Controlled Availability of Pervasive Web Services

Peter Robinson
University of Cambridge Computer Laboratory
15 JJ Thomson Avenue
Cambridge CB3 0FD
England
pr@cl.cam.ac.uk

Stefan Hild
IBM Zurich Research Laboratory
Säumerstrasse
8803 Rüschlikon
Switzerland
sgh@zurich.ibm.com

Abstract

The increasing use of computers to manage everyday business poses problems for workers when they are away from their home environment.

There are two key problems:

- *How can a mobile worker share a limited part of their home environment with colleagues in a foreign environment while maintaining its security?*
- *How can a mobile worker identify and interact with local services available in a foreign environment?*

This paper describes a scheme for controlling access to pervasive Web services together with a tangible user interface for their presentation.

1. Introduction

Ubiquitous and pervasive computing is often defined as the task of embedding small and mobile devices into existing IT and computing infrastructures, so allowing users to access and manipulate information where and when it matters, even while on the move. This characterization certainly portrays one aspect of the topic. It also leads the reader to believe that the main issues addressed in ubiquitous computing are those of wireless connectivity and the problem of manipulating data on devices with limited computation and user interfaces.

Advances in wireless communications have by and large eliminated the first concern: wireless connectivity is now commonplace and is provided through an arsenal of different bearer technologies that offer connectivity at a variety of bandwidth and cost configurations. At one end of the scale, BlueTooth-equipped devices can communicate with each other at comparatively high speeds in a peer-to-peer fashion; 802.11b wireless networks allow wireless connectivity within micro-cells that span offices and office-buildings, coffee-shops, public parks, and private houses. Both Bluetooth and 802.11 are typically operated privately and are available to the users at zero cost (although commercially operated 802.11 services are appearing). At the other end of the scale, most large-area cellular telephone networks now provide packet data services, such as GPRS services for GSM. They provide connectivity at modest speeds but at a price that is interesting even to the private, non-business, user.

Similarly, modern mobile computing devices now incorporate processors, memory, and disk configurations that are not significantly below some lower-end desktop PCs: processing and storage are therefore available to those mobile users who need it. However, limited screen space means that user interface aspects remain a concern. Application designers are learning to deal with such restrictions and methods are now available that allow applications to be written in a manner that clearly separates the application data back-end from the application logic and the user interface, thus allowing applications to adapt to a variety of end-user devices in their user interface and application behavior, both in the functionality they offer and expected usage model.

Ubiquitous and pervasive computing must be redefined accordingly. The problems are much less to do with the physical restrictions of the device and its interconnection to the rest of the Internet, and more to do with the way these devices interact with their environment and are capable of addressing user needs as the user moves through different physical environments. This means that pervasive computing devices not so much “on the move”, as “at a glance”. While the first characterization might also include “portable” computing devices such as notebook computers, the latter more accurately reflects the user’s expectation that a mobile devices such as a PDA or mobile phone can quickly be turned on (and off) to gather some piece of vital information to aid the user’s actions within the dynamic real-word environment. A further implication is that the user expects the device to adapt its behavior and derive information from the physical environment where the device is being operated. This requires the device to be capable of detecting its environment and to establish communications links into that environment.

Finally, the user must have confidence that only the desired services are invoked and that other information can not leak into a hostile environment. These challenges are addressed in this paper.

1.1. Scenarios

The application domain can conveniently be illustrated by reference to some possible scenarios in nomadic computing. Consider a worker travelling to a meeting away from his normal work place:

1. Setting out at the train station, he wants to find the time and platform of the next train to the remote site.
2. On arriving at his destination, he wants to find the nearest Internet café.
3. In the café he wants to edit a presentation stored on his office computer.
4. Leaving the café, he wants to call a nearby taxi to take him to his meeting.
5. Arriving at the meeting, he wants to exchange business cards with others present.
6. At the meeting, he wants to display the presentation that he has just edited on a local projector.
7. He then wants to share a folder of documents with others present at the meeting.
8. Finally, he wants to schedule a further meeting.

These activities can be classified in three categories:

- *Remote working* – activities 3, 6 and 7.
- *Peer-to-peer communication* – activities 5 and 8.
- *Location-dependent service discovery* – activities 1, 2 and 4.

The challenge is to find an underlying mechanism that will enable these activities together with a way of presenting that mechanism to the user that will engender confidence in the system's accuracy and security. This involves detecting and interacting with an alien environment.

