Design and Development of an Interoperation Framework in a Smart Space Using OSGi

Soma Bandyopadhyay
Innovation Lab, Kolkata
Tata Consultancy services
Soma.bandyopadhyay@tcs.com

Naga Kiran Guddanti
Innovation Lab, Kolkata
Tata Consultancy services
kiran.naga@tcs.com

***Abstract*—This article presents an approach towards interoperating i.e., working together with heterogeneous objects in a ubiquitous computing environment. A framework for interoperation in a ubiquitous environment as a ubiquitous service is proposed here. Smart spaces and environments like smart homes, smart healthcare system, and smart vehicular systems are the important application areas of ubiquitous computing. Here the smart home is considered as smart space and home gateway as a smart controlling device capable of performing interoperation among diverse objects inside the home. Proposed interoperation-framework service resides in this controlling device/gateway. The Open Services Gateway Initiative (OSGi) framework is used for developing this interoperation framework. Creation of service provider bundle and user bundle for the associated service interoperation-framework are depicted in detail with case study and results. The same concept can be extended in any smart environment to achieve a generic framework for interoperability to connect heterogeneous devices and to address multiple services.**

Keywords-Ubiquitous computing; Home gateway; OSGi; Smart sapce; Interoperation.

I. INTRODUCTION

Smart space comprises a number of diverse devices having different physical communication mechanism like wired, wireless, power line etc. as well as it serves multiple applications. Interoperation is an important attribute to build up any smart space. There have been a lot of researches made in building up a smart space like smart home, smart healthcare and vehicular system mainly to meet the challenges on interoperation, adaptation, addressing the context of the situation, and security aspects of the said ubiquitous environments. Among them interoperation is the most fundamental one to identify/address the heterogeneous devices and to define protocol of interoperation based on the semantics and dynamical adaptation. Therefore, there is a need for procedures/services for performing interoperation among diverse devices dynamically.

In this article, the focus has been given to design and develop an interoperation framework as OSGi services based on OSGi framework. Here this interoperation service acts as gateway to exchange data among the devices situated inside a home. Every device communicates to each other as well as gets data from Internet through this gateway device only. It is the only device where OSGi service is used to run. Participating devices need only standard applications, web interface, simple J2ME (java2 micro edition) based applications to perform interoperation. Hence this solution is easy to be deployed and cost effective. Also this interoperation solution is easy to manage since 1) inclusion of any new device seeking for interaction with any other participating device requires modification in interoperation framework inside gateway device only, 2) minimal modification may be required in the existing solution if the new device falls within the class of any existing participating device, 3) failure of any participating device is easily detectable here, 4) an inventory of active interoperating devices can be made readily available and Web based software updating among the participating devices can be done easily, and 5) any service addition or feature enhancement inside the proposed solution involves the only central device, i.e., this gateway device.

The case study presented here acts as RTP/RTSP (real time transport protocol/ real time streaming protocol) video data forwarder between two devices communicating only through this interoperation service. The devices in the above mentioned case study have different physical layer communication mechanism.

The remainder of this article is organized as follows. First, the related work in interoperation in smart space is presented, followed by an overview of the proposed system. The implementation and experimental study are then described in detail. The final section concludes this article with future scope of the current proposed model.

II. RELATED WORK

Interoperation in a smart space using OSGi (Open Services Gateway Initiative) framework is broadly studied in cases like 1) distributed peer to peer model [1] where OSGi platforms are required to run on multiple devices or participating nodes to distribute device dependent services over several devices, 2) mobile agent technology [2] with a prerequisite that every device has an OSGi based agent embedded in it, and 3) using R-OSGi (Remote – OSGi) in the devices for interoperation [3]. All these cases require OSGi to run in all the participating devices, this demands every device needs JVM (java virtual machine) and operating system including Bluetooth, Zigbee based devices which have limited computation power and resources. But it is unrealistic to demand that every device has an OSGi based services implanted in it. At the same time these approaches do not specify how to manage the system.

In this paper a framework for interoperation for a smart space is proposed. This is generic in nature. It runs inside a central location, where participating devices do not require any OSGi based application. This solution is cost effective, easy to deploy, and manage. Here focus is given to the design and development of an interoperation framework consisting of multiple layers that works as a middleware stack, interconnects various heterogeneous devices, and acts as a gateway of transferring any data among them. It designates unique identifier to the participating devices. Addition of new functionality to this framework can be performed easily without modification in the interoperating devices.

III. SYSTEM OVERVIEW

The interoperation service as proposed here is developed on OSGi [4] middleware. It maintains diverse connectivity to multiple heterogeneous devices like UPnP (universal plug and play) [5], Bluetooth [6], Zigbee [7], Wi-Fi [8], and Ethernet. These devices exchange data through this proposed system, where the system acts as gateway device, and also acts as router as well as control point of the participating devices (Device1, Device 2 … Device 5 as depicted in Fig. 1), which desire interoperation between each other and thus an interoperation zone is established as depicted in Fig. 1.

