RESIDENTIAL GATEWAY FOR THE INTELLIGENT BUILDING
A design based on integration, service and security perspective

Budi Erixson, Jochen Seitz
Department of Information Technology, Division Communication Networks, Technische Universität Ilmenau, 98693 Ilmenau, Germany

Keywords: Residential gateway, Service gateway, integration

Abstract: In this paper we present the architecture of a residential gateway, which is designed with the OSGi (Open Service Gateway Initiative) that coordinate with the LDAP (Lightweight Directory Access Protocol) in order to integrate and connect the various home networks and appliances with the internet securely. The presented architecture is applied in the intelligent building project, LISTIG (LAN-integrated control system for intelligent building technique), the cooperation project between Technische Universität Ilmenau, Desotron (Sömmerda), with the University of Applied Science, Jena and the HFWK (Hörmann Funkwerk Kolleda GmbH) in Germany. This project is currently still on the progress to achieve the full integration of the several of home networking technologies, protocols and services.

1 INTRODUCTION

The research and development of intelligent building applications and home networking technology grow rapidly, and become an interesting topic for the industries and science area. On the other side, there isn't any intelligent building standardization, because there are so many perspectives and approaches to implement it.

The variety of different protocols, home networking technologies and services are the things, which cause these different perspectives and approaches to design and implement the smart home. The different home network technologies, which have been developed in order to fulfil the needs of home networks are IEEE 1394 for multimedia devices, Ethernet and HPNA for computing terminals, Bluetooth and WLAN for mobile devices, Powerline for home appliances. And there are the mechanisms such as Jini and UPnP for the device discovery and EHS in order to manage the integration of new devices into the network.

The natural solution to integrate the diverse and manifold home networking technologies, protocols and services is the centred point. And this centre point is the residential gateway. The implementation of smart home could be easier when the residential gateway has the abilities and reliabilities such as:

- can connect the home network with to the world wide internet
- integrate the various home networking technologies in house.
- security remote management
- easy to extend and up date the software

In this paper we implement OSGi to design the residential gateway, because its reliable characteristics, which are: platform independence, application independence, multiple service support, service collaboration support, top level security, multiple network technology support, and simplicity.

The paper is arranged as follows. Firstly, the OSGi and its service framework and its service platform are introduced. Then, the design of the residential gateway of our project is presented, and then, the current position with the future work of the "LISTIG" project is presented. Finally, some concluding remarks are drawn.

2 OSGI (OPEN SERVICE GATEWAY INITIATIVE)

OSGi (Open Service Gateway Initiative) was founded on March 1999 .The Open Service Gateway Initiative (OSGi) mission is (OSGi, 2003):

- to enable the deployment of services over wide area networks to local networks and devices.

Erixson B. and Seitz J. (2005).
RESIDENTIAL GATEWAY FOR THE INTELLIGENT BUILDING - A design based on integration, service and security perspective.
In *Proceedings of the First International Conference on Web Information Systems and Technologies*, pages 99-103
DOI: 10.5220/0001231500990103

- to create open specifications for the network delivery of managed services to local networks and devices**.**

2.1 The Framework Specification

The OSGi framework is designed to create extensible services using the Java programming language, which provides execution environment for the needed services. The OSGi framework must provide a consistent programming model during application development. The execution environment extends it with the lifecycle management, service registry, persistent data storage and version management.

The OSGi framework provides the lifecycle management that allows the developers to divide the applications into self-installable components. These components are called *bundle*.

Bundles can be downloaded on demand and removed when they are no longer needed. When a bundle is installed and activated in the framework, it can register any number of services that can be used by other bundles. This dynamic aspect makes the software extensible on the device after deployment: new bundles can be installed for added features or existing bundles can be updated for bug fixes.

The developers design an application as a set of bundles that contain services, with each service implementing a segment of the overall functionality. The entities in the framework are:

- Services, the Java classes that perform certain functionality, usually written with interface and its implementation separated. A service is a self contained component, accessible via a defined service interface. In the OSGi model, an application is built around a set of cooperating services: it can extend its functionality at runtime by requesting more services which it requires. The framework maintains a set of mappings from services to their implementations and has a simple query mechanism (LDAP based syntax) that enables an installed service to request and use the available services. The Framework manages the dependencies among services. A developer defines a service as an interface and provides its implementation. Then she can register the service with the Framework. When a service is registered, it can be given a set of properties (name/value pairs) to enable a sophisticated retrieval based on LDAP attribute comparisons.