Several technical questions arise:

- *Discovery* – How do the user and the environment identify themselves?
- *Registration* – How are transient services introduced into an environment and removed when no longer needed? How are competing names resolved?
- *Security* – How can subversion of services be prevented? How can a user be assured that no confidential information has leaked into the environment?
- *Naming* – What sort of registry is needed to support *ad hoc*, disconnected operation with naming by context and query as well as explicitly?

2. Background

These topics have been the subject of energetic research, both before and after the coining of the phrase "*ubiquitous computing*" by Mark Weiser [11] to describe it. Developments in this area can be observed as a movement from workstation based computing, through document and Web oriented views towards computing as an appliance.

2.1. Remote workstations

Andy Hopper identifies two distinct approaches to mobile computing [2]:

- In the *caravan* model, each worker carries a replica of his home computing environment in the form of a laptop computer when he travels. It provides a familiar and safe environment for work, but is cumbersome and offers only a limited quality of service.
- The alternative *hotel* model relies on pervasive computing at the remote site together with high-speed communications back to the home environment. It replicates the user's home environment through the local facilities, but performance is likely to be limited by the communications network.

The key to the hotel model is Virtual Network Computing (VNC) [10], which provides remote display, keyboard and mouse securely over the Internet, effectively "teleporting" the user's workstation to a new environment. However, the system can only be used at appropriately equipped locations and the use of public computing facilities limits confidentiality.

More importantly, remote workstations do not exploit any knowledge of the local environment through peer-to-peer operation or location-dependent service discovery.

2.2. Document view

The Satchel system [3][7] developed at the Xerox Research Centre in Cambridge offers an alternative approach to remote access and sharing of remote documents. Individual documents are represented by tokens, which are effectively URLs interpreted by a personal Web server on the user's home computer. Tokens are managed on a handheld computer (the Satchel) which communicates with other Satchels and with fixed devices such as printers using infra-red. Communication with the home computer is provided either by a gateway supporting infra-red and connected to the Internet, or by wireless connection over a GSM network. The tokens are protected by a cryptographic signature which gives a finer-grain of control than that provided by a conventional Web server.

The Satchel client program has the appearance of a Web browser, and this provides a user interface for navigating a directory hierarchy in the usual way. Computers and other devices that are prepared to offer specialised services do so by responding to a service enquiry initiated over infra-red. This might be another user's Satchel, a printer, a computer that is prepared to serve as a display for documents, or a scanner that is prepared to convert paper documents into electronic form. Each service is provided by a specialised Web server that might return a form asking for further details, for example of the identity of a document to be printed and the number of copies wanted.

A great strength of the system is its encapsulation of the user interface and private keys in a portable user interface. This gives the user confidence in its security. The short range and directional transmission of infra-red makes the identification of services very clear. However, the cost of making a service available is considerable.

Interaction with the local environment is limited to the detection and use of peripheral devices for input and output. There are no general location-dependent services.

2.3. Web view

The Cooltown project at HP Laboratories [5] addresses the problem of service discovery for nomadic computing. The key idea is the physical presentation of Web addresses through infra-red beacons, barcodes and radio-frequency tags. The human user addresses pervasive computing facilities on a handheld computer through direct interactions with physical objects in the local environment.

The system addresses the questions of detecting the environment and interacting with it. However, standard Web protocols are used without introducing additional middleware. This strength is also a weakness, limiting the range of applications which the technology can address.

2.4. Information appliances

Donald Norman [9] characterises the coming of age of computer technology as a move towards information appliances. These are devices that contain computers but are designed to perform specific tasks, rather than being general purpose computers programmed to perform those tasks. Computers and communications may be embedded in the world around us, but they will not be workstations and may not understand documents or even Web pages.

This view of ubiquitous computing requires a new view of user interfaces. In principle, each appliance has a user interface that is designed to be appropriate for its specific task. This might be appropriate for free-standing devices but is less clear when communication with other devices is required. This requires data streams to be identified as path names in a directory system or URLs or whatever, together with authentication information to prevent unauthorised access.

Something like the Xerox Satchel offers a solution to this problem. The user carries tokens representing remotely accessible data in a handheld computer which also provides the user interface. This neatly removes the need for the appliance to have any user interface at all. Appliances simply export a service and the user's portable device provides the interface, which can suit the user's personal preferences, also avoiding the problems that working in an unfamiliar environment can pose.

This approach to computing is attractive but does not address the particular questions associated with discovery of location-dependent services and interaction with them.

2.5. Web Services

Location-dependent services do not fit comfortably within the appliance model. Fixed appliances might work but it is more likely that the user would prefer to use his own display and user interface to discover the information.

