OPEN SERVICE RESIDENTIAL GATEWAY FOR SMART HOMES

D. Zhang H. Lee X. Ni S. Zheng
Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore 119613
{daqing, hlee, nixiao, szheng}@i2r.a-star.edu.sg

ABSTRACT

This paper presents a residential gateway (RG) based, service-oriented system infrastructure for smart homes. The RG is designed to enable the interconnection of various home networks and appliances to Internet, and at the same time serves as a service execution platform. The Open Service Gateway Initiative (OSGi) and device access protocols such as UPnP have been incorporated in the RG to facilitate the management of home devices and services. Context aware services, SIP service and other OSGi application services have been developed to make the platform unique.

1. INTRODUCTION

With the development of Internet broadband access, wireless communication, smart sensors/devices and home networking technologies, smart home has become a hot area for research and development in both academia and industry. Smart homes provide people security, energy saving, convenience and a better lifestyle, they may also provide the aged and disabled people better support for independent living and social interaction. In countries with a high penetration rate of Internet broadband connection, various preliminary smart home solutions have been developed and showcased, numerous pilot runs in a certain scale are also under way [1].

A smart home is a house or living environment where home appliances can be accessed and controlled either locally or remotely. For example, you may monitor the home appliances status in the office and switch them on and off according to your needs, you may get a notification message when your home security system detects that intruders want to enter your house, you may let TV deliver programs based on your personal interests, you may even be informed by the refrigerator what is short of stock inside, and the volume of TV and HiFi can be lowered automatically while the phone rings, etc.. If we look at the potential networked devices in the home today, they basically fall into four categories:

- infotainment devices - TV, DVR, HiFi
- computer devices - PC, printer, scanner, web slate, PDA
- communication devices – telephone, fax
- control devices - all everyday devices such as kitchen appliances, lighting systems, air conditioning systems, security systems, blinds, meters (water, gas, electricity)

Currently different home networking technologies [2] have been developed to cater for the special needs of each category, such as IEEE 1394 and Fast Ethernet for multimedia and gaming devices, Etherent and HPNA for computing terminals, WLAN and Bluetooth for mobile devices, and Powerline for home appliances. Correspondingly a variety of device access protocols such as HAVi, UPnP, Jini, LonWorks, CEBus, X10, etc. have been proposed to ease the management.

To access the devices, the traditional approach is to network the same category of devices with dedicated wires and a dedicated box which implements the related protocols to ease the management, then several boxes are needed to connect all the home devices to implement the smart home solution. In case new networking technologies emerge, a new box need to be added to accommodate the new devices. Apparently this solution has several problems: 1). more than one box is needed. 2). devices in different networks cannot share resources and collaborate. 3). services need to be developed in different platforms and cannot inter-operate. Actually, most of the current service providers such as utility, telephone and cable TV companies all have dedicated wires and boxes in the home, while the computer network with a modem is most popular, the infotainment network with a set-top box and the control network with a certain gateway can be found in a lot of homes nowadays. The problems posed by this approach call for a better solution to integrate all the home networks and allow for further evolvement and expansion.

In this paper, we propose and implement an OSGi [3] compliant open service RG for smart homes. The proposed OSGi compliant open service RG has the following features:

- Bridging various home networks, either wired or wireless
- Connecting home networks to Internet with broadband access
- Providing routing and address translation
- Enabling secured remote access and data exchange of home devices
- Remote service and device management

- Allowing technologies in WAN and HAN evolve independently

The paper is organized as follows. First, the OSGi and service delivery system is introduced, followed by the system architecture of the smart home. Then the unique OSGi services are presented in Section IV. Finally some concluding remarks are drawn.

2. OSGi AND SERVICE DELIVERY

The OSGi was established in 1999 to define open specifications for the delivery and provisioning of multuple services over wide area network to local networks and devices in homes, cars and other environments. The OSGi specifications try to standardise the way for secure and reliable service delivery and provisioning, for remote life cycle mangement of services, for reusing the services as well as for bridging between different home networking standards.

