DISCOVERY AND INTERACTION IN HETEROGENEOUS DEVICE AND NETWORK ENVIRONMENTS

Elaine Catterall[1&2], Nigel Davies[1] & Adrian Friday[1]

1. Introduction

The background to this work is the vision of ubiquitous computing: to make many computers available throughout the physical environment, while making them effectively invisible to the user. The user should be able to continuously interact with many wirelessly interconnected devices, without the underlying computing architectures, and networks “getting in the way” [Weiser].

There is currently a plethora of wired and wireless networking technologies, e.g. Ethernet, IEEE 1394, HomePNA, HomeRF, IEEE 802.11b and GSM. The consequential heterogeneity in a typical network is compounded by the emergence of low-cost, high-quality mobile devices, such as PDAs and notebooks. In addition, the development of smart appliances and the prevalence of set-top boxes as well as Internet enabled games consoles, increases the complexity and diversity still further. While these technologies provide unparalleled opportunities for connectivity, they present problems with respect to interoperability.

2. Service Discovery & Interaction

The desirability of zero-configuration networks, whereby devices can join the network and “just work”, has driven the development of significant numbers of device interaction technologies, such as Jini [Jini], HAVi [HAVi], SLP [SLPv2], UPnP [UPnP], and Salutation [Salutation]. The function of device interaction technologies is to allow users and applications roaming between networks to discover and interact with the services provided by devices on the network, without requiring detailed knowledge of the local network configuration.

[1]Computing Department, Lancaster University.
[2]Telcordia Technologies, NJ, USA. Case studentship

A typical scenario would be a user in an unfamiliar environment wishing to print a document from a notebook computer wirelessly connected to the network on a printer located nearby. Other examples might be automatically finding a mail server or HTTP proxy, and controlling devices across the Internet.

Users and applications want to search and browse the services available, then select a service by matching desired characteristics or attributes. Service providers want to deploy a service or a device in the heterogeneous environment and be confident that it can be discovered and used.

A survey of existing technologies shows us that the most common discovery and interaction services include SLP, UPnP, Jini, Salutation, Bluetooth’s Service Discovery Protocol, and HAVi. In this section we examine SLP, UPnP, and Jini in more depth. We also present an overview of a discovery service, SDS, incorporating a strong security model.

2.1. Service Location Protocol (SLP)

SLP [SLPv2] uses a combination of multicast and unicast messages between three distinct system components: user agents, service agents and directory agents. User agents typically reside on client devices and perform resource discovery. Service agents advertise the location and attributes of services on behalf of devices providing services. Directory agents aggregate service information, caching service offers and responding to client enquiries on behalf of service agents. Directory agents are optional, but become necessary to provide scalability when hundreds or thousands of services are present on the network. If they are not present the amount of multicast traffic can overwhelm the network. Services are advertised using a Service URL and a Service Template. The URL contains the IP address of the service, the port number and path. The template specifies the attributes that characterise the service and their default values [Guttman]. The SLP API [Kempft] is an interface that allows developers to use SLP in their applications to support service location.

2.2. UPnP

The two components of a Universal Plug and Play service are a Device and a Control Point [UPnPArch]. When added to a network the control point will search for devices using unique IDs or device service types. The device advertises its presence when it joins the network and refreshes itself periodically. A device is also responsible for responding to search requests. The discovery protocol used is Simple Service Discovery Protocol [SSDP], which relies on [GENA] (General Event Notification Architecture), for subscribing to and unsubscribing from events, as well as notification of device availability or state variable changes. The device and the services it offers are described in XML. UPnP also uses Simple Object Access Protocol (SOAP); an XML based protocol, for information exchange.

2.3. Jini

Jini is a distributed systems platform based on Java. It provides a number of services including naming, lookup, discovery, and an event notification model. Jini presupposes an underlying platform of Java and Java RMI, which in turn rely on TCP/IP support. However, Jini services can act as proxies for non-Java devices.

Jini specifies [Sun99] a suite of three protocols that enable Jini devices to discover Lookup services on the network: the multicast request protocol, the multicast announcement protocol, and the unicast discovery protocol.

The Jini Lookup Service provides a central registry of services available within a group of Jini users. Services register with the Lookup Service and include attributes, which describe their service and stubs for their invocation through RMI. Matching is done by querying the Lookup Service for objects with matching attributes. The types of attributes are defined in the Lookup Attribute Schema Specification.

