Ametsa: a Generic Home Control System Based on UPnP

M.T. SEGARRA, A. THEPAUT, R. KERYELL, J. POICHET, A. PLAZAOLA, B. PECCATTE
GET - ENST Bretagne
Technopôle Brest Iroise CS 83818 -- 29238 Brest Cedex 3 – France

Abstract. Home services are gaining more and more interest in the computer science community. In particular, several service discovery systems (SDS) have been proposed that assume the presence of devices interpreting standard and open Internet protocols. However, the lack of home devices understanding such protocols has restricted the proliferation of control applications at home and the utilization of such systems is limited to computer science environments. The "Maison Intelligente" project aims at proposing generic services to control home devices based on SDS in order to ensure the independence from devices manufacturers. We have first developed the Ametsa service which exploits the UPnP SDS in order to be aware of arrival or removal of devices and to send or receive commands to/from available devices. We also propose gateways to connect UPnP to some other protocols. Some experiments have already been performed based on a prototype consisting on Ametsa and two gateways allowing the control of X2D and X10 devices.

Introduction

Traditional home services are proposed by home devices manufacturers by means of a proprietary control device which may be accessed either directly or from the phone network. Services allowing the (remote) control over lights, heating or shutters are usually proposed and implemented by proprietary means.

On the other hand, home services are gaining more and more interest in the computer science community. In particular, several service discovery systems have been proposed ([1, 2, 3]) that use open and standard Internet communication protocols in order to discover available devices and to send commands to them. Although these solutions are independent of device manufacturers, the lack of home devices understanding Internet protocols has restricted the proliferation of control applications at home and the use of such systems is limited to computer science environments.

The "Maison Intelligente" project aims at proposing generic services well-suited to impaired people in order to improve their autonomy at home. In this project, our work relates to the construction of the software system platform upon which home services are provided. In order to ensure our independence from home devices manufacturers, we have based our platform on a service discovery system called UPnP (Universal Plug and Play) [1]. UPnP functionality is used to be aware of arrival or removal of devices and to send or receive commands to/from available devices. As many home devices are not able to manage Internet protocols, we also propose gateways that translate UPnP protocols on the corresponding (proprietary) protocol.

We have built a first prototype of our platform consisting on a control application that centralizes information about available devices at home, and two gateways. The

control application, called Ametsa, can be accessed via a web interface allowing the user to send commands to home devices by using the UPnP system. Gateways use UPnP fonctionality in order to receive commands from Ametsa. They allow the management of two frequently used powerline protocols, X2D [9] and X10 [10].

The paper is organized as follows. Next Section introduces the "Maison Intelligente" project, its origins and goals. Section 2 presents the UPnP technology which is the basis of our work in the project. On top of this technology we have developed the generic control point Ametsa which is described in Section 3. Section 4 presents the architecture of the X2D and X10 gateways. Finally, some concluding remarks are presented in Section 5.

1. The “Maison Intelligente” Project

1.1.Project Goals

Work on the "Maison Intelligente" project started in January 2001. The three following schools, INT Evry ENST Bretagne and ENST Paris took part in this project. The first step was to carry out several enquiries towards professionals and visual or motor disable people. The results of these works led us to the following measures:

- first of all, it appeared to us that it was necessary to improve the service specifications provided by domestic networks to disable people through the analysis of the questionnaires,
- then, our objective was not only to solve the problem linked to the possibility to keep dependent people at home, but also to maintain permanent contacts with their relatives, medical entities or first aid services through a process of continuous reassuring guarantee,
- and finally, we needed to improve the knowledge linked to the evolution of information technologies applied to the smart house booming market.

The original aspect of this project also lies in the fact that in addition to the pluridisciplinary team, many partners from medical fields (Raymond Poincaré hospital for example), specialists in blind people reinsertion (SIADV, Service Interrégional d'Appui Régional aux Adultes Déficients Visuel), users' representative associations such as AFM (Association Française contre les Myopathies) also participated to the process. From the very beginning, we took into account the users' specificities. This allowed us to have a good knowledge of their medical, technical, social and economic needs and uses. It also gives good indications regarding the development of services (domotic, mobile phone Internet services) and specificities of the product such as network gateway. Together with this process, services and products are adapted according to the results and wishes expressed by the users.

