The Ubiquitous Web, UPnP and Smart Homes

Franklin Reynolds
Franklin.reynolds@nokia.com
Pervasive Computing Group
Nokia Research Center, Cambridge

Abstract. UPnP is a distributed computing framework based on Web technologies that is intended for use in local area networks – especially home networks. Over the years, valuable lessons have been learned from the efforts of various companies to develop and deploy products based on this technology.

1.INTRODUCTION

Ubiquitous computing research covers such a wide range of topics including protocol design, radio technology, human factors, system integration, etc. that it is somewhat difficult to clearly describe. Even the notion of what constitutes a ubiquitous computing environment or application is open to frequent debate. Our vision of ubiquitous or pervasive computing is captured by the following quote from the IEEE Pervasive Computing Magazine:

> "The essence of this vision is the creation of environments saturated with computing and wireless communication, yet gracefully integrated with human users. Many key building blocks needed for this vision are now viable commercial technologies: wearable and handheld computers, high bandwidth wireless communication, location sensing mechanisms, and so on. The challenge is to combine these technologies into a seamless whole."

In our lab, we have studied different approaches for creating local ubiquitous computing environments such as smart rooms or smart homes. Our work has always emphasized the importance of integrating different technologies and products, ease of use and deployment, and value of the applications to the consumer. We have paid particular attention to smart homes and have identified the following broad categories of smart home applications:

- Digital Content: audio, images, MP3, web content, etc.
- Games and entertainment
- Home automation (remote controls, security, environmental controls, etc.)
- Communication: VOIP, video calls, instant messaging, tele-presence, ambient presence
- Home office applications
- Health Care in place (assisted living and care-giving)
- Education
- Automobiles as an integrated, though mobile, part of homes

2.CHALLENGES

Smart homes pose interesting challenges for researchers. Perhaps the most important point is that end-users with little or no special knowledge of computers and networks will have to choose, deploy, integrate manage and operate most of the products and technologies in their homes. Enterprises can rely on their trained IT staff or they can hire consultants to guarantee the functioning of their networks and distributed applications. Within the home, most consumers will have to provide this service themselves. Other points worth mentioning include:

- Consumers make bad network and security administrators
- Security issues are not limited privacy or confidentiality, they include physical safety

- Access control must be flexible enough to accommodate different roles including owner, family, friend, visitor, maintenance/support, hostile, unknowns
- Smart homes will be incrementally deployed and their configuration will be moderately dynamic
- Transient connectivity will be a recurring problem
- Whenever possible no infrastructure services should be assumed
- Legacy products need to be supported and new products should be future proof
- Third party interoperability is essential
- Components will fail – someone has to figure out what is broken and how to fix it

3.UPNP

UPnP – Universal Plug and Play – is a technology intended for smart homes, small offices and other types of local area networks. It was originally created by Microsoft Corporation in 1999. UPnP is now under the control of the UPNP forum (http://www.upnp.org), an independent organization with over 770 members. The technical inspiration behind UPnP was to provide a distributed computing framework based on web technologies for small networks, especially home networks.

The Digital Living Network Alliance (DLNA) is a consortium of over 240 consumer electronics companies whose goal is to develop the standards needed for interoperable networked products for Digital Homes. DLNA specifications are based primarily on UPnP.

UPnP specifications can be divided into: Architecture and Profiles. The UPnP Device Architecture (UDA) is the core upon which the device and service specifications are built. The UDA defines two types of hosts: control points (which are clients) and devices. Devices may include “services”. Devices may be embedded within other devices. The architecture has six main features:

1. Addressing – IPv4 and IPv6 auto-configuration protocols
2. Discovery – SSDP (Simple Service Discovery Protocol)
3. Description – XML, data types, device and service descriptions
4. Control – SOAP (Simple Object Access Protocol)
5. Events – GENA (General Event Notification Architecture)
6. Presentation – HTML and vendor extensions

UPnP technology has very strong roots in standard Web technologies. It is based on IP, HTTP, Web browsing, XML and SOAP. The UPnP discovery protocol, SSDP, reuses HTTP headers. GENA is a “publish and subscribe” protocol based on HTTP. Though UPnP data types are derived from XML Schema data types, UPnP descriptions were not originally based on WSDL. This specification did not exist when UPnP was created. However, efforts are underway to encourage eventual convergence.