After a service is published, other services can use it to accomplish their tasks; they look up the service from the framework with a search filter, and will get back the matching service references. The service reference can then be used to get a Java object that implements the desired service. The framework does not actually give out references to objects implementing a service directly, which would also instantly create a dynamic dependency on the bundle providing the service. Instead, it gives out a `ServiceReference` object, which can be stored and passed on to other bundles, without the implications of dependencies. When the service is actually to be used by a bundle, a reference to the implementing object can be obtained from the current `BundleContext`, passing it the `ServiceReference`.

- Bundles, the functional and deployment unit for shipping services. A bundle is a JAR (Java Archive) file that:

- contains the resources to implement zero or more services.
- contains a manifest file describing the content of the JAR and providing information about the bundle.
- states dependencies on other resources, such as Java packages, that must be available to the bundle before it can run. The framework must resolve these packages prior to starting a bundle.
- designates a special class in the bundle to act as bundle activator. The framework must instantiate this class and invokes the start and stop methods to start or stop the bundle respectively.
- can contain optional documentation within the JAR. This can be used to store the source code of a bundle. Management systems may remove this information to save storage spaces.

When a bundle is already started, its functionality is provided and services are exposed to other bundles installed in the OSGi environment. They can then use the framework to access this functionality. For each bundle installed in the framework, there is an associated bundle object. This object is used to manage the namespace of the bundle's Java classes, by directing the loading and resolution of those classes. By establishing separate namespaces for bundles, class-name conflicts among bundles are avoided.

The framework provides its scoping as an additional precaution: if two bundles have class names in common, the fact that those names are scoped by different bundle namespaces means that there is no contention. A bundle is used to get information about the current bundle lifecycle status and to start, stop and update bundles.

2.2 The OSGi Service Delivery

The OSGi framework and services exist on the part of Java environment and the operation system in the residential gateway. The residential gateway has the connection with the world wide through the cable modem or xDSL.

In the home network architecture, the role of the gateway operator is to control the residential gateway. The gateway operator uses HTTP for services and remote management. On the other side, the appliances in home are connected to each other through the home networks, which can be accessed with the service bundles of residential gateway through the HTTP. And the provider of the OSGi services is the service provider. The architecture of the home network architecture, which implements OSGi residential gateway, is shown in figure 1.

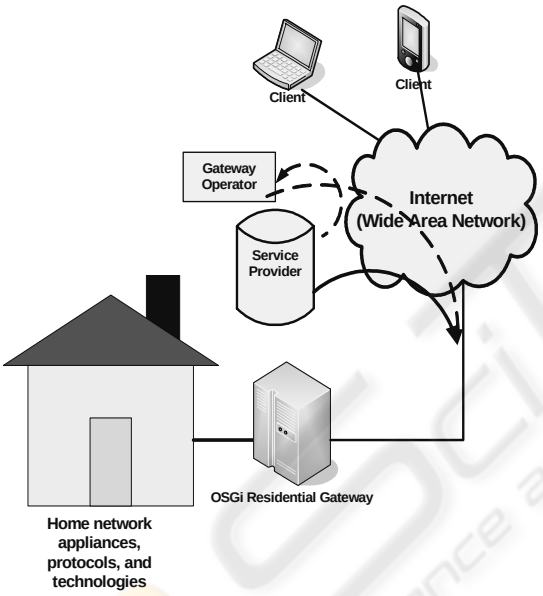

Figure 1: Service gateway framework

2.3 Lightweight Directory Access Protocol (LDAP)

Lightweight Directory Access Protocol (LDAP) provides the protocol to access the directory services with a unique method to handle a lot of records with huge volume capacity. LDAP runs over the TCP/IP or the other connection oriented services. The LDAP adopts the database and the security model of X.500 protocol and adapts the internet security standard such as SASL (Simple Authentication and Security Layer) and SSL/TLS.