1.2.Generic Tools for Home Services

Home services take profit of new technological developments in communication networks. These services are not only dedicated to a large public of users but also to specific users such as dependent people (elderly people). These services aim at improving their ability to evolute in their domestic environment and compensate their handicap. Today, as a consequence, many products are available on the market allowing the control of the heating, lightning or automatic shutter systems in a house through a cabled or cordless communication network. Occasionally, it is possible to access the control panel from

outside the house, mainly through the telephone network. This allows to set up services such as telemonitoring.

In the computer sciences domain, the concept of services provided in houses raised the interest of many protagonists. In opposition to domestic devices builders, they consider that domestic equipment able to understand the Internet communication protocols already exist and propose tools to assist the user in the process of discovery and utilization of the systems. Such opened and standard systems are independent from domestic devices builders. However, the low development of such domestic devices has slow down the development of house controlling applications (as far as we know, up to now no application exist). This situation limits the utilization of such systems to computers.

In the "Maison Intelligente » project, we propose to develop generic tools dedicated to the creation of services to control the domestic environment. These tools are based on service discovery systems. These tools are built according to standard protocols, therefore, the independence of device constructors is guaranteed.

2. UPnP Technology

UPnP (Universal Plug and Play) [1] refers to a specification that defines an architecture for pervasive peer-to-peer network connectivity of intelligent applicances, wireless devices, and PCs of all form factors. More precisely, it defines general interoperability mechanisms that enable self-configuring devices to create an ad-hoc, self-discovering system of interoperable network devices by specifying means for automatic address configuration, device discovery, command/control, and eventing. It is designed to bring easy-to-use, flexible, standards-based connectivity to ad-hoc or unmanaged networks either at home, in a small office, or attached to the Internet.

2.1.UPnP Architecture

UPnP technology uses existing Internet standards including TCP/IP, HTTP, SOAP [4], XML [5], etc. These open standards provide the communication infrastructure of the UPnP architecture. Although the UPnP architecture consists of a peer-to-peer network, nodes on the network communicate with each other in a client-server manner. Clients are called Control Points (CP) and typically provide a User Interface (UI) for end-users. Servers are called Controlled Devices (henceforth, called devices) and by definition expose a well-defined set of functions called *actions*. In all cases, CPs invoke actions, and devices respond to the actions that are received.

Within the UPnP architecture, device functionality is exposed using a set of *services*, each of which corresponds to a functional component of the device. Each service defines a set of *state variables* and *actions* that allow CPs to obtain the current state of the device and to control the device's operation. Invoking an action usually causes a change in the internal state of the device that would affect the value of certain state variables. As an example, a video recorder may have a tuner and a clock service. The last one may have a `current_time` variable and two actions (`set_time` and `get_time`) that allow to control the service and that act on the `time` variable.

A device may also be composed of several *embedded devices* and in this case the term *root device* is used. For the user point of view an embedded device behalfs as a root device, i.e. it may offer several services that may be accessed by a set of actions. However, from the designer point of view, it allows a better structuration of the UPnP network devices. All the components of an UPnP network are shown in Figure 1.

