

Controlling Appliances with Pen and Paper

Mario Kolberg*, Evan H. Magill*, Michael Wilson*, Peter Burtwistle[†] and Oscar Ohlstenius[†]

*Department of Computing Science and Mathematics

University of Stirling, Stirling FK9 4LA, United Kingdom

Email: {mko,ehm,mew}@cs.stir.ac.uk

[†]Sysnet Ltd., 457 Sauchiehall St, Glasgow G2 3LG, United Kingdom

Email: {ptb,o.ohlstenius}@sysnet.co.uk

Abstract—Networked appliances are consumer or household devices with a network interface which allows them to be connected to a (home) network. Networked appliances offer major benefits compared to their conventional equivalent. Software services can operate upon an array of these appliances to offer value added functionality. In this case the user interface to control the appliances will be inadequate. Commonly networked appliances may either be controlled using the traditional user interface, such as switches and buttons on the device, or alternatively using a PC based interface. Traditional user interfaces often cannot support the full functionality offered by the appliance. The PC based interfaces tend to be too complicated for the average user, nor will a PC based interface be suitable in many situations or indeed be available. Therefore a user interface which supports the new functionalities and is easy to use is required.

In this paper, Anoto functionality enabled pen and paper are suggested as a suitable interface. Pen and paper are well known to all users and suitable paper forms make it straightforward to fully control and program networked appliances without a PC interface. Thus an important barrier to the acceptance of networked appliances is broken. Furthermore, the presented approach can be used to integrate the control of *multiple* appliances and services.

A major challenge to this work has been the integration of different network protocols used in the home. The use of an OSGi gateway has proven a great help to integrate them. This paper discusses an approach to the integration of pen and paper with networked appliances using an OSGi gateway. Because of the use of an OSGi gateway, the approach is not restricted to controlling devices which use a specific single network protocol. Rather, the approach can be used with appliances which communicate using a wide variety of protocols.

I. INTRODUCTION

A. Networked Appliances

Networked appliances are consumer devices which contain a network interface. As such, an increased number of networked appliances may assume a networked home with an always-on Internet connection. Examples include fridges, stoves, lights, curtain drawers (puller), security cameras, stereos, and TVs. Networked appliances are attracting an increasing interest and the first products are now available off-the-shelf (e.g. Mitsubishi NetCommand TVs, VCRs, DVDs). A number of industrial trials have been conducted, such as the OnStar at Home trial launched by the Internet Home Alliance and involving over 70 homes in the Chicago area, the Telia Connected Home initiative in Europe, the Net@Home trial by France Telecom and Thomson, and Philips' HomeLab in Eindhoven, the Netherlands.

Even though some networked appliances offer additional functionality, it is when they are networked that they offer much greater potential. While some appliances may contain embedded software inside the device, the interworking of appliances and their control by external software services create significant added value. So, services can control a number of appliances [1]. For instance, a climate control service may control the heating system, air-conditioning, windows, and air fans in the house. A security service may provide simple burglar alarm functionality and may also record the picture from the entrance camera. If the owners are not at home it may also switch on lights and the stereo at random times to give the impression to passers-by that somebody is at home.

B. Deploying Networked Appliances in the Home

Networked appliances are typically deployed inside the home and are connected to a Local Area Network (LAN). Appliances communicate using various protocols, such as UPnP (Universal Plug and Play) [2], X-10 [3], IEEE 802.11 [4], Bluetooth [5], HAVi (Home Audio/Video Interoperability) [6], and MHP (Multimedia Home Platform) [7].

