Proceedings of the 5th WSEAS International Conference on Telecommunications and Informatics, Istanbul, Turkey, May 27-29, 2006 (pp491-496)

An Extensible Application Platform for Heterogeneous Smart Home Appliances and Mobile Devices

CHI-HUANG CHIU, HSIEN-TANG LING, PING-JER YEH, SHYAN-MING YUAN

Dept. of Computer Science
National Chiao Tung University
1001, University Rd. HsinChu, 300
TAIWAN

Dept. of C.S.I.E.
Tahwa Institute of Technology
1, Tahwa Rd, Qionglin, 307
TAIWAN

Abstract: Nowadays, various kinds of smart home appliances are widely deployed in our daily life. These embedded, networked, and programmable appliances could be accessed through various methods, but may not be integrated together due to the different kinds of home network protocols on which these appliances are deployed. On the other hand, with the penetration of mobile devices, people could leverage these appliances by fingers from anywhere around the world. However, the mobile accesses to these appliances are usually proprietary and require complicated development efforts. The paper aims at developing an application platform prototype named HomeBox for home appliances and mobile devices. The platform provides connectivity for heterogeneous smart home appliances, hosts an application runtime environment for smart home applications, and provides user interface transparency for mobile-devices to reduce the development efforts. The prototype has been explored in various applications to manage smart home appliances from mobile devices and demonstrate the extensibility for future devices and architectures.

Key-Words: mobile computing, smart home appliances, home network, OSGi framework, XUL, X10, UPnP

1 Introduction

In the past, home appliances were only plugged into wall power sockets and remained solitary, but now many of them are empowered by embedded microprocessors and inter-connected by wired or wireless network. Home network specifications (e.g., JINI [1], UPnP [2], X10 [3]) have been defined to organize a variety of home appliances ranging from lamps to televisions. With these specifications, applications could discover appliances, know their functions, issue commands to them, and even monitor their status. In such global network infrastructure, you can monitor the security alarm or control the air conditioner of your sweat home from everywhere around the world just on your cellular phone.

On the other hand, to manage these smart home appliances and the programs that use them, the needs of smart home platforms arise. OSGi alliance provides standard specifications for applications, runtime environment, and service interactions.

Different home network specifications have quite different histories and proponents, however. For example, UPnP has many applications on multimedia devices on TCP/IP network, while X10 is often used to control the power switches in the house by the signals on the existing power wires. Since there is no general standard encompassing their characteristics for now, the need for interoperability among these broadly deployed home networks, and also appliances running on top of them, becomes a serious problem.

Computing platform is yet another problem. The differences among mobile devices and appliances are so significant that applications have to be fully-customized. For instance, some devices have the capability to execute Java MIDlet programs but some devices may only have a WAP browser. How to make these applications adapted to various kinds of mobile devices is a crucial issue.

To solve these problems, this paper propose an application platform prototype, HomeBox, to provide a novel home network applications framework with full capabilities for heterogeneous home networks and various kinds of mobile devices. With the support of HomeBox platform, developers could concentrate their efforts on application logic without caring the specifications of mobile devices, different home network protocols, and the risk of incompatibilities for future devices.

The rest of the paper is organized as follows. Section 2 presents the background of related technologies and research works. Section 3 shows the design goal of the platform. Section 4 organizes the architecture of the platform. Section 5 discusses the details of major components. Section 6 describes the implementation experiences and working scenario. Finally, Section 7 concludes the paper with our future direction.

2 Background

2.1 Home Network System

In the past, computer network systems were mainly used by companies for business. Nowadays, as the prices of computers are going down and embedded system becomes more popular, it is not uncommon to have more than two systems within a single house or office. These systems may have to share files and peripherals, such as printer. The requirement of sharing resources inspires the concept of the home network system, which connects shared resources in a small local area. Through the home network system, devices in the home can communicate with each other.

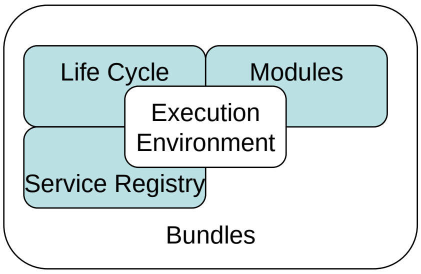

Fig. 1. The Structure of the OSGi Framework

In this paper, the term “home network system” is more specific to the network combined with various kinds of protocols and services which focus on smart home appliances.

