

Providing Printing Web Services

Andrei Oliveira da Silva, Felipe Rech Meneguzzi, Paulo Henrique de Souza Schneider, Rodrigo Borges Barcelos, Vicente Oberto Rodrigues, Daniel Waldman, and Ana Cristina Benso da Silva.

CPSE – Research Center on Embedded Software

Av. Ipiranga, 6681 - Prédio 30 - Bloco 4 – Sala 242
90619-900 Porto Alegre, Brazil

Abstract— Mobile devices are becoming more and more powerful and even able to connect to wireless networks. Therefore, if they have network access they can also access services available in the Internet or in its local wireless network. As these devices are becoming popular and useful for domestic and corporate users, sets of services were and can be created to help these users in their daily tasks. This paper introduces some concepts about web services and the proposed technologies for their implementation. The ongoing research deals with the creation of web services to offer printing services on wireless networks. The research is supported by the cooperation project between the Pontifícia Universidade Católica do Rio Grande do Sul and Hewlett-Packard Company.

Index Terms— pervasive computing, embedded systems, UPnP web services, wireless networks, printing services.

I. INTRODUCTION

Web services are definitely a tendency inside the business world. Just as people want to sell their products online, commonly known as e-business, companies want to make their services available to be easily accessed by network connected clients. These web-shared services are called web services.

The vision of having all devices connected in a wireless network and people accessing their homes and exchanging multimedia files through their PDAs is each day coming closer to reality. Cooltown [1] is the code name of Hewlett-Packard's project to implement this vision as actual devices.

The use of this kind of application would be interesting in many fields and activities, such as art galleries, where it might be possible to request a copy of a picture by the PDA and directly print it, and to the common citizen looking for information about his wife and newborn baby through his PDA. Also, the medical area can be benefited, where a doctor can remotely verify the health status of his patient.

This work was supported by the HP-Brazil/ PUCRS (CPSE – Centro de Pesquisa em Software Embarcado).

The authors are with the Centro de Pesquisa em Software Embarcado (CPSE), Porto Alegre, RS, Brazil 90619-900 (e-mail: {andrei, felipe, paulo, borges, vicente, waldman}@cpts.pucrs.br).

A. Benso is with Faculdade de Informática, Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Porto Alegre, RS, Brazil 90619-900 (e-mail: benso@inf.pucrs.br).

The web services model is about to bring all this comfort and facilities to life. The involved research lab has been studying, testing and evaluating the integration between several technologies and creating examples of such a web services.

The prototypes development and experiments are built over an HP platform created specifically to supply Java technology to embedded systems, called Chai. Although many of the studied technologies do not have an appropriate documentation, references or examples, the group has to adapt and create some libraries to reach the proposed goal. Therefore, the research of new tools and the creation of new Chai libraries are parallel tasks in the project.

The desired scenario of this prototype is to have an environment where several devices are connected through wireless networks and all of them have to ability to access web service resources without the need of manual configuration or software installation.

This work presents a web service prototype that supplies printing services on a wireless and pervasive environment. The document is organized as follows. Section two presents the technology overview for web services and distributed computing. Section three shows the project context. Sections four and five presents, respectively, the prototype development and results, and our conclusions and future work.

II. INTRODUCTION TO WEB SERVICES

The concept of web services is new, and its capabilities were not yet fully explored. The proposed work will try to make this area more useful and popular.

Web services are, basically, services available in the Internet. Thus, instead of installing the required software, people will look for the services that will act in the role of a local, installed, program and, remotely, will make use of its benefits.

Since mobile devices have small storage space and, according to the advance of the wireless and broadband communication, web services might have a boost in popularity since the cost of remote access is significantly lower than it was some years ago, and the current perspective is that this cost will continue to decrease.

The proposed application will let a PDA identify printers in

a wireless network, and allow it to print a file in a chosen printer. This web service is based in the lookup and print services. The result can be seen as a mobile dynamic device driver, so the user does not need to have a printer driver installed in his device, since a lookup service will automatically find the printers available in the network. This scenario will easily allow the user to choose the printer he wants to use without the need of manual configuration.

A future approach is to provide a custom lookup service for each available network service, not only for printing, but also for fax or telephony services.

