國立交通大學

資訊科學與工程研究所

碩士論文

基於訊息導向中介軟體之異質環境家庭自動化平台

A MOM-based Home Automation Platform in Heterogeneous Environments

研究生：陳俊元

指導教授：袁賢銘　教授

中華民國九十五年七月

基於訊息導向中介軟體之異質環境家庭自動化平台

A MOM-based Home Automation Platform in Heterogeneous Environments

研究生：陳俊元　　　　Student：Chun-Yuan Chen

指導教授：袁賢銘　　　　Advisor：Shyan-Ming Yuan

國立交通大學
資訊科學與工程研究所
碩士論文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年七月

基於訊息導向中介軟體之異質環境家庭自動化平台

研究生：陳俊元　　　　　指導教授：袁賢銘

國立交通大學資訊科學與工程研究所

摘要

近年來已有多種之可支援家電控制之家庭網路技術，如 X10、UPnP、HAVi 等；但採用不同家庭網路技術的家電無法互通，也無法組成進一步合作構成家庭自動化系統。目前已出現許多異質家庭網路互通性之研究；但有關異質環境之家庭自動化研究依然十分匱乏。本研究提出一家庭自動化平台；使支援不同家庭網路技術之家電得以可動態合組家庭自動化系統。異質環境下之家庭自動化系統須面對家電規範彼此不相容、難以支援事件驅動類型之自動化動作、可靠性、延展性、擴充性與重新組態之問題。本平台底層遵循開放服務閘道器規範 (Open Service Gateway Initiative)，以訊息導向中介軟體作為核心整合所有家電網路。本平台以根據家庭自動化規則進行之訊息路由為方法，完成可動態組態之事件驅動類型家庭自動化；家電可動態加入或退出平台而不影響正在其他進行之自動化動作。本平台以持續性之訊息與訊息訂閱達到高可靠性，透過一對多之訊息模式與跨越子網路之訊息提供高延展性。平台本身由彼此獨立之服務合作組成，所有服務之軟體元件均可根據彼此之相依關係透過網際網路動態安裝、啟動，具備高擴充性。本平台並以基於可延伸標記語言(XML)之描述子抽象化家電服務，增加系統之彈性，減少軟體之更新需求。家庭自動化可透過本平台延展至網際網路，與網際網路服務結合，便利生活。

A MOM-based Home Automation Platform in Heterogeneous Environments

Student：Chun-Yuan Chen　　Advisor：Shyan-Ming Yuan

Department of Computer Science
National Chiao Tung University

Abstract

While there have been many home networking technologies such as INSTEON, UPnP and HAVi, appliances supporting different home networking technologies cannot collaborate to finish Home Automation (HA). Recently, many studies of interoperability among heterogeneous home networking technologies have been done, but researches on further Home Automation in heterogeneous environments are still critically lacking. This paper proposes the MOM-based Home Automation Platform (MHAP), which accomplishes event-driven Home Automation in incompatible home networks. MHAP is independent of any home networking technology and integrates home networking technologies in home gateway. For users, MHAP provides the easy-to-use and standardized way to configure complex Home Automation scenarios by rules. Through introducing Message Oriented Middleware (MOM) and Open Service Gateway Initiative (OSGi), MHAP offers reliable automatic operations, fault tolerant and reconfigurable Home Automation, high extensibility and large scalability. The large scalable collaboration in MHAP is among appliances, other MHAP gateways and Internet services such as Web Services.

