 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 fields that can be 'physically filled in' by an attached sensor. It would be possible to have several such fields within a single page and to allow those fields to be filled in by any of a variety of sensors — or by a human with a keyboard or stylus. We are working towards definition of a set of standards through which all of that can be realised.

In the meantime, the system we have implemented has proved to be quite powerful. Many options can be built around 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.

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 following examples with it:

Physical browsing. The user obtains information pages about items that they find, for example, while browsing a shop. And if a user finds no entry for a particular object, the resolver may invite them to add one.

Physical registration. To register (or unregister) entities such as printers and furniture as belonging to a place — and thus visible in the place's web pages — the user scans them at the place's administration page.

Light control. The user scans the 'lights on' barcode at 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. Variations of this application occur in the examples given above in Section 5.1: the virtual family message board at the café, the medicine bottle through which clinicians communicate, and fault-reporting for the printer. Note that these bulletin boards are not, in general, publicly accessible. For example, two families who scan the same café will see different, private sets of messages by virtue of scanning them at different web message board pages.

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
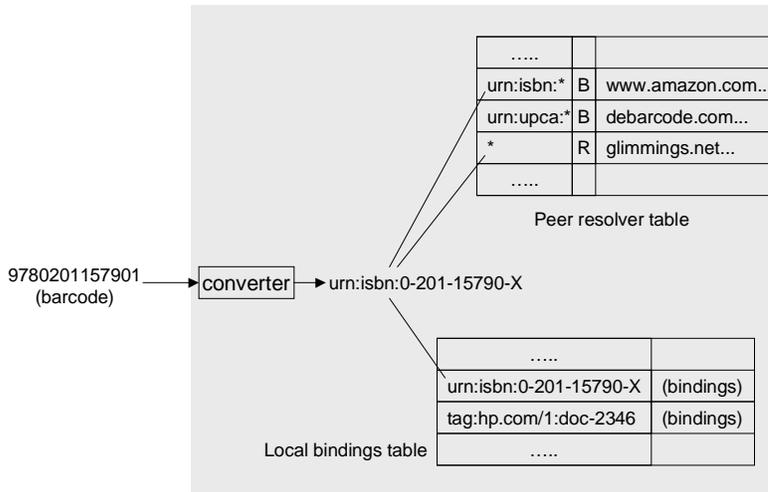
Fig. 4. CoolTown resolver components showing entries for a given identifier.

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. This application composes two naming contexts: the 'my music' naming context, which supplies the URLs of the user's MP3 files, and a naming context that returns the music rendering service for the scanned device.

5.4 CoolTown resolvers

The following requirements for resolvers emerged from constructing the foregoing 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 4), 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.

Resolution services. Resolvers provide the operations specified for URI resolution by Mealling and Daniel [13]: 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 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 can in a supermarket 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 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 4 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 binding objects as we defined them above, 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 a project whose aim is to provide ubiquitous and contextual identifier resolution, as a means of correlating real-world entities with virtual 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 an 'augmented 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. The paradigm provides a diversity of 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.

Some applications, such as the 'My music, your place' example in Section 5.3, 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 (see Figure 2.B). 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.

As future work, we are collaborating with the San Francisco Exploratorium to evaluate use of our paradigm by museum visitors who pick up web links from scanned identifiers next to exhibits.

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 are formulating an XML schema as a standards proposal for first-class binding objects. And we are defining standards proposals for sensor input to web forms — a mechanism which, we believe, goes beyond identifier resolution in its applications.

Acknowledgements

References

[1] John Barton & Tim Kindberg. "The challenges and opportunities of integrating the physical world and networked systems". HPL Technical report HPL-2001-18, 2001. Submitted to Mobicom 2001. Available as http://www.champignon.net/TimKindberg/MobicomChallengeAsTR.pdf.

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

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

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

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

[6] D.L. Hecht, "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, pp. 341-352.

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

[8] T. Kindberg. "Tag URIs". Proposal to be submitted as informational Internet-Draft. Available as http://www.champignon.net/TimKindberg/tagDraft.pdf.

[9] T. Kindberg and J. Barton. "A Web-based Nomadic Computing System". *Computer Networks*, Elsevier, vol 35, no. 4, March 2001, pp. 443-456. Available as http://www.cooltown.hp.com/papers/nomadic/nomadic.htm.

[10] T. Kindberg, J. Barton, J. Morgan, G. Becker, D. Caswell, P. Debaty, G. Gopal, M. Frid, V. Krishnan, H. Morris, J. Schettino, B. Serra, and M. Spasojevic. "People, Places, Things: Web Presence for the Real World". Proc. 3rd Annual Wireless and Mobile Computer Systems and Applications, Monterey CA, USA, Dec. 2000. p 19. Available as http://www.cooltown.hp.com/papers/webpres/-webpresence.htm.

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

[12] M. Mealling. "Dynamic delegation discovery system (DDDS)". Internet-Draft Draft-ietf-urn-ddds-03.

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

[14] R. Moats. RFC2141: "URN syntax". 1997.

[15] S. Radia, M. Nelson and M. Powell. "The Spring naming system". Tech. report 93-16, Sun Microsystems Laboratories. 1993.

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

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

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

[19] 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/.

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