UPnP device and service profiles must conform to the mandatory behavior of a UPNP device or service. However, UPnP Device and Service profiles are not limited to the technology specified in the UDA. Other protocols, codecs, etc. may be specified as optional or vendor extensions.

Several UPnP device and service profiles have been defined including:

Internet Gateway Device (IGD) V1.0
MediaServer V1.0 and MediaRenderer V1.0
Printer Device and Print Basic Service V1.0
WLAN Access Point Device V1.0
Device Security V1.0 and Security Console V 1.0
Lighting Controls V1.0
Scanner (External Activity V1.0, Feeder V1.0, Scan V1.0, Scanner V1.0)
Printer Enhanced V1.0
Quality of Service V1.0
Basic Device V1.0
Digital Security Camera V1.0
HVAC V1.0
Remote UI Client and Server V1.0

4. LESSONS LEARNED

We have gained valuable experience during the last couple of years of study and working with other companies to use UPnP and related technologies to develop technologies for smart homes. In this section we try to summarize the most important lessons.

It is worth mentioning again that smart homes and small offices are often managed by consumers with little or no training. These networks are often deployed incrementally – one product at a time. And people other than the home owners will occasionally need to be able to use the products.

Whenever possible, home networks and applications should auto-configure or self-organize. UPnP is based on IETF Zeroconf protocols. This is often, but not always sufficient. For example, consider the case of a visitor who would like to be able to change the channel of the TV - but first the visitor has the determine which of several, visible network access points belongs to his or her host's home network.

Another recurring problem for consumers is that even after they finally get everything working, eventually something will fail. The failure might be due a power failure, or the network operator may be having problems or the home product may stop working. In any case, the burden is on the consumer to determine the nature of the problem and the best course to take to correct the problem. The lack of support for debugging or introspection in distributed toolkits is a serious problem.

Home networks change over time and CEA companies are often loath to sell products that are dependent on products from other companies. These two different pressures have led us to avoid unnecessary dependency on specialized infrastructure. Whenever possible, UPnP uses peer-to-peer protocols. One of the complaints about UPnP protocols, such as SSDP, is their reliance on multicast. Whatever the disadvantages in terms of "chattiness" or scalability, multicast is a useful way to build ad hoc protocols in small networks.

The ability to browse or discover available devices and services is am important part of self-organizing networks, but it is also an important part of the user experience. Unfortunately no service discovery protocol has yet proven itself to be suitable in the home, the enterprise and the public Internet. Today, there are many different discovery protocols and they all use different wire protocols, description languages and query languages. There is probably no single, simple protocol for discovery that would be suitable for all types and sizes of networks, but it would be very valuable if we could standardize on description language and query language.

SSDP Descriptions are rich in meta-data. Like all service discovery protocols, SSDP provides enough information for client to determine what services are available and what protocols should be used. SSDP descriptions may also include all sorts of other information like product name, vendor ID, etc.

There is a natural temptation to add all sorts of other metadata to device and service descriptions such as current physical location, owner assigned – human friendly name, etc. so that this information can be discovered using the discovery protocol. However, some meta-data (such as current physical location) may be quite volatile and most discovery protocols are not optimized for volatile data. In our experience it seems clear that there benefit to being able to advertise and discover volatile meta-data, but more work on discovery protocol support is needed.

UPnP Presentation is an interesting innovation. Descriptions may include a reference to a Presentation URL that is usually a web page. In this way, the user interface to a device can be discovered using SSDP. Support for HTML 4.0 is mandatory but Presentation pages are allowed to include optional, vendor specific extensions such as VBasic. The clear separation of the programmatic service APIs and the UI component by the description makes it simple for programs or users to use SSDP to find what they need.