2.1 OSGi Framework

The OSGi specification defines a service framework which provides an execution environment for services. The framework includes a Java runtime environment extended with service life cycle management, persistent data storage, version management, and a service registry. The life cycle management provided by the framework allows application developers to divide their applications into small self-installable components, called *bundles*. Bundles are software components implemented in Java that can be downloaded, installed, and activated in a framework. When activated, bundles may register services in the service registry, to provide services to other bundles in the framework. When a service is no longer needed it can easily be removed from the framework, without affecting other bundles. Bundles can be installed and updated by the framework in a dynamic and scalable fashion. New bundles can be installed for added features, and existing bundles can be modified and updated, without having to restart the entire system. The service registry is used to find and use other services in a secure and controlled manner. The service gateway operator has full control over the platform and it decides which service bundles are allowed to use.

2.2 OSGi Bundles

In the OSGi environment, bundles are the only entities for deploying Java-based applications. A bundle is a Java Archive (JAR) file that comprises Java classes and other resources, it provides services to other bundles. The bundle typically contains Java class files and non-executable resources, such as icons, HTML files, help files, etc. A bundle also contains a manifest file, describing the contents of a JAR file and providing information about the bundle. The manifest file contains special headers to describe the bundle to the framework and states the bundle's dependencies on external Java classes. Furthermore, the manifest file designates a special class to be used as the bundle activator. The framework must use this class and invoke its start and stop methods to start or stop the bundle respectively.

2.3 OSGi Service Delivery System

Fig. 1 shows a typical OSGi service delivery and provisioning architecture and how the services are d

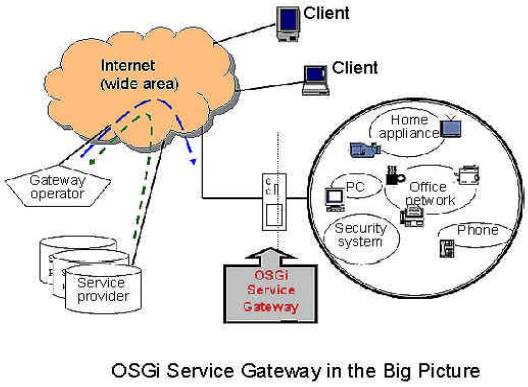

Figure 1. OSGi service delivery and access architecture

There are different entities in the OSGi architecture. The OSGi framework and services are put on top of the JVM and OS in the RG, the RG itself has the functionality of routing, firewall and NAT. The RG is connected to Internet through xDSL or Cable modem, the gateway operator has full control over all the RGs in the homes, usually it uses HTTP for remote service access and management. The home appliances are connected to RG through various home networks, they can all be accessed from RG using service bundles. When a new device or new feature is needed, then a corresponding service bundle which speaks the device's "language" needs to be put in the OSGi framework. And all the services and devices can be accessed through HTTP either inside the home network or in the Internet. The service providers are those who provide the OSGi services.

2.4 The OSGi Backend System

The OSGi backend system enables the service/gateway operator to manage the gateways and deliver services to end users, it supports functions such as configuration, diagnostics, update/upgrade and billing [4]. The key features of the backend system include:

- Remotely administer the connected OSGi based service gateways
- Perform service publishing and service delivery
- Start/Stop and bill delivered services
- Remotely troubleshoot and solve user problems

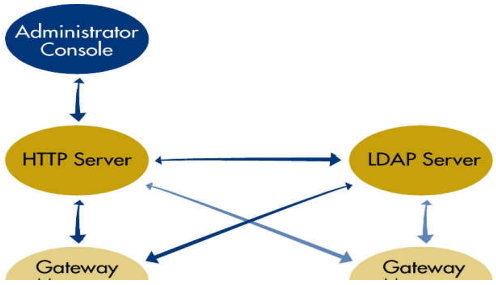

Figure 2. Basic topology of OSGi backend system

The overall structure of the OSGi backend system is shown in Fig. 2, it includes the following fundamental components:

- Service gateway (G1 – G10): the gateway where the OSGi framework is embedded.
- Gateway manager: manage a set of service gateways and interact with HTTP server and Database.
- LDAP server: management engine for database.
- HTTP server: enables GUI administration for the network administrator.
- Administrator console: web browser to visually manage the network.