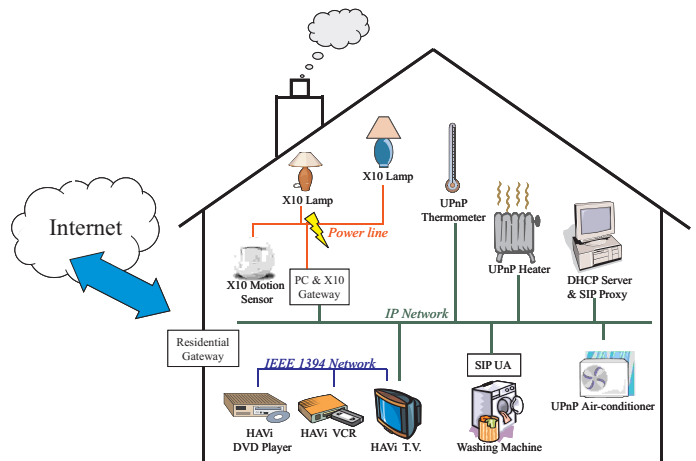


Fig. 1. Appliances in the Home.

The home is connected to the Internet. The two domains are connected together by a Residential Gateway (RGW) with services such as Network Address Translator / Firewall functionalities (Figure 1). These provide protection from unauthorized access into the home and also a translation of

addresses in the IP format to the format used inside the home. Furthermore, the gateways also host the software services which control the appliances. One example of such a gateway is the Open Services Gateway [8–10] standardised by the Open Services Gateway Initiative (OSGi). OSGi is a large industrial consortium (around 40 major companies are members), which aims to deliver a platform for providing services over the lower-level networking standards. OSGi runs on any Java based system, and may be provided in a set-top box, or a standard PC.

C. OSGi

OSGi offers a set of APIs for managing user services, and provides means for device access, configuration management and user management. OSGi is neutral with respect to the services it offers. Rather it is concerned with the (remote) management of those services (installing, starting, stopping and updating) without the need to restart the gateway. Security is a prime concern of the initiative.

The OSGi framework is relatively simple, but offers many possibilities for service providers. However, the interest here is in its ability to control devices regardless of their underlying protocol and to allow their remote control. OSGi uses the concept of bundles which represent either device drivers or services. Services offer some particular functionality to other services or the outside world. Device drivers allow the communication with physical devices. There is one device driver for each supported network protocol, such as UPnP or X10. Typically, devices announce their existence. For example, when an UPnP device is added to the network and announces itself, the UPnP driver within the framework listens to the new device on the network and adds a representation of it to the framework. The newly added device is now available for use by services on the framework. Its appearance is just like another service, albeit with rather low-level functionality. Thus other services may use it via the uniform OSGi service interface. Devices that do not support plug and play can be added manually to the framework, e.g., X10 devices. Consequently, the framework may offer a collection of UPnP, X10 and other devices and (user) services do not need to be aware about the protocol used by a particular device. For instance, a climate control service may control a X10 curtain puller and an UPnP air-conditioning unit, or a security service may be triggered by an X10 motion sensor and control a HAVi VCR to record the picture of burglar. Thus OSGi can be seen as the "glue" which links devices (which support different network protocols) together. The two examples are shown in Figure 2.

One major obstacle in deploying networked appliances and OSGi is the user interface. OSGi uses a typical computer based interface to configure and remotely control the devices and services. This may be a basic command line interface or a graphical interface. Both interfaces are largely inadequate for computer illiterate users and thus are a major barrier to the wider acceptance of the technology. The remaining of the

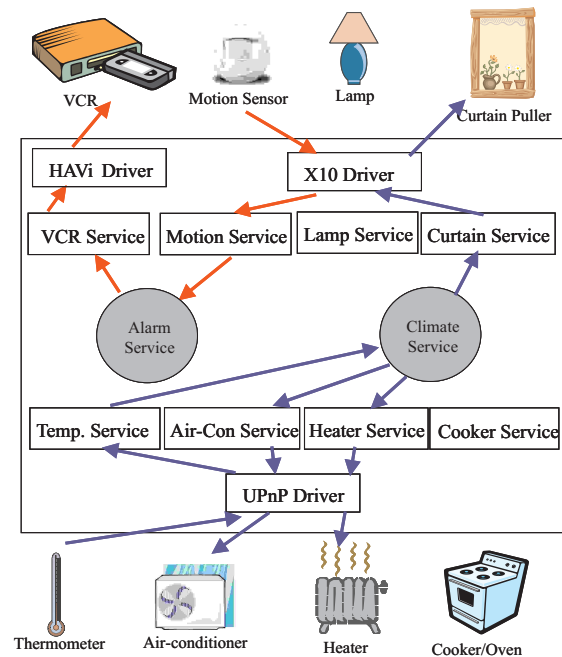


Fig. 2. An OSGi Framework configuration.