In the residential gateway environment, the LDAP will play not only as a directory service but as the management engine for the database. The LDAP is chosen as the directory service, and its characteristics are:

- Global directory service. LDAP is designed as the directory service that allows the users to access the information as the unique identification.
- Open standard and interconnectivity. The LDAP can be adapted by the vendors or users, which use the TCP/IP. This makes the connection to the internet.
- Easy to customize and to extend. It is flexible to regulate the interface display, and easy to combine the LDAP with another programs, because LDAP is an embedded program.
- Security and controlled access protocol. The authentication process will make the transaction process more secure.

The LDAP protocol works with client-server model. The client will send the request or message to the server, and then the directory server will answer with its ability in answering the requests at the same time. The server will read every ID for the identification. This transaction uses the Basic Encoding Rules.

There are 9 operations in the LDAP protocol, which divided into 3 parts. There are:

- Interrogation operation
  a. Search (data search operation)
  b. Compare (data comparison operation)

- Update operation
  a. Add (data addition operation)
  b. Delete (data deletion operation)
  c. Modify (data modification data operation)

-Authentication and control operation
  a. Bind. The processes in this phase are:
  - The client makes an authentication to the directory server.
  - The TCP connection, client sends the distinguished name and authentication mandate.
  - The server adjusts the mandate authentication, and then the answer will be sent to the client.
  - If the mandate is rejected, the answer is *anonymous bind*
  - Authentication process will be finished, when the connection status is opened, until the client makes the de-authentication.

b. Unbind. The interruption connection between client and server.

c. Abandon. The transaction is cancelled.

3 THE INTELLIGENT BUILDING DESIGN AND IMPLEMENTATION

The general and basic architecture of intelligent building includes these following technology components:
- Residential gateway
- The home area network
- Broadband access network
- Device access technologies

3.1 Residential Gateway Hardware Architecture

The basic component, which we use to implement as the residential gateway is produced by the HFWK (Hörmann Funkwerk Kolleda GmbH), who cooperate with the University of Applied Science, Jena and The Technische Universität Ilmenau, and Desotron (Sömmerda).
Generally, the fundament of this basic component is like a normal PC, with the main board, which has the measurement of 120 x 124 mm, with the processor, chip graphic, IDE controller, USB controller, and memory. It has Boot-PROM and 2 PCI sockets.
Below is the technical specification of the hardware (table 1), which is depicted in figure 2 and it uses Ethernet (10/100 Mbps) for the broadband access.

Table 1: Hardware technical specification

Main processor	486 Atlas CPU with math. Coprocessor
Memory	128 MB SDRAM
IDE-Controller	1 channel ATA
Boot-Prom	8 MB Flash RAM
COM-Port	2 x LVTTL
USB-Port	1 x version 1.1
PCI-Bus	2 x standard 32 bit 5V
Keyboard	Standard PS/2

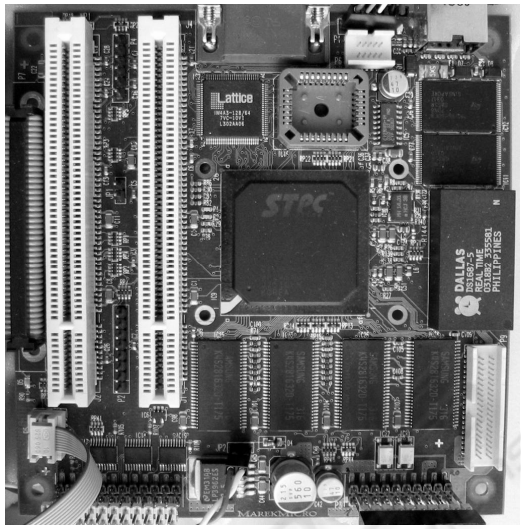

Figure 2: Hardware architecture

3.2 Residential Gateway Software Architecture

The residential gateway software architecture consists of components or entities. There are:
- The drivers component, that deals with the various wide area networks or home area networks connection of the residential gateway.
- The operation system and the Java run time environment
- OSGi framework, that provide an environment for bundle life cycle and bundle execution environment
- Bundle services, such as HTTP service, preferences, log service, device manager, and administration.