Acknowledgements

The author is indebted to Dr. Shyan-Ming Yuan, his advisor, Mr. Chi-Huang Chiu, Mr. Ming-Chun Cheng, Mr. Ruey-Shyang Wu and Mr. Tsun-Yu Hsiao, the candidates for Ph.D. of the Department of Computer Science, National Chiao-Tung University, Dr. Ping-Jer Yeh of Computer Science, National Chiao-Tung University, and Miss Hui-Wen Lin of MediaTek Inc, for their assistance and helpful comments in this research. The author especially thanks to Mr. Ming-Chun Cheng for his thoughtful reviews of the earlier drafts of the manuscript.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

In the last three decade, the science and technology for accomplishing Home Automation (HA) have progressed tremendously. On the other hand, with proposing of Digital home, many digital appliances may increasingly join the Home Automation system at home. Home Automation makes human life convenient through connection and collaboration between appliances and sub systems at home such as secure system [1]. For collaboration, such smart appliances and sub systems use home networking technology [2] to communicate with each other. Therefore, there have been more then twenty different kinds of home networking technologies on market such as X10 [3], INSTEON [4], UPnP[5] and HAVi [6].

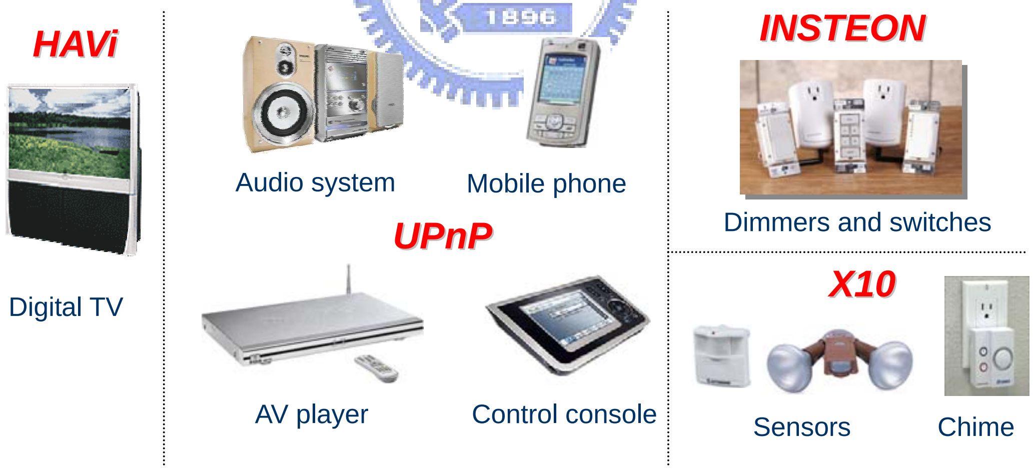


Figure 1-1 appliances supporting different home networking technologies

Although there are many appliances supporting different home networking technologies (Figure 1-1), building a Home Automation system should use the appliances supporting one of them traditionally. A home network middleware doesn’t understand the messages of the other home network. Even the appliances in one home network couldn’t physically connect to the other home network. However, in Digital Home environment, a home gateway such as a PC or a set-top-box can physically connect to each kind of home network. Recently, there have been substantial studies performed on one-to-one and one-to-many interoperability of heterogeneous home networking technologies. While some studies on device control in heterogeneous environments have been done, researches on Home Automation in such situation are still critically lacking.

1.1 Motivation

Although a home gateway can physically connect each home network, the appliances can’t communicate with each other because of the low interoperability of existing home network technology. For example, even a power line network connects to a wireless LAN through a home gateway; the event message from an UPnP media player is not understood in X10 network. Furthermore, a Home Automation system can’t consist of appliances in heterogeneous environments. Therefore, this paper proposes the MOM-based Home Automation Platform (MHAP), which integrates all home networks and offer Home Automation in heterogeneous environment.

1.2 Objective and System Goals

The objective of this study is to integrate all home networks and offer Home Automation in heterogeneous environment. Therefore, the study proposes a MOM-based Home Automation platform (MHAP) in heterogeneous environments to achieve the objective.

Figure 1-2 shows the use cases of a basic Home Automation system in heterogeneous. The rule manager is the user configuring the scenario of automation. The appliance operator is the user controlling the appliances. The system administrator is the user managing the system. The appliances includes X10, Jini and LonWorks appliances. The external systems participating the Home Automation include location, temperature, motion detect and time system.