3. SMART HOME ARCHITECTURE

The system diagram of the smart home is shown in Fig. 3, basically it involves the following four technology components:

- Broadband access networks which provides the "always ON" high speed digital pipe to the home.
- Residential gateway which interconnects the broadband access network and the home area network.
- Home area network which connects various home devices
- Device access technologies that facilitate the discovery and interplay of the home devices and services.

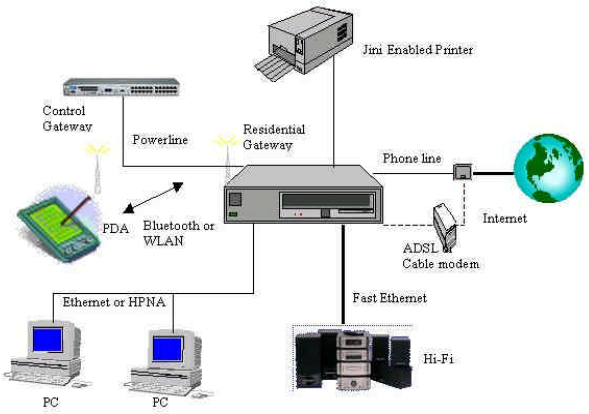

Figure 3. System diagram of the smart home

3.1 RG Hardware Architecture

As shown in Fig. 3, the centre of the smart home is the RG which enables interconnectivity of home appliances and connectivity for those devices to the Internet. The RG has been built around a highly integrated Intel Celeron processor with 256-Mbyte memory and 16-Mbyte flash memory. One or more Hard Disk Drives can be added to host the services while all the system software and key data are stored in the flash. The separation of system space and user space has the advantage of high reliability and flexibility to host as many applications as possible without crashing the system. The RG utilises embedded Linux as the OS, it adopts a modular design so as for customers to selectively include or exclude interface components as needed.

For broadband access, the prototype of the RG has the ADSL module (512 kbps) and Ethernet (10/100 Mbps), while future versions may include the cable modem (1.5 Mbps) and other access technologies. It is expected that the external interfaces of the RG will be driven by the cost and availability of external access technologies. A V.90 (56 kbps) interface is also included for legacy purpose, it can be used for simple, non-multimedia applications such as existing standalone fax machine. The V.90 interface also serves as a backup access system if other external interfaces are unavailable.

For in home networking support, although fast Ethernet is implemented to support gaming and multimedia applications, "no more new wire" has been the design principle of the RG. The current version of RG support home networking standards such as HomePNA using existing telephone lines, wireless LAN standards (802.11b) and Bluetooth. For control devices, RG uses powerline or USB to connect with a control gateway, and the control gateway is used to take care of all the control appliances such as washing machines, refrigerators, and lights etc.. through device control protocols such as LonWorks, CEBus, EIB and X10.

3.2 RG Software Architecture

While the RG hardware architecture is easy to understand, building the software framework to support various devices/services is quite complicated. One reason for this is the difficulty in developing service software and managing services for home users. For example, device installation and configuration should be simple and automatic, remote device and service troubleshooting should be supported, services should be shared among themselves. As many people cannot even program their VCRs, so for them, managing smart appliances must be as simple as turning on the radio. Therefore, to enable the recording service, there must be

a way to register the deployed services, discover the services and make use of the services.

Another difficulty is how to accommodate different home networking technologies and standards. As most of them will coexist for the foreseen future, the software framework must ensure that these technologies and standards can work together and inter-operate without user's intervention.