The Web Services model [4][6] offers a more general view of ubiquitous computing[1]. A Web Service is an interface that exposes a set of tasks available on a network. Their advantage is that they provide two functions that will be of significant value as we attempt to migrate existing IT infrastructures into a dynamic environment where consumers of services bind dynamically, and on-demand to a changing services backend.

First, it provides service description through its Web Services Descriptions Language (WSDL). Today WSDL is an XML-based and machine-readable description of available RPC interfaces, much like where available in the IDL definitions under CORBA. Efforts are on their way to extend WSDL to include descriptive components that classify the service that is offered, for example by adding semantic information to the available RPC exits.

Secondly, it provides a framework for dynamic service discovery, chiefly through the user of its registry infrastructure that is built around Universal Description, Discovery and Integration (UDDI) registries[2]. Work is under way to enable those registries to be arranged in a hierarchical fashion, or to allow private UDDI registries to exist that provide service descriptions for a particular (private) environment or geography.

Those two elements provide a self-sufficient infrastructure that allows clients to discover available services dynamically, and also to bind to them and invoke them.

What is not yet adequately covered within the Web Services spectrum is how such user interface components of the discovery can be arranged in a user-friendly manner, and how information on Web Services that one user has discovered or wants to provide can be passed between users.

3. Architecture

Our aim is to allow a nomadic worker to use pervasive computing facilities in a way that is:

- *Clear* – offering a clear choice of service to the user.
- *Calm* – presenting no greater complexity than dealing with a human agent.
- *Confident* – assuring the user that security is being maintained.

In addition, it is sensible to build on open standards. We therefore use the Web Services model.

[1] http://www.webservices.org/

[2] http://www.uddi.org/

3.1. Service tokens

The first issue is the identifying a convenient representation for Web Services. The model here is a letter of authority, identifying the service and conferring permission to use it. In the terminology of computer operating systems, it is a capability [12]

The service to be identified might be an information source such as a Web page, an information repository that can be modified such as a file or a directory, or a device such as a printer, projector or workstation. In the Web Services model, any of these can be represented as a service and identified by the address of the service provider and an index pointer for the service at that provider. It is convenient to think of the provider as an object and the service as a method invoked within that object. This pair confers access rights and so must be protected by a cryptographic signature. A service token is just a triple combining the identity of the provider with the identity of the service and a signature issued by the provider.

Tokens also carry implicit access control rights and expiry times, which are recorded in the environment of the associated method in the provider object. This allows a single file to be exposed as read-only or to limit write permission for a fixed period of time. Access rights are stored in the server. Different tokens would be created for a single service that is being delivered with different access controls or with different expiry times.

The provider must also allow the refinement of tokens. A user holding a token conferring full access rights to a directory hierarchy must be able to ask the provider to issue a further token conferring more limited rights that can be passed on to a third party. The two tokens may well invoke the same code within the provider but the methods will be distinguished by additional information in their environments that records the access rights and expiry times.

Each user stores tokens in a computer that communicates with the providers to arrange for refinement and with other users to pass on the tokens.

3.2. Physical representation of tokens

Users may exchange tokens electronically, using any appropriate wireless communications. This might be infra-red as with the Xerox Satchel, or it might use radio such as Bluetooth or GSM. However, some services will be advertised to all users. These could be represented by infra-red beacons or RF tags as with HP's Cooltown, but it would be preferable if they could be produced more cheaply. For example, a token for a user's home Web page might be included on his business card.

The obvious solution is to produce printed representations of tokens, for example as a barcode. Individual users could produce these for services that they wished to make available to others. The railway company could include barcodes on signs at the station and even on tickets. Different destinations on a map could be marked by different barcodes that represented the service tokens for Web pages giving details of the times and platforms of trains to the corresponding destination. It would also be possible to display varying service tokens on a screen as part of a video or computer presentation.

3.3. Service tokens in practice

To illustrate this scheme, let us revisit the scenarios sketched in the introduction:

1. Setting out at the train station, he wants to find the time and platform of the next train to the remote site.

   The railway company offers its timetable as a Web Service and displays physical tokens around the stations to advertise the service to users with handheld computers. Invoking these services displays the time and platform for departures. The same tokens could be printed on tickets. A traveller could show this to a public terminal to display the same information.

2. On arriving at his destination, he wants to find the nearest Internet café.

   Advertisements for the café could carry physical tokens pointing to a Web page advertising its services. Variants of the token could pass further information to the advertising service to customise it to the location of the advertisement and give specific instructions for the route to be followed.