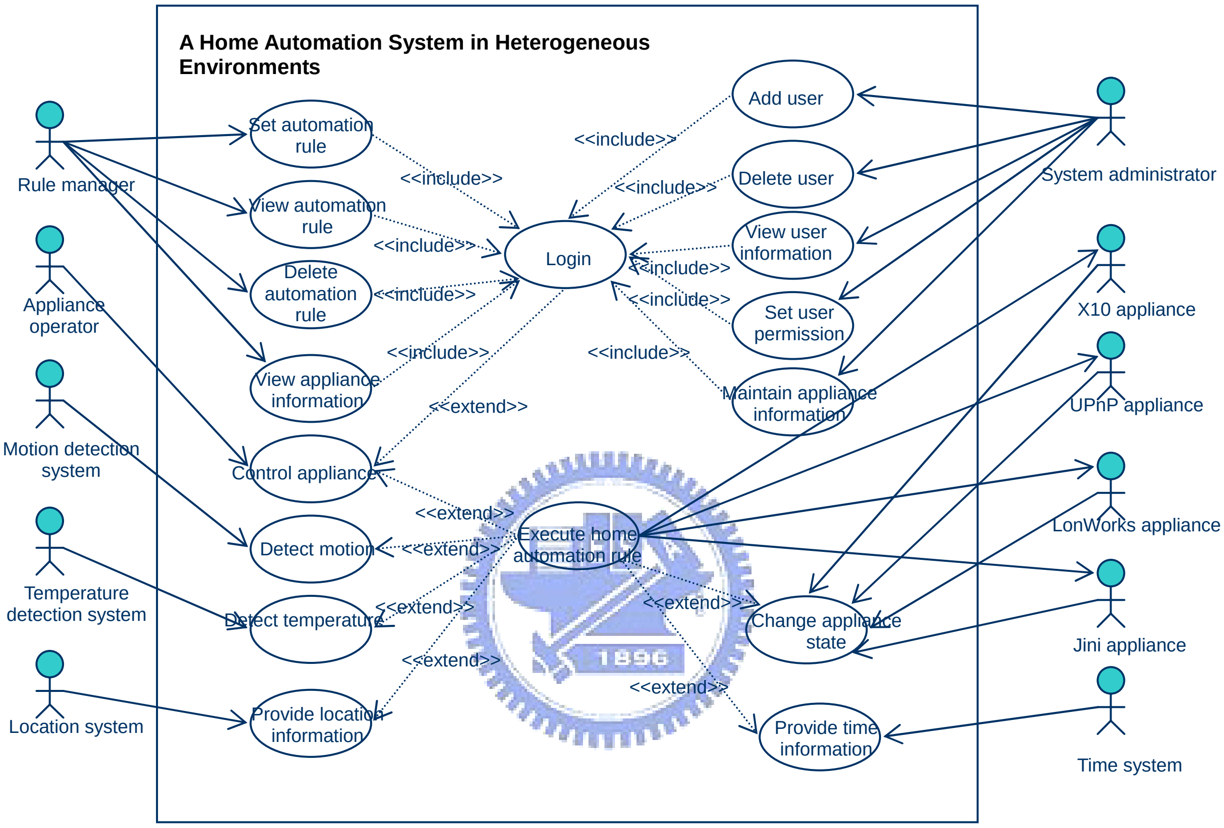


Figure 1-2 the use cases of a Home Automation System in Heterogeneous Environment

As Table 1-1 indicates, an ideal Home Automation system offers easy-to use configuration and control functions for user. Therefore, the user can reconfigure the scenario of Home Automation every time when his requirement changes. The user can also control and monitor the appliances and the subsystems at home through a consistent method. They needn’t prepare different controllers to control different appliances or subsystem.