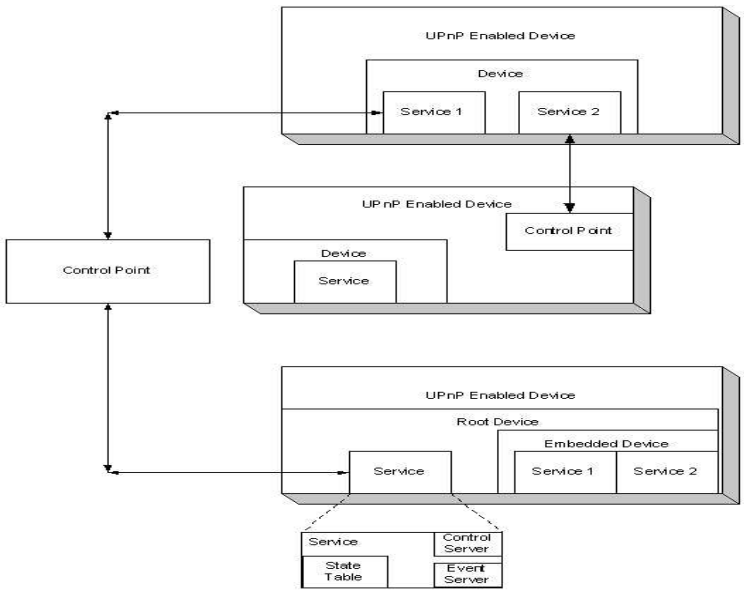

Figure 1 *UPnP Components*

In order to enable autonomous device interoperability, members of the UPnP Forum [1] constructed a set of device and service definitions (templates) which can be used to model various common devices. Since the behavior of these device and service templates is well defined, CPs can interoperate with any device that implements the services that are supported by the CP. In this way, CPs and devices can be built independently by different manufacturers with the assurance that they will interoperate according to the functionality defined by the corresponding UPnP device/service templates.

2.2.UPnP Functionalities

Four functionalities are provided by UpnP, based on open and standard communication protocols such as SOAP (Simple Object Access Protocol), GENA (Generic Event Notification Architecture) [6], and SSDP (Simple Service Discovery Protocol) [7]. All of them use HTTP as the transport protocol upon TCP but also on UDP for multicasting.

2.2.1.Network Addressing

Since the UPnP architecture is built on top of the Internet Protocol (IP), each node in the network requires a unique IP address. This address is assigned either via a Dynamic Host Configuration Protocol (DHCP) server [8] or via the Auto-IP protocol if a DHCP server is not available. When a DHCP service becomes available, all nodes are required to obtain an address from it. Once a device or a CP has got its address assigned, it is considered as "added" to the UPnP network.

2.2.2.Discovery

When a CP is added to the network, it needs to discover (i.e., locate) all the devices in the network that it is able to control. This is accomplished via the Simple Service Discovery

Protocol (SSDP) [7] by broadcasting a discovery request that identifies the functional capabilities that the CP wants to control. Any device that exposes those capabilities responds to the request by identifying itself to the CP.

The device response contains the URL to the "XML device description document," which identifies the services that the device implements, as well as the specific actions and state variables that are supported by each service. By parsing this information, the Control Point is able to determine the exact capabilities of each device. This allows a Control Point to determine if it wants to interact with and control a particular device.

When a new device is added to the network, it may broadcast an identification notification to the network. This notification informs existing CPs that a new device has been added to the network and is available to be controlled. The notification information also includes the URL of the new device's description document, as described above.

2.2.3.Command/Control

Once a CP has determined that it wants to control a particular device, it uses a Simple Object Access Protocol (SOAP) [4] to invoke any of the actions exposed by the device's services. The behavior of each action is well defined by the service template document.

2.2.4.Eventing

As the internal state of a device changes, either in response to an action or via some internal condition, the device can inform one or more CPs of the state change using Generic Event Notification Architecture (GENA) [6]. With this protocol, CPs that desire to be informed of state changes within a particular device must register for that device to receive event notifications. A given device may be monitored by multiple CPs. When an internal state change occurs, the device sends an event notification to each CP that has registered for the device. This event notification includes an identification of the state variable that has changed, along with its new value. The set of state variables that are evented by the device is defined in each of the service description that are supported by the device. Additionally, each evented state variable may be moderated such that rapid changes in that state variable do not cause excessive network traffic.

2.3.UPnP Implementation

At the time of starting the "Maison Intelligente" project, several implementations of UPnP existed that offer all the functionalities of the specifications (Microsoft, Metro Link, Intel). However, only the one developped by Intel was free and available for the Linux operating system. It is a C implementation that makes UPnP functionalities accessible via local method invocation. We have therefore implemented a generic CP, called Ametsa, and two UPnP gateway devices on top of the Intel implementation. Ametsa receives announces from all devices on the UPnP network and is able to send commands to them while gateway devices interface UPnP world with the X2D [9] or X10 [10] world, according to the nature of the device counterpart.