3. In the café he wants to edit a presentation stored on his office computer.

   This involves two sets of transactions. First the user has to take control of a workstation in the café. Then he has to allow that workstation to read and modify a file on his home filing system.

   Each workstation in the café could carry a physical token for the service allowing a user to reserve it. The service could display feedback on the workstation's screen to confirm that the badges had not been exchanged. The reservation service would allow the user to transfer some tokens to the workstation which would be available for his session on it.

   Before setting out, the mobile worker prepares a token giving complete access to his whole home filing system. Before using the workstation, he could refine this using a directory browser on his handheld computer to produce a token offering access to a particular file or directory for a limited time, and then pass this token to the workstation as part of the reservation process.

4. Leaving the café, he wants to call a nearby taxi to take him to his meeting.

   An advertisement for a taxi company is rather like an advertisement for the Internet café. However, the service indicated by the token could use real-time information on the availability and location of taxis to give an indication of the waiting time. After accepting a reservation the service could continue to report on the taxi's progress towards its prospective customer.

5. Arriving at the meeting, he wants to exchange business cards with others present.

   There are all sorts of good reasons for retaining printed business cards. Including a physical token would allow them to be used as a pointer to the person's home Web page. However, people could just hold up cards printed with their home page physical tokens and allow all the other people at the meeting to inspect them by reading the barcodes, avoiding the need to exchange conventional cards.

6. At the meeting, he wants to display the presentation that he has just edited on a local projector.

   Using a projector is rather like editing a file in an Internet café. However, the user would refine a read-only token for the presentation in his home filing system and his portable computer would provide a graphical user interface. This would reflect the user's preferences for the interface rather than any more idiosyncratic combination imposed by the projector and presentation software. Devices export programming interfaces rather than user interfaces.

7. He then wants to share a folder of documents with others present at the meeting.

   The user prepares a refined token for the directory, just as he did for the workstation in the Internet café. A physical representation of this could then be printed on any local printer using a scheme similar to that for controlling the projector. Alternatively it could just be displayed on the screen of his handheld computer. The other people present could then capture this token and acquire access rights to the files.

8. Finally, he wants to schedule a further meeting.

   A user's home page service might well include a pointer to a scheduling service that would manage the owner's diary automatically. However, it would also be possible to run such a service locally on the user's handheld computer. The question then becomes one of how the two users' services can contact each other in the absence of any centralised registry. The solution is for the user to prepare a token for the personal service and display it on the screen of his handheld computer, allowing the other user to inspect it and so gain access to the local scheduling service.

4. Evaluation

Service tokens address the questions of service discovery and interaction with local services, but would they be genuinely usable in practice?

It is too easy to describe a user interface as "intuitive" without analysing what this actually means. (It often means that the system's author can use it.) Victoria Bellotti [1] poses five criteria by which user interfaces to ubiquitous computing systems can be judged:

- *Address* – How does the system know that I am addressing it?
- *Attention* – How do I know that the system is listening to me?
- *Action* – How does the system infer the context of a command?
- *Alignment* – How do I know that the system has understood?
- *Accident* – How do I recover from mistakes?

These provide a convenient framework for analysing approaches to ubiquitous computing.

The tangible nature of the physical tokens ensures clarity of address. The user can control his handheld computer's field of view so that it only inspects a single token. Attention is signalled by activity resulting from invoking the Web Service referred to by the token. Of course, there is the possibility of fraudulent tokens. A mischief-maker might swap the tokens attached to two workstations in the Internet café. However, the response on the workstation's own screen from its reservation service gives a direct confirmation. An unscrupulous taxi company might paste its own tokens over those on a rival's advertisement. However, this is no different from pasting a different telephone number over the contact details on an advertisement.

The physical tokens are extremely cheap to produce and can represent services that have been tailored for a specific location. This conveys the context of a command implicitly and the resulting information from the service can confirm the alignment.

The user retains a personalised interface to the services being invoked on his handheld computer, so intervention after an accident is achieved using familiar mechanisms.

The main observation is that problems with ubiquitous computing systems arise through the intangible nature of their control systems. Voice control and local radio introduce the problems that Bellotti has characterised. Introducing a tangible component into the user interface greatly reduces the potential for difficulties with address and action. Giving the user a portable, personal display allows confirmation of attention and alignment, and provides a familiar environment in which to deal with accidents.

5. Implementation

A full implementation of the system is currently in progress. This section presents some of the key aspects resolved so far.