The proposed system designates a unique identifier to each device during the time the devices join into the interoperation zone. It has interfaces for demanding any registered service – example streaming video, playing movie, sharing images. It has also interfaces to add new services.

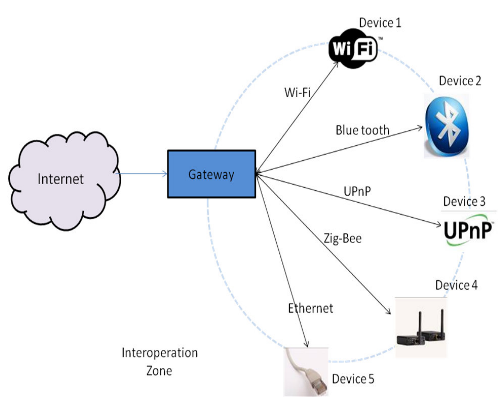

Figure 1. Interoperation network

Participating devices use readily available agents (with modification if necessary) to demand services from the proposed gateway or interoperation system, by means of the interfaces exposed by the same.

IV. SYSTEM INPLEMENTATION

This section portrays the system implementation in detail. The proposed interoperation framework is developed by using Knopflerfish [9]-OSGi framework. It runs as OSGi service. It has both service-provider and service-user bundle [10]. The overview of the proposed architecture is depicted in the Fig. 2.

The proposed system comprises the following multiple layers:

- Interface
- Core control (Threaded, interoperation control module, with multiple event handlers)
- Services

The interface layer provides the APIs (application programming interface) to interact with the interoperation framework viz. 1) joining the interoperation zone and getting registered with proposed system, 2) obtaining a unique identifier, 3) requesting various services. Interface layer uses message producer, topic publishing mechanism for handling events. Some APIs are: 'interopStart' - that starts the core control layer of the interoperation framework, 'interopStop' - that stops the proposed service.

Core control layer, the main component of interoperation framework, runs as thread. It triggers the services requested by the diverse devices after interaction with the service layer and after communicating with concerned communication-driver for interacting those devices.

Device detection and providing of a unique identifier to these devices are some of its main functionalities. Here IPv6 (Internet protocol version 6) address is assigned as unique identifier.

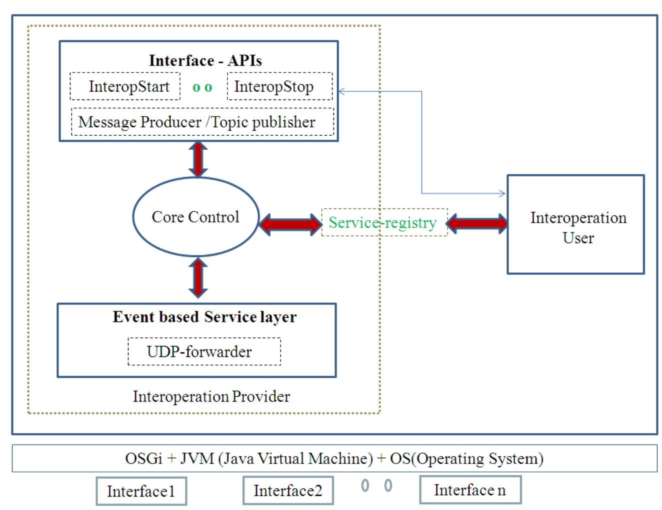

Figure 2. Functional blocks of interoperation framework

This core module registers events, associates services with these events, and performs the event handling on getting a notification of the occurrence of an event. It interacts with the service registry of the OSGi framework for interacting with other OSGi service bundles. It communicates with the multiple interfaces like Bluetooth (for communicating with mobile), Wi-Fi, Zigbee, UPnP through the operating system for exchanging data with heterogeneous devices in interoperation zone.

The service layer performs the service specific functionality as activated by the core layer. The services are furnished to the concerned participating device which invokes/requests that service by using APIs provided by the interface layer. Services act as event handler. Here, the point to be noted is that gateway on which this interoperation framework runs is the only common point among any devices for interoperation. Some of the service examples are - 1) acting as a data-forwarder among any devices, 2) video streaming from Internet, from local video server, 3) sharing non real time data, 4) getting notification of receiving SMS (short message service), email in TV (Television) while watching it.

The service user bundle of the proposed system invokes the interoperation provider and starts the interoperation framework in active mode. Following blocks summarize the steps which need to be invoked sequentially to create a service provider bundle and user bundle here.