As shown in Fig. 4, the RG software architecture consists of three layers: the OSGi layer, the system layer and the physical interface layer.

The OSGi layer is composed of two key components: service framework and service bundles. Service framework provides a service hosting environment as well as a set of common APIs to develop service bundles. It also includes several basic service bundles such as http service, log service, configuration management, permission administration, preferences, user manager, device manager etc.. The rest of the service bundles are provided by service developers, it is important to note that adding any new functionality in the service platform is usually translated into adding new OSGi service bundles. For instance, in order to support service access using SIP, a SIP service bundle is needed handling all the communication issues [5]. And a device access service bundle is needed to bridge the home network protocols and access the devices connected to heterogeneous home networks. Context aware service bundles are needed to support context awareness.

The system layer refers to the Java Virtual Machine and Operating System (OS) in the RG, the OS performs the functionality of IP forwarding, firewall and Network Address Translation (NAT). The physical interface layer deals with the low-level communications with various WAN and LAN connections of RG.

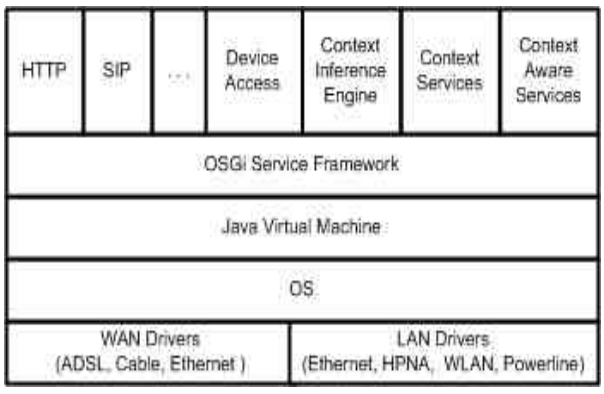

Figure 4. Software Architecture of RG

3.3 Device Access Software Architecture

As the home networks and devices vary and the number of devices is increasing, the device plug-and-play becomes critical. In addition, automatically discovering and deploying each device's capability is highly desirable. This is enabled by the device access software architecture. The proposed device access system architecture includes the device access service embedded in RG and a device access software stack embedded in each device.

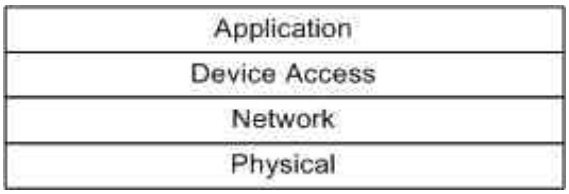

Figure 5. Software Architecture of smart devices

As shown in Fig. 5, the software architecture of a device consists of the application layer, the device access layer and networking layer. While the application layer presents the features and capabilities of the device, the device access layer is in charge of announcing its presence to the network, discovering other devices in the network and making use of the services provided by other devices. The networking layer takes care of the data transmission in the network.

With the proposed mechanism, all the devices provide services which can be discovered and used by other devices in the home network. At the same time they also register the services in the service gateway, thus all the devices can inter-operate with each other with the OSGi service gateway as the bridge.

4. OSGi SERVICES

In order to verify the service-oriented smart home infrastructure and provide value to end users and developers, some services have been identified and implemented:

- Simultaneous internet access to multiple home users
- Voice and video over IP
- Home surveillance
- Remote file management
- Home control and automation
- Notification
- SIP
- Context aware

The first two services are built on top of the Linux OS, while the rest of the services are OSGi services [6], running on top of the OSGi framework in RG.

4.1 Home Surveillance Service

The home surveillance service allows picture and video captured using a web cam to be monitored remotely, it also allows playback of the recorded videos. In the image capture mode, the surveillance service captures still image periodically. The user can set the frequency of the capture operation. In video capture mode, the

video can be HTTP-streamed over the Internet or Intranet, and stored into the secondary memory to be replayed later. Both the real-time video and stored video can be played on a Web browser without waiting for the complete stream data to be downloaded. Compared with Real-time Transport Protocol (RTP), HTTP-stream doesn't require Java Media Framework (JMF) plug-in in the Web browser and is able to go through firewall easier. Transmitting across the Internet in real time would, however, require a high-bandwidth network, so that the client end can play the media data continuously.