2.4. Service Discovery Service (SDS)

The secure Service Discovery Service is a component of the [Ninja] distributed services platform. The SDS is a directory-style service that provides a contact point for making complex queries against the cached service descriptions advertised by services. The architecture is composed of five components: SDS servers, services, capability managers, certificate authorities, and clients [Czerwinski]. Service descriptions and queries are specified in XML.

The security model is an important feature of this discovery service. Privacy and integrity are maintained via encryption of all information sent between system entities. The trustworthiness of particular services is determined using public-key certificates to provide authentication.

3. Analysis

Many of the existing service interaction technologies operate in a similar way, and have a number of common features: client agents; service agents; registries with update and clean-up mechanisms; discovery and lookup mechanisms; and negotiation mechanisms to provide access to and use of the services. Additionally, most rely on IP multicast, posing problems for devices that use point-to-point technologies for connectivity.

Also common is the lack of adequate security models. An exception is the secure Service Discovery Service. Here, security is provided by the use of encryption and public-key certificates.

Query support is often difficult [Hughes] or non-existent, with reasoning taking place on the client. Because the dynamic state of a service is not typically part of the service advertisement or discovery phases of the protocols, clients must enumerate through the services to identify the best one. Richer state-based query semantics are desirable, particularly when supporting mobile access, where constraints such as partial network connectivity or low bandwidth are evident.

There is limited support for meta-data, and among the mainstream protocols, no support for context or location awareness (although some protocols, notably SLP, allow administrative scopes). Location awareness is very useful feature in ubiquitous mobile computing. Location-based usage typically involves some interaction with the physical environment associated with a location context, or the provision of information associated with a location context. José's paper [José] describes a Location-Based Services platform intended to scale to the wide area. Explicit location scopes give two collocated devices access to the same service information, irrespective of their network or administrative domain.

An additional shortcoming of the existing discovery and interaction technologies is that the issues of user, role, or application specific data, are not addressed. An example of user data could be a user's bookmarks of their favourite services.

Some reliability support is present in most of the service discovery and interaction technologies, with the concept of leases almost universal. See [Bettstetter] for a more in-depth comparison of the protocols.

Despite the logical similarity of the technologies, the implementations are diverse, and therefore incompatible. One of the reasons for this is that they were constructed with different domains in mind, and in the spirit of experimentation. For example, SLP is a very scalable discovery protocol, intended to serve enterprise networks; UPnP targets home and small office computing environments; while HAVi was designed to enable interoperability in home AV networks, and presupposes 1394 as the underlying technology.

There are special requirements arising from mobile computing and ad-hoc networking. Connectivity of mobile devices in wireless networks is often intermittent, and devices and services will roam between different networks. The principle requirements for the platform are:

- **Scalability** - Devices joining the network want a complete picture of the status of the network and the services available. When the network consists of many devices offering perhaps hundreds of services, this information must be well managed, otherwise the network could easily become saturated [McGrath].
- **Interoperability** – Networked devices may spontaneously and unpredictably join the network. Automatic configuration is needed, without user concern for the underlying transport mechanisms and network technology, if interoperability is to be achieved.
- **Reliability** – This is a particular concern in mobile networks, especially in view of the fact that mobile devices may be intermittently connected to the network. Recovery in the event of failure or removal of services is vital.
- **Flexibility** – Users and applications want to query and select services based on their current state. Rich query semantics are necessary.
- **Security** – This is required for purposes of authentication, to maintain privacy, and promote "trust".
- **Performance** – Reasonable response times and minimising the use of bandwidth are both important considerations. A lightweight, efficient solution is necessary if it is to be used on resource-poor wireless devices.

4. The Need for an Integrated Approach

The diversity of the domains in which the different device interaction technologies operate mean it is unlikely that a "winner" will emerge as the dominant platform. Therefore there will continue to be a corresponding diversity in device interaction technologies.

One approach to the problem of interoperability between the device interaction technologies is the construction of bridges between protocol pairs, such as between Jini & SLP. This solution is not ideal, as it focuses on an interface between specific protocols and so lacks a unifying model of interoperability. This makes it harder for application developers to design products for this environment. We feel a model that can support full interoperability between current and future device interaction technologies is necessary for a number of reasons:

- To support the development of services that can operate in heterogeneous environments.
- To support the migration of application across different devices and networks.
- To support aspects of ubiquitous computing.

More general approaches to the interoperability problem include IBM's effort to address these same issues of interoperability [IBM], and the work by the Mobile Network Computing Reference Specification Consortium [MNCRS]. The Open Services Gateway initiative (OSGi) aims to provide a complete end-to-end system architecture for remote service providers to deliver services to client devices on local networks [OSGi]. One scenario envisaged by OSGi is that of utility companies accessing local devices in consumer's homes through a gateway, to provide services such as remote meter reading.