A. *Support Technologies*

To reach our goals, some technologies were studied and others are under research to improve its performance or explore most of its features. In this subsection it will be presented a brief explanation about the various technologies presented to support web services. These technologies are the following:

1) *WSDL*

The Web Service Description Language [11] provides a straightforward mechanism to describe web services. It is based in XML (Extended Meta Language) [13], where a file stores the service information. It allows the verification of service properties before its use, or during its execution, to gain information about what the client needs.

When the application is fully lookup service dependent, the description of the service will let the user decide which of the found service is the best for the application.

This technology was not used in the current prototype. However, in a future approach, it can be used to let the lookup service filter the network-collected results, so the client could query only color printers, for example.

2) *SOAP*

The Simple Object Access Protocol [4] is one of the most used protocols for data exchange, and it uses XML to describe the information that will be sent through the network. This technology has been used by web services and data base applications. The lightweight and simplicity are his principal virtues.

A SOAP message can be transmitted over any protocol, but it is usually sent inside HTTP (Hypertext Transfer Protocol) messages. It allows RPC (Remote Procedure Call) method calls and to determine a path from where it will pass through.

UPnP [2] uses SOAP messages to lookup services and other protocols also use it to exchange information.

3) *UPNP*

UPnP inherits the same idea of PnP (Plug and Play) but it is more generic since it identifies any device connected in the network. This technology is not as simple as described above, and there are few points that have to be explained.

The Universal Plug and Play works with two components, the control point, which acts in the role of the client, and the device hosts, which are the servers.

The functionality of an UPnP network acts in the following schema: a device wants to make its services available, so it will instantiate a device host and announce itself in the network. For example, there is a television that wants to be available to be accessed by any mobile gadget, so it must be a device host and announce itself in the network to be found by any control point.

The control point is the client that will look for a service, in this case an announced device host. A control point example can be seen as a PDA looking for a television to watch his favorite cartoon.

This technology uses HTTP, SOAP, XML and SSDP (Simple Service Discovery Protocol) [2] for data exchange and service discovery.

4) *Other Technologies*

Other web service technologies were studied and discarded and some are still under study. The replacement of technologies and its comparison is essential in the scope of the project to decide which one is better for embedded systems.

Between the discarded technologies, UDDI is one of them and it was not used because its scope is very large, the Internet, and this makes it out of context for wireless local network on embedded systems.

The authentication is also under study, but was not yet incorporated to the project. LDAP (Lightweight Directory Access Protocol) [10] was the researched technology to act in this role, and a future implementation should use it.

5) *Considerations*

The huge variety of options to implement the prototype and its recent creation may generate some confusion when the question is the technology study and decision of which one will be used.

To start up the implementation, the chosen technologies were basically SOAP, UPnP and WSDL, but another ones will be tested after the release of an effective application with these tools.

III. CASE STUDY

The prototype described below was fully developed inside CPSE lab. The CPSE (Research Center on Embedded Software) is a joint research project involving Hewlett-Packard R&D Lab and the Faculty of Informatics (FACIN) [7] at PUCRS. It is focused on the development of applications based in pervasive computing inside the context of web services in a wireless network with mobile devices.

To implement this type of application, the project has an infrastructure with one HP Jornada Pocket PC with a Bluetooth [9] card, one HP Laserjet 4600 printer and a Bluetooth and IEEE 802.11b wireless network.

IV. PRINTER WEB SERVICE

A. Overview

The focus of the project team is to develop embedded software for mobile devices and printers. The current endeavor in which the group is involved in is to develop an application within this scope, which is to build a print service that can be located automatically in the network and used without the need for configuration by the user.

The main idea of the prototype is to use the UPnP discovery protocol to search for printers available in a wireless network and send a URL to be printed. This is the focus of the CPSE research, integrating the HP's web service technology, namely HP ChaiServer, and emerging tools, such as UPnP or SOAP.

The Figure below describes the environment and the behavior of the presented prototype.