Table 1-1 configuration and control use cases for a Home Automation System

Use Case	Actors	Description
Login	Rule manager, System administrator, Appliance controller	*The User uses account and password to login to MHAP.*
Set automation rule	Rule manager	*The user sets the rules for Home Automation.*
View automation rule	Rule manager	*The user browses the rules for Home Automation set before.*
Delete automation rule	Rule manager	*The user removes the specified rule for Home Automation set before.*
View appliance information	Rule manager	*The user browses the states of specified home appliance.*
Control appliance	Appliance operator	*The user sets the states of specified home appliance.*

On the other hand, a Home Automation system requires the administration mechanism to manage the user permission and device information. Table 1-2 indicates the function of administration in a Home Automation.

Table 1-2 administration use cases for a Home Automation System

Use Case	Actors	Description
Add user	System administrator	*The system administrator adds the account for new user.*
Delete user	System administrator	*The system administrator deletes the account of specified user.*
View user information	System administrator	*The system administrator browses the account of specified user.*
Set user permission	System administrator	*The system administrator sets the permission of specified user.*
Maintain appliance information	System administrator	*The system administrator sets URL for the profiles of the appliances.*

An ideal Home Automation system offers both the scheduling and event-driven automation. As Table 1-3 describes, the appliances can cooperate with any other subsystem or appliances even if they are supporting incompatible home network technologies.

Table 1-3 device automation use cases for a Home Automation System

Use Case	**Actors**	**Description**
Detect motion	Motion detection system	*External motion detection system detect the change of motion states and then MHAP checks if any Home Automation operation should be finished according to the rules.*
Detect temperature	Temperature detection system	*External temperature detection system detect the change of temperature states and then MHAP checks if any Home Automation operation should be finished according to the rules.*
Provide location information	Location system	*External Location system detect the change of location states and then MHAP checks if any Home Automation operation should be finished according to the rules.*
Execute Home Automation rule	Appliance operator, Motion detection system, Temperature detection system, Location system	*The MHAP executes Home Automation operations to control the X10 appliance, UPnP appliance, LonWorks appliance or Jini appliance.*
Change appliance state	X10 appliance, UPnP appliance, Lonworks appliance, Jini appliance	*The appliances change it state by itself, and then MHAP checks if any Home Automation operation should be finished according to the rules.*
Provide time information	Time system	*External time system detect the change of time states and then MHAP checks if any Home Automation operation should be finished according to the rules.*

To meet all the Home Automation requirements, the MHAP has six main design goals indicated and explained as follows:

1.2.1 Event-driven Home Automation

In a Home Automation system, events represent the changes of states of appliances or sub systems. Some Home Automation operations need events to trigger them. Therefore, a complete Home Automation platform should support the event-based Home Automation. MHAP is proposed to provide the events across different home networks. To take a simple example, in MHAP, the change of the power state of an UPnP media player is an event which may trigger another Home Automation operation like closing an X10 light. With such mechanism, the person at home can close the light in living room by powering off the television in bedroom.

1.2.2 Protocol and Platform Independence

Another goal of MHAP is being device and protocol independent. Therefore, Home Automation not limited by the kinds of home networking technology. MHAP is designed for heterogeneous environments and over any existing or new device and protocol. Any kinds of the appliances capable of any home networking technology such as UPnP multimedia players, X10 motion sensors or INSTEON dim lights could join a MHAP system. When adding an appliance which uses an additional protocol, it can also cooperate with the existing appliances to finish Home Automation through MHAP.

1.2.3 Reliability and Fault Tolerance

Reliability and Fault Tolerance: Certain kinds of Home Automation operations, especially the operations of security systems, must be reliable (Figure 1-3). For example, there were a series of Home Automation operations executing, which automatically lock the door at 11:00 PM. If a child shut down the home gateway at that time, the MHAP should continue to finish the previous locking operation after system restart.