Another approach is the COUGAR Device Database Project [Bonnet]. This is an early system that represents a device network as a database. This system is employed in monitoring data from sensor devices. Bonnet concludes that the application of database technology can provide flexible and scalable access to large collections of devices. This work also highlights the need to maintain meta-data in a decentralised way and devise appropriate query plans to utilise the information. It concludes that the application of database technology shows promise for providing flexible and scalable access to large collections of devices. Unlike our approach this system replaces the current service discovery and device interaction technologies with it's own version. We see the existing technologies as an essential underpinning to our own platform.

Ubiquitous computing applications will find it necessary to discover and interact with a wide range of services and devices. Moreover, the applications will be required to do this without considerable administrative and configuration overhead, and with minimal user intervention. The design of our platform will offer a unifying API for application developers, to support the development of ubiquitous computing applications in highly dynamic and heterogeneous service environments.

We anticipate that a crucial issue, particularly in terms of utility, will be the ability to reason about state selection and interaction on behalf of lightweight, potentially mobile clients.

5. Conclusions

The explosion of device interaction technologies has resulted in attempts to "bridge" the different technologies. We have pointed out the shortfalls of this technique and argued that a new architecture implementing a unifying service model is required.

It is our intention to build a system based on the unified device interaction framework. Work is currently focused on establishing a testbed using various interaction technologies such as UPnP, Jini and SLP. Work will follow to refine the requirements, define the unifying model for interoperability, and design the architecture to support the rich interaction we envisage.

References:

[Bettstetter] Christian Bettstetter, Christoph Renner, "A Comparison of Service Discovery Protocols and Implementation of the Service Location Protocol", Proc. EUNICE 2000, September 2000.

[Bonnet] Phillippe Bonnet, Johannes Gehrke, Praveen Seshardri, "Querying the Physical World", *IEEE Personal Communications*, October 2000.

Special Issue on Smart Space and Environments
[Czerwinski] Czerwinski, S.E., Zhao, B.Y., Hodes, T., Joseph, A.D., Katz, R.H. An Architecture for a Secure Service Discovery Service. Mobicom '99 Seattle, Washington, USA. ACM 1999.
[GENA] GENA website available as, http://www.upnp.org/draft-cohen-gena-client-01.txt
[Guttman] Guttman, E., Perkins, C., Kempf, J. Service Templates and Service Schemes. RFC 2609 (1999)
[HAVi] HAVi Consortium http://www.havi.org/
[Hughes] Hughes, E., McCormack, D., Barbeau, M., Bordeleau, F. An Application for Discovery, Configuration, and Installation of SLP Services. MICON 2000.
[IBM] IBM White Paper, "Discovering devices and services in home networking". http://www-3.ibm.com/pvc/tech/networking.shtml
[Jini] Jini. http://www.sun.com/jini/
[José] José, R., Davies, N. Scalable and Flexible Location-Based Services for Ubiquitous Information Access. First International Symposium on Handheld and Ubiquitous Computing, HUC'99, Karlsruhe, Germany, 1999.
[Kempft] Kempft, J., Guttman, E. An API for Service Location. RFC 2614. (1999)
[McGrath] McGrath, Robert E., "Discovery and Its Discontents: Discovery Protocols for Ubiquitous Computing" Technical Report UIUCDCS-R-99-2132 http://choices.cs.uiuc.edu/2k/
[MNCRS] Mobile Network Computing Reference Specification. http://www.mncrs.org/
[Ninja] The Ninja Team. The Ninja Project http://ninja.cs.berkeley.edu
[OSGi] Open Services Gateway Initiative, Specification Overview, Version 1.0, January, 2000. http://www.osgi.org/
[Salutation] The Salutation Consortium http://www.saluation.org
[SSDP] SSDP, website available as, http://www.upnp.org/draft_cai_ssdp_v1_03.txt
[SLPv2] Guttman, E., Perkins, C., Veizades, J., Day, M. Service Location Protocol, Version 2. RFC 2608 (1999).
[Sun 99] Sun Microsystems Inc. Sun Jini Specification http://www.sun.com/jini/specs/
[UPnP] Universal Plug and Play Forum http://www.upnp.org/
[UPnPArch] Universal Plug and Play Device Architecture, website available as, http://www.upnp.org/UPnPDevice_Architecture_1.0.htm
[Weiser] Weiser, M. "Some Computer Science Issues in Ubiquitous Computing". *Communications of the ACM*, 36(7): 75-84.