paper discusses how Anoto enabled Pen and Paper can be used to solve this issue.

II. USING PEN AND PAPER TO CONTROL APPLIANCES

A. Anoto functionality enabled Pen and Paper

Digital pens and paper enabled with Anoto functionality are an extremely easy to use interface to computer applications. Anoto enabled paper is standard paper with a special pattern printed on it. The dots are 0.3 mm apart. These dots are slightly displaced from a grid structure to form the proprietary Anoto pattern. A minute portion of the total pattern uniquely defines its position in the full pattern, 60 000 000 square kilometers, which is equivalent to an area exceeding that of Europe and Asia combined.

To allow for different applications, a portion of the pattern can be linked to a particular application. Thus the information can be processed in an application specific way. For instance, when writing on a PostIt[®] note, the writing information will be associated with the PostIt[®] application on the computer. Or, the writing in a diary will be associated with a diary application. Even within a page the writing can be located precisely. If the user ticks a check box or adds an appointment for a particular time in their diary, the linked application can use this information. In other words, the application is aware of where the user has been writing on the page. This is very useful for filling in forms.

While the paper with the pattern allows to identify the application, the pen identifies the user. This way the recorded writing in a diary can be associated with the diary application of the pen owner's PC.

The form is titled "APPLICATION CONTROL VIA ANOTO FUNCTIONALITY". At the top, it features the University of Stirling logo, the text "UNIVERSITY OF STIRLING TRANSLATING TECHNOLOGY PROJECT", and the "sysnet" logo. The form is divided into three main sections: "ROOM", "DEVICE", and "ACTION".

ROOM: Includes checkboxes for LIVING ROOM, OFFICE, BEDROOM, KITCHEN, and OTHER, followed by a text input field.

DEVICE: Includes checkboxes for LAMP, FAN, HEATER, AIRCON, and OTHER, followed by a text input field.

ACTION: Includes checkboxes for ON, OFF, and PROGRAM.

Below these sections is a horizontal timeline from 00:00 to 12:00 in 15-minute increments. At the bottom, there are logos for the Department of Computing Science and Mathematics, a signature, and a "SEND" button with a paper plane icon. Vertical text on the left and right edges reads "DESIGNED AND DEVELOPED BY SYSNET. ALL RIGHTS RESERVED. COPYRIGHT © 2007 SYSNET".

Fig. 3. An example paper form to control networked appliances.

When writing with a digital pen on a paper printed with the pattern, digital snapshots of the pattern are taken (about 50 per second). That is, the pen records its movement over the pattern with a tiny camera and stores the information in memory inside the pen. Every snapshot contains enough information to make a calculation of the exact position of the pen. The "intelligence in the paper", derived from the pattern, makes it possible to perform operations by just ticking a box with the pen. Once a page is completed the information is transferred to the mobile phone using Bluetooth, from where it is sent on to the service provider. The service provider processes the information and commonly sends it to its target destination.

This technology offers an ideal interface to control networked appliances in the home. All that is required is a paper form which reflects common appliances in the home. The various rooms of a house are shown with their respective appliances, as are the various actions to control appliances. To switch on the living room lamp, a user simply ticks just three boxes: "living room", "lamp", and "on". The lamp could even be programmed to come on and off at certain times if the form offered a timeline. An example paper form is depicted in Figure 3.