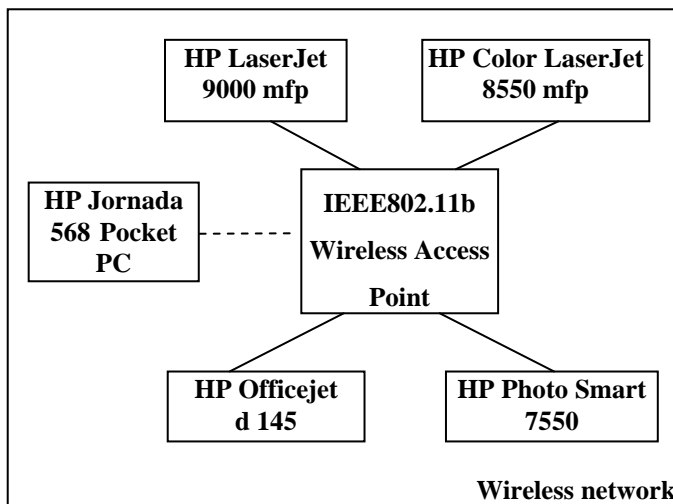


Figure 1. A Pocket PC getting in a wireless network, where there are four printers available

Figure 1 shows a scenario of how the environment can be built. There are four printers, with distinct features, connected by and wireless access point and a pocket PC with a Bluetooth card will get in this network and look for a printer with the desired features to print a picture, for example.

The UPnP control point will be placed in the Jornada, and each printer will have a print ChaiService announced by its own UPnP device host. Each printer's ChaiService homepage will present the features that the device offers. For example HP LaserJet 9000 mfp only prints black & white documents, in the other hand the Color LaserJet 8550 mfp allows printing color pages, as the OfficeJet has more features than the printing, such as fax, and the Photo Smart allows high quality photo printing.

B. Underlying Technologies

The executing platform is the HP Chai, which is a Java Virtual Machine for embedded systems. This Java platform is being deployed in several HP products, such as the LaserJet 4600, and the Jornada Pocket PC.

Another technology used in the implementation of the print

web service is the ChaiServer, which is HP's web services provider. ChaiServer integrates an embedded web server implemented in Java and runs over the Chai VM. The web server part of ChaiServer is intended to provide an interface that can be accessed by a simple web browser. This tool provides the infrastructure for web service applications called Chailets, or ChaiServices. The name Chailet is inspired in the dynamic HTML interface creation inherent to Java servlets.

When a ChaiServer is started, it opens a connection port to stand by for HTTP requests, much like an HTTP server does. This interface starts up displaying the options regarding the Loader Chailet, which is responsible for loading new Chai Services into the corresponding ChaiServer.

The last technology used in the implementation of the prototypes is the UPnP protocol, whose dynamic discovery protocol should be integrated in the Chai Services so that these can be detected automatically in a network. The prototype implementation used Siemens's UPNP Stack implementation [5]. This tool provides a Java API for the creation of UPNP devices and control points in a low size package, what makes it interesting to embedded systems. Also, it abides to the limitations in Java code executable by Chai due to its embedded nature.

C. Implementation

The project has developed essentially two ChaiServices with very distinct objectives: one to provide a URL printing service, and the other to announce the ChaiServer as a UPNP device.

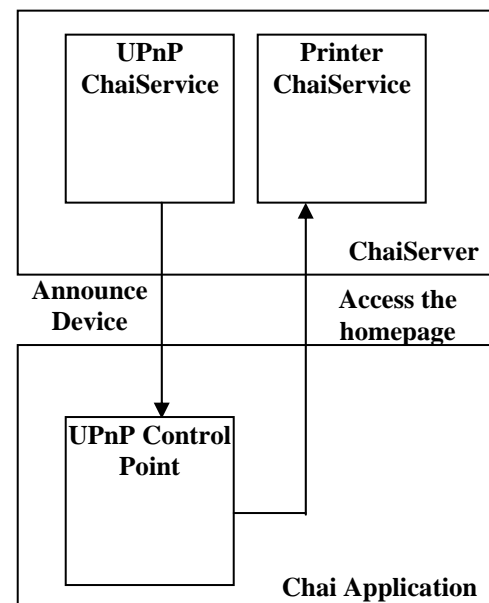


Figure 2. Prototype functionality diagram

The printing service's main implementation objective was to provide a printing interface in a network in such a way that no configuration is necessary prior to send the printing commands. The driver-level printing functionality was not the main focus of the service implementation, therefore an external printing program is called by the Chai service instead

of using some kind of printer driver implementation.

It was used in the prototype a printing tool called HTMLPrint 2.02, which is a trial version distributed by Bersoft [14].

The objective of the print job is to call the HTMLPrint process, and pass the URL as a parameter. The restriction to make it run correctly is to have the HTMLPrint executable file in the same directory the ChaiServer is running.