Surprisingly, UPnP Presentation pages are not a commonly used feature. There seem to be business and technical reasons. The UI of a product is very important to CE companies. Anything that unduly limits their ability to differentiate themselves will not be popular and the standard browsing model is somewhat limited (though Ajax is a move in an interesting direction). On the other hand, the cost of each company crafting unique yet interoperable control points by traditional methods has its own problems. Within the UPnP Forum these problems motivated the creation of the Remote IO working group and specifications. It remains to be seen if the Remote IO service will be successful.

The original UPnP specifications did not include any security features. It was assumed that any needed security would be provided by other means, such as 802.11 WEP, NATs, firewalls and VPNs. This lack of standard authentication, authorization and secure communication mechanisms is perhaps the most important missing feature in the UPNP architecture. The first generation of products that shipped with UPnP, frequently had UPnP disabled by security conscious users. In some cases, the vendors shipped their products with UPnP disabled by default due, in part, to security concerns.

Considerable effort has been invested to design security mechanisms that can be deployed and managed by consumers (i.e., users with no particular understanding of networks or network security). The work shows promise, but it is still too early to say if the current approaches will be successful in practice.

UPnP reuses many ideas from the Web; however, interactions between control points (clients) and devices may be significantly different from the interactions of Web browsers and servers. UPnP (and we predict) the ubiquitous computing web, are not limited to retrieving web content. Interactions with devices are NOT idempotent and they will often have side effects. Mobile devices do not always have the same IP address, they do not always attach to the same network. What URI should be used to reference resources hosted on such a mobile device? UPnP takes a very simple and pragmatic approach to these issues. Device and service profiles are discouraged from designing stateful protocols, but it is not prohibited. Use of hostnames in URIs is discouraged. Devices must be prepared to modify the URIs of content and descriptions. We predict the “ubiquitous web” will have to content with the same issues.

Some UPnP interactions may include multiple devices and services. For example, when a user wants to set up and control sessions between multiple data sources and sinks. UPnP allows streams and other session oriented protocols to be added, when needed, to specific device or service profiles. It is interesting to note that there has been no compelling need to add to the core UPnP architecture, protocols with stronger partial failure semantics, complex session or transaction semantics. HTTP and SOAP are perceived as sufficient. It is also interesting to note that even though SOAP on top of HTTP is

very inefficient; neither the cost of its implementation or its performance is perceived as a significant problem.

The last point is that Web browsers would seem to be an excellent means to provide a user interface to UPnP products. In practice, this is rarely done. Standard browsers are not a very appealing application platform because of the difficulty of adding protocols, rich UIs, sophisticated application logic, distributed events and other distributed services to different browsers running on the same OS (much less different browsers on different OSes). Java, AJAX, etc. are all interesting technologies that may eventually lead to good browsers that can act as more flexible distributed computing platforms.

5.ACKNOWLEDGEMENTS

I am very grateful for the assistance and support of my colleagues in the Nokia Research Pervasive Computing Group. They contributed significantly to the ideas in this paper.

6.REFERENCES

[WSDL] http://www.w3.org/TR/wsdl, “Web Services Description Language (WSDL) 1.1”, Erik Christensen, Fransico Curbera, Greg Meredith, Sanjiva Weerawarana, WC3 Note, March 15, 2001

[UPNP] Contributing Members of the UPnP™ Forum. (2000, June 8), “Universal Plug and Play Device Architecture”, http://www.upnp.org/download/UPnPDA10_20000613.htm

[DHCP] http://www.ietf.org/rfc/rfc2131.txt, “Dynamic Host Configuration Protocol”, D. Roms, March 1997, IETF RFC 2131

[V4] http://www.ietf.org/rfc/rfc3927.txt, “Dynamic Configuration of IPv4 Link-Local Addresses, S. Cheshire, B. Aboba, E. Guttman

[SOAP] http://www.w3.org/TR/soap/, Various SOAP specifications…

[SCHE1] “*XML Schema Part 1: Structures* “,Henry S. Thompson, David Beech, Murray Maloney, Noah Mendelsohn, http://www.w3.org/TR/xmlschema-1/

[SCHE2] “*XML Schema Part 2: Datatypes*”, Paul V. Biron, Ashok Malhotra, http://www.w3.org/TR/xmlschema-1/