Creation of service provider bundle:

1. Creation of service interface
    - Creation of service interface package
    - Defining the service interface class
2. Creation of service provider & implementation of service interface class
    - Creation of service provider
        - Creation of a service implementation package
        - Importing the service interface class (defined in service interface creation in step- 1)
        - Implementing the provider class
    - Implementation of service interface class
        - Use the same service implementation package
        - Importing the service interface class (defined in step-1)
        - Implementation of the interface class
3. Package of interface class to be exported to manifest.mf file

Creation of service user bundle:

1. Creation of service interface
    - Same as mentioned in service provider
2. Creation of service user class
    - Creation of service user
        - Creation of a service user implementation package
        - Importing the service interface class (defined in service interface creation)
        - Implementing the service user class

V. EXPERIMENTAL RESULTS

The complete system is implemented by using an Intel desktop-board with Linux where the interoperation is performed by using one Wi-Fi and one Ethernet interface. A local video server connects over Ethernet and a laptop connects over Wi-Fi / Ethernet with the proposed system. The software and hardware configuration details are shown in table 1 and table 2 respectively. The experimental setup is depicted in Fig. 3.

During experimentation interoperation services are executed first. The laptop registers with interoperation framework and requests RTP/RTSP video streaming service specifying the port number. Here port 3000 is used as receive port for the video service. A manual setup has been made at the local video server side with IPv6 address of the proposed system and its associated port number 2500 needed for UDP (user datagram protocol) data forwarding service. The local video server and laptop use VLC (video LAN client) media player [11], Fig. 4 and Fig. 5 represent the setup respectively. Fig. 6 depicts the interoperation framework in running condition.

 ISBN: 978-1-61208-100-7

TABLE I. SOFTWARE CONFIGURATION

The software Configuration	
Software configuration	***Description***
OSGi Framework	Knopflerfish 2.3.3
Java Virtual Machine	JDK 1.6
Operating System	Linux - Ubuntu 9.10
Java-IDE	eclipse 1.2.2

TABLE II. HARDWARE CONFIGURATION

The Hardware Configuration	
Hardware configuration	***Description***
Target platform	Intel® Desktop Board D945GCLF2
Processor	Dual-Core Intel Atom Processor 330 integrated at 1.6GHz

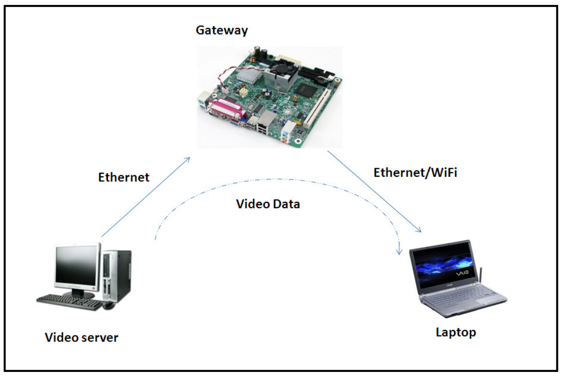

Figure 3. Experimental setup

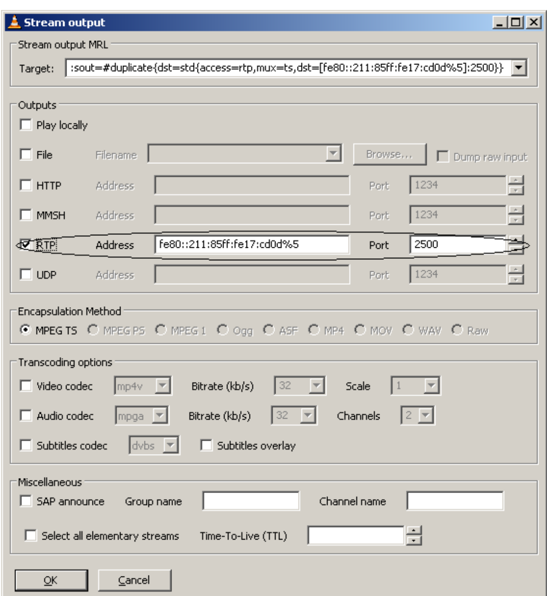

Figure 4. VLC media player setting (video server side)

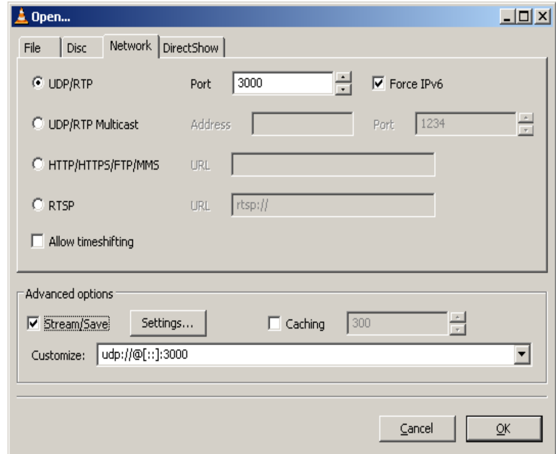

Figure 5. VLC media player setting (Laptop)