2.2 OSGi Technology

The OSGi technology (Open Services Gateway Initiative) [4] is a set of specifications from the OSGi Alliance to define a standard computing environment for networked services. OSGi is component-oriented with the capability of life-cycle management to install, update, and remove components on the fly. Moreover, the alliance has already standardized the components for basic services such as HTTP servers, configuration, logging, security, user administration, XML, and many more.

The core component of OSGi specifications is the OSGi framework, which provides a standard environment for OSGi applications. Figure 1 illustrates the structure of the OSGi framework. The framework uses Java as its core execution environment (as shown in the center of the figure). The modules subsystem defines the class loading policies (based on the Java standard) and adds modularization features. The term *bundles* are application packages in OSGi framework, and they are managed dynamically by the life cycle subsystem. Finally, the service registry subsystem provides a comprehensive model to share objects between bundles.

The Role of OSGi in our HomeBox Platform

More and more manufacturers of smart home solutions have chosen the OSGi technology as the base runtime environment for their home gateway devices. Therefore, our HomeBox platform leverages and enhances the OSGi framework to maintain the program life cycle, deployment, and service integration. In this way the HomeBox platform can be deployed on every environment with OSGi support.

2.3 Related Works

The current researches related to our ideas could be separated into two groups. One of them focuses on interoperability between different home networks [5][6][7]. Such solutions usually utilize proxy to connect two different network protocols. The drawback is that extensibility is poor when there is a need to integrate more protocols.

The other group of researches focuses on accessibility of home appliances from mobile devices [8][9]. However, the presented solutions are usually bounded to specific type of mobile devices and home networks. There is little detailed information available on the architecture integrating different home networks and mobile devices.

3 Design Goals of the HomeBox Platform

The HomeBox platform integrates appliances, applications, and mobile devices across the home networks. Different environment has different requirement. Therefore, HomeBox is designed with the following objectives in mind:

1. Protocol Interoperability

There are plenty of home network protocols nowadays, yet the connection and control mechanisms of them are quite different. There is no unified approach to accessing the appliances residing in different network systems. That is to say, if a programmer wants to develop an application to control several appliances together, he/she has to be familiar with so many protocols and implements the desired functions for each protocol. As a result, the interoperability for different home network protocols is viewed as the most important factor in the HomeBox platform.

2. Platform Portability

There are many choices of computing platforms when executing applications on mobile devices, such as the J2ME platform, the .NET framework, HTML browsers, and WAP browsers. In order to be compatible with various platforms, developers have to spend lots of time porting their programs to each execution environment. Thus, the HomeBox platform should be designed to improve the portability of applications and simplify the porting effort.

3. Extensibility through Capability Abstraction

Appliances are a kind of resource for applications to use. But current practice, from a developer's perspective, is to hard-code them in the program itself, which reduces extensibility. One of our platform's objectives is to classify the capability hierarchies of appliances and to design abstraction interfaces for them. Since the capability abstraction is applied to not only the present but the future devices as well, existing programs on the HomeBox platform can recognize and leverage future appliances without any modification, provided that these appliances adhere to the abstraction.

4 System Architecture

With the objectives described in the previous section, the HomeBox system can be divided into three service parts: mobile devices service, smart home applications runtime service, and home appliances service, as shown in Figure 2.

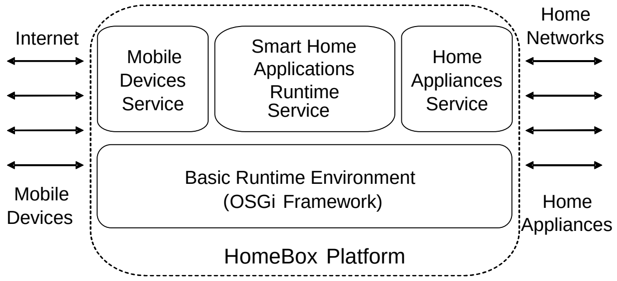

Fig. 2. The System Architecture of the HomeBox Platform

These services are all based on the OSGi framework as the runtime environment, as mentioned in Section 2.2. Therefore, they share the same property of service-oriented architecture (SOA). That is to say, services can be installed dynamically and independently, and be chained together simply by registering themselves with the OSGi Service Registry. Therefore, the three services in the HomeBox platform can work independently and also cooperate with each other through predefined interfaces and functionalities.