For this approach to work, three stages are required. Firstly, after selecting the desired action on the form the data is transmitted to a service provider hosting an application which will process the pen data and translate it into a paper independent data format (XML). Finally, this information is then sent to a service in the OSGi gateway of the pen owner's home. The pen service locates the desired appliance in the home and triggers the wanted action. Figure 4 depicts this process.

This approach allows a user to control any appliances in his home from anywhere in the world! Using the pen and paper interface fully hides the computation and processing of information by computers from the user. As far as the user is concerned no computer use is involved and all they need to do is to tick some boxes on a paper form.

The paper form and the service provider application can be created using the standard Anoto development kits. In the next section the integration with OSGi frameworks is discussed.

B. Integrating the Pen Interface with OSGi

An XML language has been defined as the interface between the service provider application and the service inside the OSGi framework. In the following this service is referred to as "pen service" (see Figure 4).

The XML reflects the data on the paper form. Figure 5 shows two such (prototype) XML specifications. The first one specifies that the fan and air conditioning in both the office and kitchen are to be switched on immediately. The second example illustrates the possibility of programming a device. Here the lamp in the living room is to be switched on at 6pm and to be switched off again at 11:30pm.

The pen service processes the message, extracting the required device information, i.e., device, location, and actions. After the message has been received and parsed by the pen service, it consults the "device location service" to find references for the devices the user wishes to control (Step 4 in Fig. 4). In the first example in Figure 5 four different devices are named, a fan in the office, a fan in the kitchen, an air-conditioner in the office, and an air-conditioner in the kitchen.

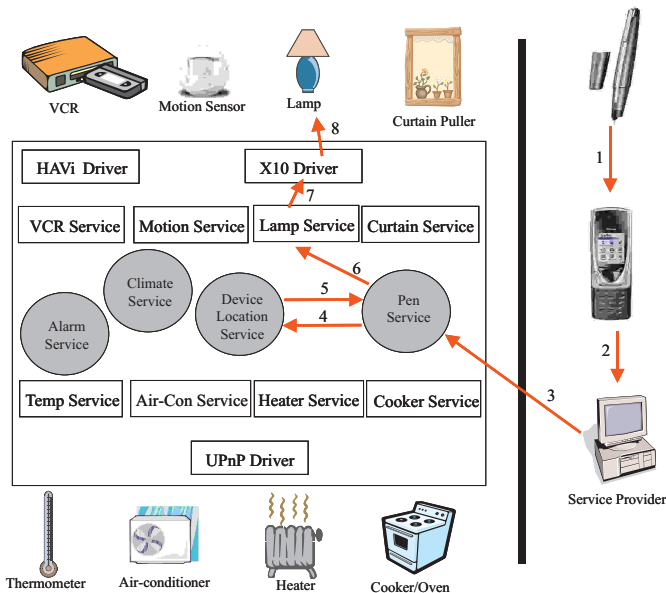


Fig. 4. Message Flow for Controlling an Appliance with an Anoto Pen.

```

<?xml version="1.0" ?>
<ApplianceControl>
  <Room>office</Room>
  <Room>kitchen</Room>
  <Device>fan</Device>
  <Device>aircon</Device>
  <Actions/>
  <ON/>
</Actions>
</ApplianceControl>

<?xml version="1.0" ?>
<ApplianceControl>
  <Room>livingroom</Room>
  <Device>lamp</Device>
  <Actions/>
  <ON when=1800/>
  <OFF when=2330/>
  </Actions>
</ApplianceControl>

```

Fig. 5. XML received by the Service inside the OSGi framework.

The device location service keeps a database about the location of all devices registered with the framework. Device location refers to the physical location of the appliance within the home (e.g. a particular room). The database is updated when a device is removed or added to the framework. Some protocols carry location information (e.g. X10, although provided manually during installation of the device). For other protocols, the information about the location is added manually, by the service provider. This lack of automation is not important as this information only needs to be supplied once because household appliances are not frequently moved around.