This implementation is the entity of OSGi service delivery platform, which the software is embedded in the residential gateway to fulfil the services and to integrate the various home networks, appliances, and technologies.

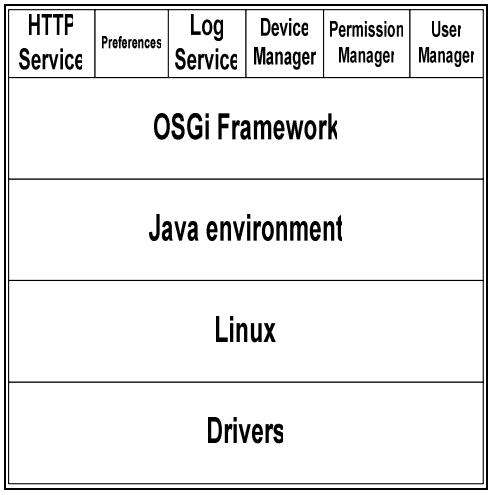

Figure 3: Software architecture of residential gateway

4 CURRENT STATUS AND FUTURE WORK

Our system has been written with Java, and we adopted OSCAR, the free implementation of OSGi in order to build our OSGi service gateway specification. This residential gateway is currently running on the Linux operation system (we use Blackdown's java virtual machine). And we are integrating the LDAP server on our machine in order to build the OSGi backend system, that enable the gateway operator to manage the gateway and deliver the services to the end users, that will give a backup for the functions like configuration (Upgrade, update, and search machine) .This system is the other part to complete the OSGi service delivery system. Both of the LDAP server and the HTTP server will be built on the same residential gateway. There will be then coordination between the LDAP and OSGi to regulate the residential gateway.

There will be not only the basic services bundles, but also the LDAP bundle as the bundle service on the OSGi framework. This bundle will cooperate with the other bundle services. When the user wants to search or to operate the services, he has to make the authentication first. It will increase the security of the smart home environment.

5 CONCLUSION

In this paper we presented our research progress, which is the design and implementation residential gateway for the smart home application, which uses OSGi. This residential gateway, which integrates the home networks, appliances, protocols and technologies, is built with the integration and service oriented perspective.

The backend OSGi system and the service gateway platform, which is realized as the residential gateway software architecture, are the entities to adopt the OSGi service delivery system, which integrates several home networks, appliances, protocols and technologies, and to fulfil the needs of the end users.

The use of LDAP protocol and server, and the bundle LDAP as one of the services of service gateway on the residential gateway will give a positive influence and added value in data base management and security aspects.

REFERENCES

D.Valtchev and I.Frankov,*"Service Gateway Architecture for a Smart Home"*, IEEE Comm. Magazine, Apr.2002, pp.126-132.

Open service Gateway Initiative Overview, *"specification Overview"*, http://www.osgi.org/resources/doc/spec_overview.pdf

N.Zervos et al, *"Hardware Architectures for The Efficient Implementation of Multi-Service Broadband Access and Multimedia Home Networks."* Telecommunication System 23; 34 ,2003, pp.352-367

V.Lawrence,*"Digital Gateways for Multimedia Home Networks"*, Telecommunication System, 23:3, 4, 2003, pp.335-349

Markt & Produkt, *"Konzepte für die Heimvernetzung"* rfe Sept. 2002, pp.18-20

Funkschau,*"Das vernetzte Haus gewinnt Profil"*, Apr.2002, pp.50-52

Markt&Technik,*"Gebäudeautomation-rundum sichern"*, Nr.37,Sept.2002, pp.20-22

OSGi Service Platform Release 3, IOS Press, Amsterdam, March 2003.

K. Hofrichter, *"The Residential Gateway as Service Platform"*, IEEE International Conf. On Consumer electronics, pp. 304-305, USA 2001

Howes, A.Timothy, 1999, *"Understanding And Deploying LDAP Directory Services"*, MacMillan Network Architecture And Development Services, Indianapolis, USA

Mark Wilcox, 1999, *"Implementing LDAP"*, Wroxx Press Ltd, Birmingham, UK