SHARS - Smart Home Applications Runtime Service

The service, underlying core of the HomeBox platform, manages the life cycle of smart home applications. It is much sophisticated than the counterpart in the OSGi framework. For example, in smart home applications, only one instance of some type of services is allowed to be activated, e.g., the home automation manager; while some of them should be invoked for one-time use, e.g, a simple remote controller. Moreover, in location-aware applications, each instance may be initiated for a user during the existence of him/her. These scenarios are not well-handled in OSGi.

In addition, the service provides an API wrapper for OSGi services and other components in the HomeBox platform. The wrapper provides a set of standard interfaces for smart home application developers, eliminating the need for code change after any modification of the services or platform in the future release.

MDS - Mobile Devices Service

Since the SHARS manages the life cycle, the mission of MDS is to present the mobile devices with appropriate user interfaces. With the HomeBox framework, the smart home applications only need to define one version of the display models and the MDS service is responsible for transforming the models to interactive user interfaces for various mobile devices. As a result, the application developers do not need to handle the detail of the user interfaces for different kind of mobile devices. From the developers' view, the service provides the end-devices transparency and they do not need to consider various specifications of mobile devices.

HAS - Home Appliances Service

The service integrates devices from various kinds of home networks. It handles all technical differences of various protocols and networks (e.g., JINI, UPnP, and X10), and exposes a standard service interface to eliminate the need for application developers to write different code for different kinds of appliances. From the developers' point of view, the service provides the home network transparency and they do not need to follow different home network protocols.

5 Design Details

5.1 Integration of Heterogeneous Home Networks

In order to integrate heterogeneous home networks, the HAS designs some universal interfaces to provide a unified procedure for smart home applications and better extensibility for future home networks. Figure 4 shows the block diagram of the HAS. The HAS provides two groups of services: physical services and logical services. Each physical service provides supports from a home network service like a device driver in an operating system. The logical services work as a proxy for each appliance on the home networks.

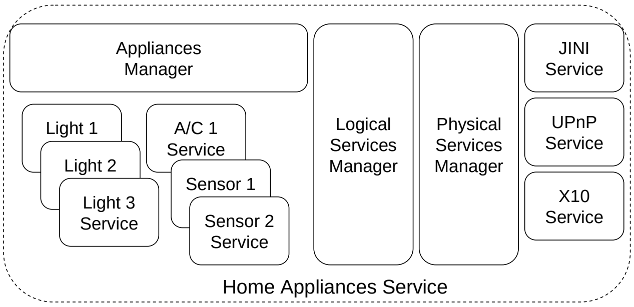

Fig. 3. Block Diagram of Home Appliances Service

Physical Services

The Physical Services Manager provides an abstract interface to integrate heterogeneous home appliances across heterogeneous home networks. Although some physical services such as JINI and UPnP are bundled in the OSGi specifications, the Home Appliances Service still needs to wrap them to interface with other sets. With the abstraction of network protocols, new home network protocols could be supported in the HomeBox platform by implementing a new physical service which conforms to the specified interfaces.

The interface of the Physical Services is named HomeBox Appliances Provider Interface. It organizes the features of existing home network protocols and is separated into four interface categories: Devices Discovery, Action Call, Property Query, and Event Notification.

First, Devices Discovery interface could discover the existence of appliances. Second, Action Call interface could dispatch the command to appliances. Third, the Property Query interface could get the status of the appliances. Finally, the Event Notification interface could inform Smart Home Applications about the change of the appliances state.

Logical Services

Once a device is discovered by the Physical Services Manager, a corresponding logical service will be created and registered as a reference to a home appliance. The logical services also have a set of interfaces for Smart Home Applications to control the home appliances directly. The interface is named HomeBox Appliances Interfaces. It is similar to the Provider API applied in the Physical Services but is more specific to managing a single device.

The Appliances Manager provides a lookup service for each logical service and an identity string will be assigned to each device. Generally, a detached device will be assigned the same identity string when it is attached again. Some protocols such as UPnP also have an identity string for each device but some protocols do not have a unique string. For the devices without a unique identification, the appliances manager assumes that the devices with the same configuration are the same.