3. The Ametsa Control Point

As previously mentioned, the Ametsa CP is a software built on top of the UPnP implementation that is able to discover devices that it is capable to manage. It benefits from the UPnP technology by calling the API provided by the Intel implementation. Therefore, Ametsa provides methods intended to send an action to a device, to subscribe to a service, to search for a particular device, to be informed from the state of a service, etc. All these functionalities may be accessed directly by shell commands, by remote method invocations, or through a WWW browser. The two first methods are implemented in Ametsa as a plug-in that may be inserted in the Ametsa architecture dynamically. The last one uses the remote method invocation to allow the user to send actions to devices. They are described in the next paragraphs.

3.1.Shell Interface

When an application executes locally to Ametsa, the shell represents a very simple mean to access to its functionalities. The shell offers three types of commands:

- plug-in related commands. They allow to start the execution of a plug-in, to stop a running plug-in, to show its methods or to execute a particular method offered by a plug-in;
- devices related commands. They correspond to the commands allowing to control devices, and to show services and variables state;
- general commands that allow service or device search, to show the type of available devices, etc.

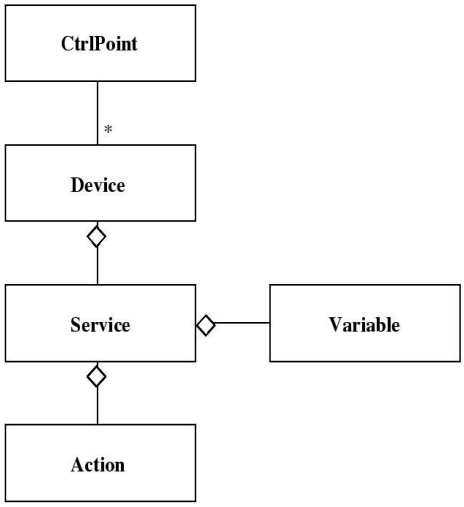

Figure 2 *Class structure*

3.2.Remote Method Invocation

The simplest mean to access Ametsa functionality when executing on a remote site is the remote method invocation. We have implemented this type of access by using the gsoap library [11] that allows for sending and receiving SOAP messages by simple method calls. Two entities provide this type of access in our platform:

- SOAP server. It is designed as an Ametsa plug-in and allows access to all functionalities of Ametsa;
- SOAP client. It is designed as a library that allows remote communication with the SOAP server of Ametsa. It is not simply a set of methods but it offers an object model for the home control structure. Figure 2 shows this model. An instance of the class `CtrlPoint` may manage several `Devices` and a device may in turn be

composed of several `Services` with `Actions` and `Variables`. Methods of these classes allow to control and obtain information from the entities they represent.

3.3.WWW User Interface

The user interface is based on a web navigator that allows the user to be aware of available devices, send actions, etc. It is implemented as a set of PHP [12] and XSLT [13] scripts. The first one uses Ametsa to ask for information on available devices and the XML file describing the gateway (see Section 2.2.2). The latter transforms the XML based description on an HTML page that can be viewed by the user.

4. UPnP Gateways

In order to allow us to control non-UPnP devices, we have developed several gateways that transform UPnP commands sent by Ametsa to the corresponding protocol. We have currently implemented two gateways to manage X2D and X10 devices, respectively. Both have been designed as an UPnP root device (see Section 2.1). Figure 3 shows the architecture of our gateways and their relation with Ametsa.