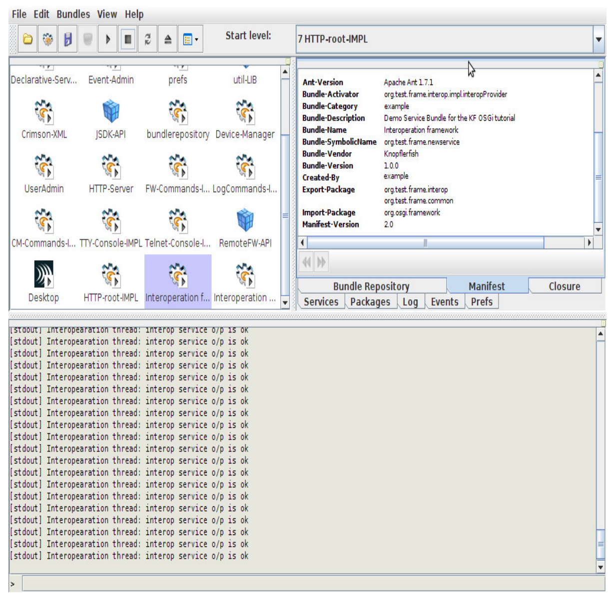

Figure 6. Interoperation with UDP forwarder service

VI. CONCLUSION AND FUTURE WORK

In this article, design and development of an interoperation framework for a smart environment like smart home consisting of heterogeneous devices with different physical communication scheme like Bluetooth, Wi-Fi, Ethernet, UPnP, Zigbee have been proposed. The framework acts as control point as well as a router among any devices. It does not require any corresponding OSGi module at the heterogeneous devices seeking for interoperation. It provides interfaces to add new services into its service layer. Therefore it is easy to manage, cost effective and easy to be deployed in real world.

Suggested system is developed based on Knopflerfish's OSGi framework and runs as OSGi service.

Video streaming based on RTP/RTSP from a local video server through the proposed interoperation service is

 ISBN: 978-1-61208-100-7

depicted as a use case. Besides, the methods of writing bundles for service provider and service user are discussed.

The limitation of the proposed system lies in its incapability of doing complete automation along with identification of context. However, the proposed system can be enhanced further by providing web based GUI (graphical user interface) for the purpose of automatic triggering of any existing service or to add new services and to make it context aware with limited use of sensors and considering mobile as the essential resource of user context data .

There is also scope for further work in making the complete system generic so that it can be used in any smart space, example - as a part of a car gateway. We are continuing to carry researches on the above mentioned future scope of work and enhancement of the proposed system.

REFERENCES

[1] Chao-Lin Wu, Chun-Feng Liao, and Li-Chen Fu, "Service-oriented smart home architecture based on OSGi and mobile agent technology", Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transaction, vol.37, issue 2, March 2007, pp. 193-197,doi: 10.1109/TSMCC.2006.886997, 22.03.2010.

[2] Seung Keun Lee, Jeong Hyun Lee, "OSGI based service mobility management for pervasive computing environments", 24th IASTED Internation Multi conferecnce on Internet and Multimedia sytems and Applications, Feb 2006,pp. 159-164, 05.04. 2010.

[3] Jan S. Rellermeyer, Michael Duller, Ken Gilmer, Damianos Maragkos, Dimitrios Papageorgiou, and Gustavo Alonso "The software fabric for Internet of Things", Lecture notes in computer science, Proceedings of the 1st international conference on The internet of things, 2008, pp. 87-104, 05.04. 2010.

[4] OSGi, http://www.osgi.org/Specifications/HomePage, 10.04.2010

[5] UPnP, http://www.upnp.org,10.04.2010

[6] Bluetooth, http://www.bluetooth.com, 17.07.2010

[7] Zigbee, http://www.zigbee.org/, 17.07.2010

[8] Wi-Fi,http://www.wi-fi.org/certified_products.php,17.07.2010

[9] www.knopflerfish.org, 10.04.2010

[10] Choonhwa Lee, David Nordstedt, and Sumi Helal, "Enabling Smart Spaces with OSGi", IEEE CS and IEEE ComSoc, PERVASIVE computing journal, vol.2, issue 3, pp. 89 -94, doi: 10.1109/MPRV.2003.1228530.computing journal, vol.2, issue 3, pp. 89 -94, doi: 10.1109/MPRV.2003.1228530, 24.03 2010.

[11] http://www.vlcmediaplayer.org, 10.04.2010

 ISBN: 978-1-61208-100-7