5.2 Access from Mobile Devices

To support the mobile devices access, the HomeBox platform adopts the ART (Adaptive Remote Terminal) [10] as the core component of SHARS (Smart Home Applications Runtime Service) and MDS (Mobile Device Service). The ART sever is a runtime execution environment for mobile applications. ART applications (the applications that follow the specifications) will be executed in the ART server and the user interface could be adapted to various kinds of ART clients (e.g., Java, J2ME, .Net Compact platforms). In addition, some limited UI widgets also support the access from Web browsers, WAP and XHTML phones with a HTTP connector in ART server.

The main idea behind ART is to divide an application into two parts, UI and logic, in both development and runtime. This idea lowers the development cost and makes the adaptation possible. Figure 5 illustrates the process of building an ART application. In the ART specification, user interfaces are defined in XML files which follow the XUL (XML User Interface Language) standard [11]. The XUL files define the user interface and logic files (written in Java) defines the program logic.

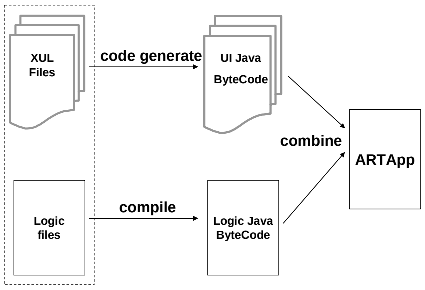

Fig. 4. Development Process of Creating an ART Application

The ART system not only provides the access for mobile devices but also manages the life-cycle of ART applications, described in the SHARS service mentioned in previous section. On the HomeBox platform, all smart home applications are ART applications. The developers should follow the development guideline to fit the mechanism provided by the ART system.

Moreover, each application instance in the ART server could be attached by more than one client to support the collaboration. The HomeBox platform has extended this collaboration feature and specified four application execution management models. These models define the way to handle concurrent connections and the life-cycle for each instance to fit the need of smart home applications.

Always On — An Always On application will be started along with the initialization of the HomeBox, and all accesses to it will be attached to the only one instance. The model is designed for home automation applications because their home automation rules should always be applied whenever the system is available.

Collaboration — Similar to the Always On model, but the application will be started when the first user is attached; and terminated when the last user is detached. Applications which need to be accessed from multiple devices with the same instance should use this model.

Single — In this model, an instance is prepared independently for a client connection session. The model could be applied for applications like “remote controller.”

Session — Similar to the Single model, but the instance of the application will be kept for a period of time to wait the next connection from the same device. If an application wants to keep the states of some variables, the model could be applied.

5.3 Extensibility for Unknown Appliances

Another important design issue for the HomeBox platform is to build a solution not only for currently available smart home appliances but also with the extensibility for the future appliances. Therefore the platform has built a service descriptor mechanism to provide such extensibility. Figure 6 shows the interactions diagram for the HomeBox platform.

As far as extensibility is concerned, the platform provides a system operator role to feed the metadata to the platform. The metadata contains the class libraries of the physical services and the service descriptors to map the HomeBox Appliance Interface to the physical operations of the underlying home network operations.

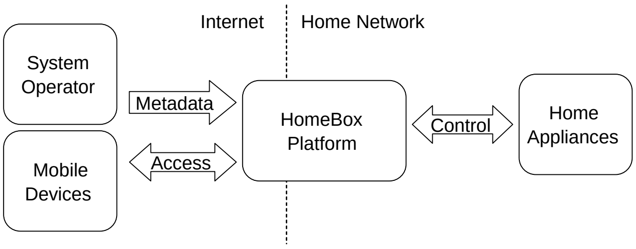

Fig. 5. The Interactions with HomeBox Platform

The following steps show the process to setup a new device when it is detected:

1. Discover a new device in the home network from an existing physical service.
2. Send a request to the System Operator to retrieve the service descriptor of the device.
3. Update the version of physical service class library according to the required version described in the service descriptor if need.
4. Initialize a logical service for the appliance.
5. Register the logical service at the appliances manager.

The previous steps are named Device Initialization Process in the HomeBox platform. With such process, the platform could make sure that: 1) the physical services could be updated to the latest version. 2) The new released home appliances could be supported in the platform with the help of the system operator. In addition, all the components of the HomeBox platform could be updated automatically by the periodical synchronization with the system operator to make sure the availability of the latest version.