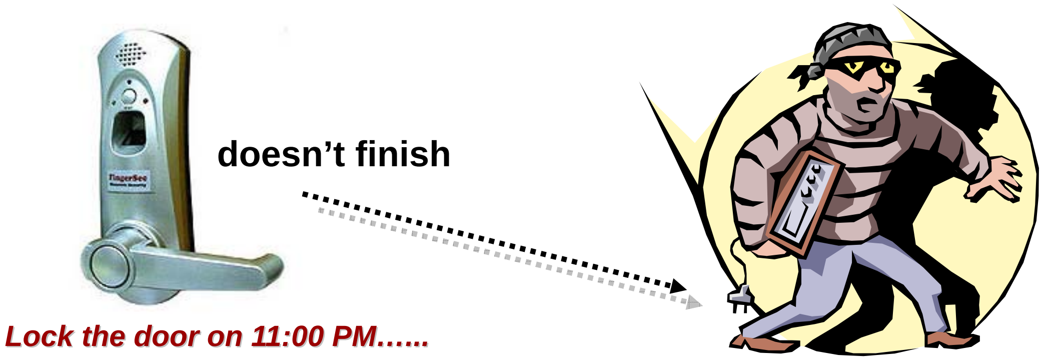


Figure 1-3 reliability requirement of Home Automation System

On the other hand, when some appliances or certain part of the network fail, the system should be still effective because the Home Automation operation may relate to security closely. The MHAP should offer fault tolerant function to ensure the system's working to handle the situation, especially when cooperating with Internet services.

1.2.4 High Scalability

People may buy new appliances and the appliances are joining Home Automation increasingly. However, increasing of the number of appliances at home should not apparently affect the performance of the MHAP. On the other hand, MHAP is designed to scale from home domain to building, campus and Internet. That is, the appliances at home can cooperate with other appliances at the building or an Internet service to facilitate human life.

1.2.5 Extensibility

High extensibility is an important goal of the MHAP. There may be some appliances only supporting additional home networking technology on market. The extensibility of MHAP makes new smart appliances and new home networking technologies join the MHAP dynamically.

1.2.6 Easy to Use

MHAP is designed for providing human reconfigurable Home Automation and human convenient way to configure Home Automation at home. Every change of the Home Automation setting or scenario needn't complex processing such as downloading separate macro program or configuration to each appliance. On the other hand, MHAP is proposed to provide convenient service deployment and management mechanisms. The Home Automation software needs update when the user plugs the additional appliances into the platform. MHAP will automatically download the updating software bundle from service provider through internet. When updating and activating the additional bundle, the other service running will not be affected. Besides, the user can easily install, activate, deactivate and uninstall the software bundle from MHAP without stopping all services.

1.3 Problems

Although different home networks can be physically connected through home gateway, MHAP needs to solve problems on Home Automation in heterogeneous environment.

1.3.1 Low Interoperability of Home Networking Technologies

Most home network technologies have low interoperability. Although connecting physically, the message of one home networking technology can't be understand by the other home networking technology. For example, even the power line network used by connected to a wireless LAN used by UPnP, the event message of UPnP could not be understand in X10 network.

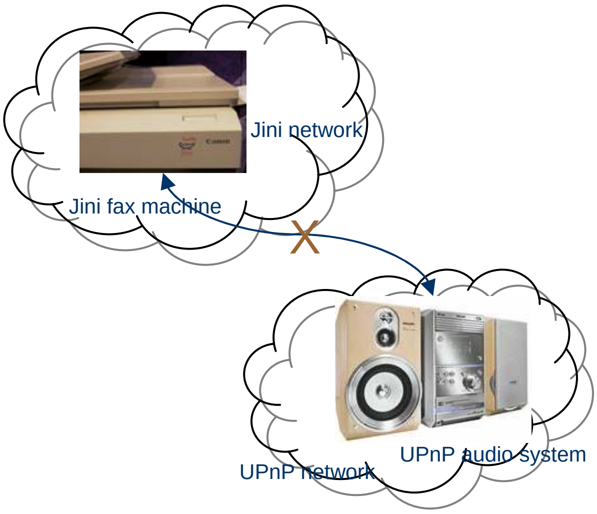


Figure 1-4 low interoperability of home networking technologies

1.3.2 Hard to Support Event-driven Model

A complete Home Automation system needs event-driven model to trigger the device operation. But in heterogeneous home networking environments, low interoperability of the home networking technologies led to the absence of the events passing through different home networks.