It was implemented only the UPnP device in the first release of the prototype, because the construction of a control point would be more complex task. To test the functionality of the device, a default control point was used.

To implement the UPnP device component it was built a class that instantiates the Siemens UPnP package device and calls its methods, such as to announce or remove device hosts, and it communicates the Chailet HTML interface.

The development of Chailets was made using templates provided by the ChaiServer, such as an echo to a string. Since the ChaiServices are written in Java, the Chai environment is portable to any platform with JDK 1.1 or later. So the development of applications seems pretty familiar.

The ChaiServer works as an HTTP server that receives a GET or POST method referenced to each Chailet homepage and with the received action it executes the application. This transaction is transparent to the user, because the browser does it by itself.

The required action and only restriction to run the UPnP ChaiService is to set the classpath to the Siemens package and the Bersoft tool in the print server machine.

D. Results

The first release of the prototype was released as the Figure 2 presents it. There is a pair of ChaiServices implemented with an HTML interfaces, as all other ChaiServer applications.

It is possible to announce a device through the UPnP Chailet, in this case it is announced the own ChaiServer homepage where the Printer Chailet is installed.

The test was done with the ChaiServer installed in a computer that acts in the role of a print server.



Figure 3 UPnP Chai Service user interface

In the Figure 3 it can be seen that the HTML graphical interface of the UPnP Chailet, and Figure 4 shows the print

Chailet interface.



Figure 4 Printer Chai Service user interface

V. CONCLUSION & FUTURE WORK

The experiments with the described technologies generated not only prototype evaluation versions but also knowledge to incorporate other features to the project or even another technologies.

The expected approaches will bring more features to the final application. Between the future implementation tools, the dynamic printer driver will be done with the built of the control point identifying the available printers in the network.

Another previewed improvement is the announcement of each ChaiService by itself by the use of the UPnP features. There is already an HP developed application based in the discovery of all ChaiServices available in the local network, but it is based in the SLP (Service Lookup Protocol), which is an IETF [15] draft for locating services, but it is not yet an IETF standard.

What can add virtues to the project is the creation of an UPnP implementation, or any other discovery protocol including more features to the Chai library. To reach this goal, there is a group in the project studying the implementation of SOAP and XML packages, which is the tendency in the lookup or discovery services.

The projects works with prototypes, each time a new technology is discovered or fully understood, it is generated an application to test the integration with the Chai appliance platform.

It is previewed to future releases the Chailets UPnP announcement ChaiService, the Chai control point implementation and a SOAP based discovery service protocol until the end of this year.

REFERENCES

- [1] P. Cooltown, Cooltown. HP United States Available: <http://www.cooltown.hp.com/>.
- [2] UPnP, Universal Plug and Play Forum. Microsoft United States Available: <http://www.UPnP.org/>.
- [3] UDDI, Universal Description, Discovery and Integration of Business for the Web. Available: <http://www.uddi.org/>.
- [4] SOAP, Simple Object Access Protocol, World Wide Web Consortium (W3C). SOAP Version 1.2. Available: <http://www.w3.org/TR/2001/WD-soap12-20010709/>.
- [5] UPnP, Siemens UPnP Stack (Java Version) Programming Guide. Siemens Aktiengesellschaft, 2001.

- [6] D. Livingston, Advanced SOAP for Web Development, Upper Saddle River, NJ: Prentice Hall PTR, 2002.
- [7] FACIN, Faculdade de informática, Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS). Available: <http://www.inf.pucrs.br/>
- [8] M. Miller, Discovering Bluetooth, Sybex Inc; Alameda, CA 2001
- [9] Bluetooth, The Official Website, Bluetooth. Available: <http://www.bluetooth.com/>.
- [10] LDAP, Lightweight Directory Access Protocol, IBM Corporation, International Technical Support Organization. Understanding LDAP. First Edition, 1998.
- [11] WSDL, Web Services Description Language Version 1.1. World Wide Web Consortium (W3C). Available: <http://www.w3.org/TR/wsdl>.
- [12] Y. Shohoud, Introduction to Web Services. Available: <http://www.vbws.com/book/Chapter1/>.
- [13] M. Morrison et al. XML Unleashed. Indianapolis, Indiana: Sams Publishing, 1999.
- [14] HTML Print, Bersoft Inc. Available: <http://bersoft.com/>
- [15] IETF, Internet Engineering Task Force. Available: <http://www.ietf.org/>