5.1. Handheld computer

Our users will have to carry some personal computer equipped with wireless communications as their personal interface to pervasive Web services. How much of a burden must this be? Many users will already be carrying personal mobile phones; perhaps these could be used as the interface.

Recent mobile phones offer a possible solution. These have ample processing power and provide an environment for additional applications. They support infra-red, Bluetooth and GPRS communications. They have colour screens and also include a 640×480-pixel colour camera. All this is provided in a 180cc package weighing 160g which also provides voice communications.

The camera is particularly interesting because it provides a natural means for reading physical tags in the user's immediate environment. TRIP codes [8] are circular bar-codes that can be read using a digital camera and simple image processing. They require about 20 pixels for each bit of data, so a 50×50 pixel image could easily convey a 96-bit token. This is only 1% of the image area (10% in each linear dimension) of a VGA camera.

The physical size of the physical token can be as little as 1cm in diameter with no upper limit. They could be printed on railway tickets, on advertising posters or even displayed on public display screens.

5.2. Communications

Two different communications systems are envisaged. A cheap (or even free) local wireless system such as BlueTooth or 802.11b is used for peer-to-peer exchange and a universal (but more expensive) system such as GPRS is used to contact remote services, particularly those in the user's home environment. Of course, one service available over the local system might be a cheaper relay for long-haul traffic.

5.3. Wearable computer

The system could also be presented through a wearable computer. A pair of spectacles augmented with a head-up display and a video camera connected to a PDA equipped with wireless communications would provide the computing environment. A user would simply look at a scene including a printed service token – the scene from the camera would be analysed, the corresponding Web Service invoked and the results presented on the head-up display.

6. Conclusions

This paper has considered the problems faced by a nomadic computer user in locating services in his environment and communicating with them. In particular, the requirement to export restricted access to parts of his home environment to other users in a remote location has been addressed by introducing service tokens. A physical representation of service tokens facilitates the discovery of services in a remote location.

The use of service tokens has been illustrated by a range of case studies and the tangible interface evaluated against usability criteria. The resulting system offers controlled access to local and remote services together with a straightforward tangible user interface for the identification of services.

A practical implementation is in progress and an extended variant for use with a wearable computer has been described.

7. References

[1] Victoria Bellotti, Maribeth Back, W Keith Edwards, Rebecca E Grinter, Austin Henderson and Cristina Lopes: Making sense of sensing systems – five questions for designers and researchers, *Proceedings ACM Conference on Human Factors in Computing Systems*, Minnesota, April 2002, pp 415-422.

[2] Frazer Bennett, Tristan Richardson, Andy Harter and Andy Hopper: Teleporting – making applications mobile, *Proceedings Conference on Computer Support for Collaborative Working*, Chapel Hill, October 1994.

[3] Mike Flynn, David Pendlebury, Chris Jones, Marge Eldridge and Mik Lamming: The Satchel system architecture – mobile access to documents and services, *Mobile Networks and Applications* **5**(4), December 2000, pp 243-258.

[4] Karl Gottschalk, Stephen Graham, Heather Kreger and James Snell: Introduction to Web services architecture, *IBM Systems Journal* **41**(2), 2002, pp 170-177.

[5] Tim Kindberg and John Barton: A Web-based nomadic computing system, *Computer Networks* **35**(4), March 2001, pp 443-456.

[6] Heather Kreger: *Web services conceptual architecture*, IBM Software Group, Somers NY, May 2001.[3]

[7] Mik Lamming, Marge Eldridge, Mike Flynn, Chris Jones and David Pendlebury: Satchel – providing access to any document, any time, anywhere, *ACM Transactions on Computer-Human Interaction* **7**(3), September 2000, pp 322-352.

[8] Diego López de Ipiña and Sai-Lai Lo: Sentient computing for everyone, *Proceedings IFIP Conference on Distributed Applications and Interoperable Systems* (DAIS'2001), Krakow, Poland, September 2001.

[9] Donald A Norman: *The invisible computer – why good products can fail, the personal computer is so complex, and information appliances are the solution*, MIT Press, 1998.

[10] Tristan Richardson, Quentin Stafford-Fraser, Kenneth R Wood and Andy Hopper: Virtual Network Computing, *IEEE Internet Computing* **2**(1), January 1998, pp 33-38.

[11] Mark Weiser: The computer for the 21st century, *Scientific American* **265**(3), September 1991, pp 94-104.

[12] Maurice Wilkes and Roger Needham: *The Cambridge CAP computer and its operating system*, North Holland, 1979.

[3] http://www.ibm.com/software/solutions/webservices/pdf/WSCA.pdf