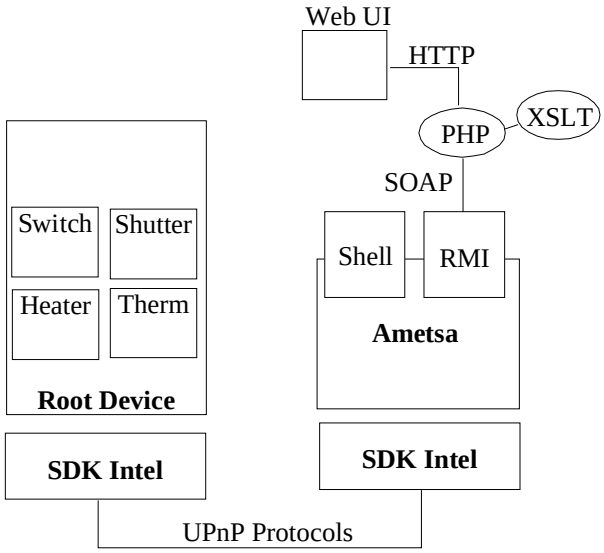

Figure 3 *Architecture of the platform*

The root device maintains the information that is necessary to interface the non-UPnP devices, mainly a unique identifier for each available device. It is composed of a set of embedded devices, one per physical device. It receives actions from Ametsa and submits them to the corresponding embedded device.

Embedded devices use functionality of a library that effectively send commands to physical devices by building the necessary network frames.

5. Concluding Remarks

In this paper, we have presented the first results of our work on the "Maison Intelligente" project. We have proposed the use of UPnP as the basis for building a generic home

control application. Other similar systems exist that offer a service discovery functionality (Rendez-vous [14], SLP [15]) or discovery and control over devices (Jini [3], Salutation [2]) but only UPnP is based on standard, open protocols. Moreover, means to access UPnP devices are provided on the OSGi framework [16] so that they can be controlled over the Internet.

We are currently working on the development of a CORBA [17] plug-in for Ametsa that will provide an object model of the home control architecture. Indeed, as SOAP does not deal with objects, the use of SOAP for remote method invocation implies the re-implementation of an object model on the client side. This increases the footprint of the client side code and makes it difficult the deployment of our solution in a portable way.

We are also working on the adaptation of our system to blind people. By exploiting the results of the TéDéVi project [18] we intend to add a vocal synthesis to our platform so that the current state of available devices if confirmed by a sound.

Acknowledgements. The autors would like to thank the foundation "Louis Leprince-Ringuet" and the GET for their financial contribution to this project 'Smart Home and Independent living for Persons with Disabilities and Elderly People'.

References

[1] UPnP Forum. Home Page. *www.upnp.org*
[2] Salutation Consortium. Home Page. *www.salutation.org*
[3] Jini Community. Home Page. *www.jini.org*
[4] D.Box. "Simple Object Access Protocol (SOAP) 1.1". May 2000 *www.w3c.org/TR/SOAP*
[5] T. Bray. "Extensible Markup Language (XML)". October 2000 *www.w3.org/TR/REC-xml*
[6] J. Cohen. "General Event Notification Architecture Base: Client to Arbiter". September 2000.
[7] Y.Y. Goland. "Simple Service Discovery Protocol/1.0". October 1999
[8] R. Droms . "Dynamic Host Configuration Protocol". March 1997
[9] DeltaDore. Home Page. *www.deltadore.com*
[10] X10 Protocol. Home Page. *www.x10.com*
[11] R. Van Engelen. "Generator Tools for Coding SOAP/XML Web Service and Client Applications in C and C++"
[12] D. Sklar, and A. Trachtenberg. "PHP Cookbook". O'Reilly & Associates. 2002
[13] W3 Consortium. "XSL Transformations (XSLT) Version 1.0". November 1999 *www.w3.org/TR/XSLT*
[14] Rendez-Vous. Apple Computer, Inc. 2003 *www.apple.com/macosx/jaguar/rendezvous.html*
[15] E. Guttman. "Service Location Protocol, Version 2". June 1999 *www.faqs.org/rfcs/rfc2608.html*
[16] Open Services Gateway Initiative. "Specification Overview". *www.osgi.org/resources/spec_overview.asp*
[17] Object Management Group. "CORBA Basics". *www.omg.org/gettingstarted/corbafaq.htm*
[18] A.Thépaut, J.Y. Bouvier. "Projet TéDéVi : Télé-enseignement pour déficients visuels", Cité des Sciences et de l'Industrie. May 2002