If the location service finds entries in its database that match the parameters, it returns references to the corresponding devices to the pen service (Step 5 in Fig. 4). These references point at the device representations inside the framework. With these references, the pen service can then carry out the operation on each of the devices. The instruction is initially sent to the device representation (Step 6 in Fig. 4), from where it is passed on to the driver of the protocol the physical device is using (Step 7 in Fig. 4), and finally across the home network to the physical device (Step 8 in Fig. 4).

If the device is to be programmed as in the second XML

example in Figure 5, the pen service stores the data in a local database until the time the action is to be carried out. The pen service then refers to the location service and sends the action request to the device as explained above.

Clearly, with the basic example discussed in this paper, there is an issue that the paper only fits certain homes and hence is home specific. However, the forms can be designed in a more generic way. For instance, to program an "Away from home" service, the paper form does not need to show the available appliances in the home, but may rather show user profiles, such as "work in kitchen" or "go upstairs". This way the forms can be kept generic to fit a wide range of homes. However, the focus of this paper is to discuss the technical issues involved in controlling appliances with pen and paper rather than how best to design forms.

III. CONCLUSIONS

This paper discusses the control of networked appliances. The existing user interface to configure and remotely control these appliances requires a standard computer interface. This is particularly unsuitable for the target audience - which is often inexperienced with the use of computers. Furthermore, the interface is also unsuitable as a computer may not be available while the user is on the move.

This paper argues that Anoto enabled pen and paper are a suitable alternative. All users are used to pen and paper and hence this interface does not pose a barrier to adoption of this technology. Also, users on the move are usually able to fill in a simple paper form. Using Bluetooth and the mobile network, the data is transferred to a service provider where it is processed and sent to the user's home. Here the OSGi gateway offers a single point of control to execute the requests to appliances inside the home.

Programming a VCR using this technology will be as simple as ticking a box in the TV magazine against a particular TV programme. The service provider can derive the identity of the user from the used pen and the wanted program from the paper pattern. Thus the corresponding request is sent on to the OSGi gateway in the pen owner's home and then onto the networked VCR.

Besides being a useful technology for controlling appliances, the pen and paper interface may also be useful for configuring appliances. For instance, service manuals may have selection boxes printed on the Anoto pattern which when ticked, result in the corresponding device being set-up accordingly. Again the VCR may be a suitable example. Furthermore, pen and paper may be a suitable tool for service engineers when installing and configuring a central heating boiler for example. This potentially keeps the costs for training service engineers down and also may shorten the time it takes to install and configure a boiler in the customer's house.

The work as presented in this paper has been implemented as a prototype application. A video of a demonstration is available [11].

REFERENCES

- [1] M. Kolberg, E. Magill, and M. Wilson. *Compatibility Issues between Services supporting Networked Appliances*, IEEE Communications Magazine, Vol. 41, No. 11, November 2003.
- [2] UPnP, <http://www.upnp.org> (viewed: 3/25/2004)
- [3] X-10, <http://www.x10.org> (viewed: 3/25/2004)
- [4] IEEE 802.11, <http://grouper.ieee.org/groups/802/11/main.html> (viewed: 3/25/2004)
- [5] Bluetooth, <http://www.bluetooth.com> (viewed: 3/25/2004)
- [6] HAVi, <http://www.havi.org> (viewed: 3/25/2004)
- [7] MHP, <http://www.mhp.org> (viewed: 3/25/2004)
- [8] Open Services Gateway Initiative: OSGi Service Platform (Release 2), IOS Press, 2002.
- [9] Open Services Gateway Initiative, www.osgi.org (viewed: 3/25/2004)
- [10] D. Marples and P. Kriens, The Open Services Gateway Initiative: An Introductory Overview, IEEE Communications Magazine, Dec. 2001, pp 110-14.
- [11] Sysnet and University of Stirling, Video on Controlling Appliances with Pen and Paper, <http://www.cs.stir.ac.uk/%7Emko/capp.html>