4.2 Remote File Management

The remote file management service allows files to be remotely downloaded from the RG, and uploaded to the RG file system in a secure manner via the Internet via a web GUI. The security of the file transfer is protected by signed Java code. Digital certificate generation and maintenance can be performed using Sun's software tools. The service makes use of the HTTP service for presentation of its web GUI. It presents a dynamic directory tree of the file system of the RG using applet, to meet the exacting user interactivity and direct manipulation requirements. For file upload operation, the web-based GUI allows the user to browse and choose the file to be sent to the RG. For download operation, the web GUI allows intuitive point and click operations implemented using Java's swing API.

4.3 Home Control and Automation

The home control and automation service allows secure control of home appliances locally or remotely using X-10 control protocol. Devices like lights, fans, washing machines can be added to or deleted from the X-10 control network. The control network devices can be controlled by the home control and automation service using the web GUI. The home control and automation service keeps a persistent record of all the X-10 devices it controls as well as their current states. By using the Macro, home automation can be implemented by using events triggering events.

4.4 Notification Service

The notification service is used to notify that a certain event or some urgent events occur. It may notify the user through SMS, MMS, phone or e-mail and notify services through event.

4.5 SIP Service

Besides the HTTP service natively provided by OSGi framework, an Session Initiation Protocol (SIP) service has been implemented in service gateway so that both HTTP and SIP terminals can be used to access the services anywhere, anytime. As SIP has been supported for instant message and in 3G, using SIP to access OSGi services will support wider range of applications and devices.

4.6 Context Aware Services

Context aware services deal with the abstraction, inference and utilization of context information in intelligent environments [4]. They utilize the context information such as location, time and environment to adapt to the rapidly changing situations.

5. SUMMARY

In this paper, an OSGi compliant RG based, service-oriented smart home solution has been proposed and implemented. Compared with the other OSGi based RGs [7][8], the proposed solution exhibites the following unique features:

- SIP is incorporated in the OSGi compliant RG so that both SIP and HTTP can be used for service access, thus future devices like PDA, hand-phone, PC and notebook can all be used to deliver and access services in smart homes.
- Context aware services are incorporated in the OSGi compliant RG so that both explicit and implicit information can be utilized to meet the different needs of inhabitants.

6. REFERENCES

[1] Li Gong, "A Software Architecture for Open Services Gateways," *IEEE Internet Computing*, Jan./Feb. 2001, pp 64-70.

[2] B. Rose, "Home Networks: A Standards Perspective," *IEEE Commun. Mag*., Dec. 2001, pp 78-85.

[3] OSGi Specification v3.0, http://www.osgi.org

[4] D. Zhang, X. Wang, K. Leman, W. Huang, "OSGi Based Service Infrastructure for Context Aware Connected Homes", *The 1st International Conference on Smart Home and Health Telematics*, Paris, France, Sept 2003

[5] D. Zhang, "A New Service Delivery and Provisioning Architecture for Home Appliances", *IEEE International Conference on Consumer Electronics*, LA, USA, June 2003, pp378-379.

[6] S. Zheng, D. Zhang, X. Ni and S. Liu, "A Case Study of OSGi based Service Collaboration", *The 7th IEEE International Symposium on Consumer Electronics*, Syndey, Australia Dec. 2003.

[7] D. Valtchev and I. Frankov, "Service Gateway Architecture for a Smart Home," *IEEE Commun. Mag.*, April 2002, pp 126 – 132.

[8] K. Hofrichter, "The Residential Gateway as Service Platform," *IEEE International Conf. On Consumer Electronics, USA, 2*001, pp 304-305.