6 Implementation

Currently the HomeBox environment is implemented mainly on the Oscar 2.0 [12]. Oscar is an open source implementation of the OSGi framework specification. One of Oscar’s goal was to provide a fully compliant implementation of the OSGi framework specification, but currently it is only compliant with a portion of the OSGi 3 specifications.

6.1 Sample Home Applications

In order to demonstrate and prove the concept of the HomeBox platform, two sample smart home applications are implemented. The first one is a universal remote controller. Any ART client on different platforms could control the lights in X10 protocol, and also television and air conditioner in UPnP protocol. This application demonstrates the capability for mobile devices to control smart home appliances in different home networks.

The second sample application is a home automation rule engine for smart home appliances. Users could define rules to map events to actions among the appliances just from their mobile phones or Web browsers, and afterwards some home automation mechanism will be applied accordingly. In this demo, UPnP televisions, UPnP lamps, and X10 motion detectors could be integrated together. The combination of rules could turn on the lamp when the motion is detected, and also turn off the television and air-conditioner when there is no motion object detected. The sample application demonstrates the capability not only to control devices across different home network but also to integrate these appliances to work together smoothly.

7 Conclusions and Future Work

The HomeBox platform demonstrates an ideal solution for developers to integrate the heterogeneous smart home appliances from mobile devices. All the applications in this platform have device transparency to be accessed from various kinds of mobile devices. Moreover, the platform follows the OSGi specification and could be deployed directly into any embedded system with OSGi support. The application life-cycle management mechanism also shows the capability to work as a core platform for home automation applications.

Another achievement of this platform is to integrate the home appliances across different home networks seamlessly. These appliances could not only be managed from a mobile device but also be triggered by events from the appliances in different home networks. Furthermore, the service descriptors from the system operator also enhanced the future extensibility of HomeBox platform.

Comparing to other systems mentioned in the related works section, the HomeBox platform provides interoperability for different home networks, compatibility for different mobile devices, reusability for the smart home applications, and extensibility for the future.

This paper focuses on the enabling architecture of integrating heterogeneous home appliances from mobile devices. Further work is needed to provide more comprehensive and precise abstraction of interfaces for similar home appliances in order to facilitate developers to build more reusable applications for home automation solutions.

References:

[1] JINI Community: JINI Network Technology Specification, http://java.sun.com/products/jini/.

[2] UPnP Forum: The UPnP Standard Specifications, http://www.UPnP.org/standardizeddcps/default.asp.

[3] X10 Remote Controllers, http://www.x10.com/.

[4] OSGi Alliance: OSGi Specifications, http://www.osgi.org/osgi_technology/download_specs.asp?section=2.

[5] Chau, O.S., Hui, P., and Li, V.O.K.: An Architecture Enabling Bluetooth / JINI Interoperability, *Proc. IEEE PIMRC*, Barcelona, Spain (2004).

[6] Song Yean Cho, Dae Young Seo, Tai Yun Kim: Gateway Framework for Home Appliance's Interoperability Based on Heterogeneous Middleware in Residential Networks, *International Conference on Consumer Electronics* (2002).

[7] J. Allard, V. Chinta, S. Gundala, G. G. Richard III: JINI Meets UPnP : An Architecture for JINI/UPnP Interoperability, *Proceedings of the 2003 Symposium on Applications and the Internet* (2003).

[8] Latvakoski, E.J., Paakkonen, P.: Remote interaction with networked appliances attached in a mobile personal area network, *IEEE International Conference on Communications* (2003) 769- 773.

[9] Mariana Nikolova, Frans Meijs and Peter Voorwinden: Remote Mobile Control of Home Appliances, *International Conference on Consumer Electronics* (2002) 100-101.

[10] Li-san Yao: An Adaptive Mobile Application Development Framework, *Master Thesis of National Chaio Tung University.* http://www.cis.nctu.edu.tw/~gis89801/LSYao.pdf. (2003)

[11] David Hyatt, Ben Goodger, Ian Hickson, Chris Waterson: XML User Interface Language (XUL) 1.0. http://www.mozilla.org/projects/xul/xul.html. (2001)

[12] Oscar – An OSGi framework implementation: http://oscar.objectweb.org/