1.3.3 Reliability

Such a Home Automation system needs reliability. As the automation operation interrupted for some reason such as fault of the home gateway, the reminder operation may need to be finished. Therefore, MHAP needs to take extra effort to make both the operation and the event triggering the operation reliable.

1.3.4 Scalability and Extensibility

A Home Automation system should consider the scalability problem. The number of appliances at home will constantly increase. If the Home Automation system can’t scale well, it will influence the automation operations or even crash the system. Sometimes people may hope that the automation scales to Internet. However, the home networking technologies such as X10 and INSTEON can't scale to Internet. The MHAP should solve the problem caused by the existing home networking technologies. Besides, the MHAP has to make the device be identified by Internet Service. In the future, new home networking technologies may be proposed. A Home Automation system should work with the appliances of the networking technology. A Home Automation system in heterogeneous environment needs to solve such extensibility problem.

1.3.5 Deployment and Management

A Home Automation system may need to be updated for support new appliance or home networking technology. The internet accessibility provide home gateway the opportunity, but different service provider may provide different update method or software bundle. While updating, maybe the Home Automation can't be interrupted or stopped. The deployment and management problems may exist in a Home Automation system.

1.4 MHAP Solution

To accomplish the goals and solve the problems, MHAP solution introduces the Message Oriented Middleware (MOM) [7] into Home Automation systems and provides event-driven Home Automation. Figure 1-5 represents the MHAP solution. Inside home, MHAP on home gateway connects each home network and the rule engine service performs as an application router. MHAP accomplish Home Automation through platform and protocol independent MHAP message flow. Application router handles the message flow among the appliances depends on the Home Automation rules of MHAP.

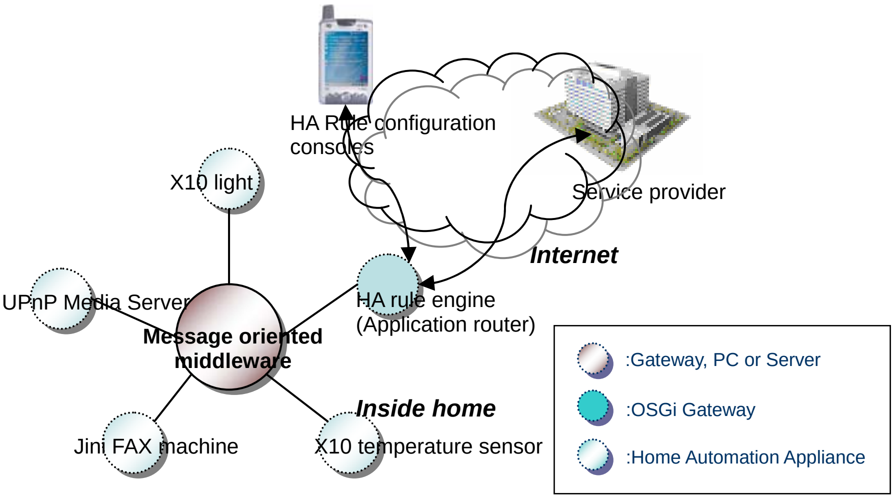


Figure 1-5 overview of HMAP

MHAP is entirely deployed on the standard deployment and service management infrastructure, for example, Open Services Gateway initiative (OSGi) [8]. Therefore the services or software bundles on MHAP could be installed and updated through Internet from service provider and easily managed.

1.4.1 Message Oriented Middleware (MOM)

MHAP is based on Message Oriented Middleware (MOM). The MOM introduces the event-driven into the platform since the event is a special message in MOM. The model of MOM in MHAP is publish/subscribe. This kind of model scales well while the receiver such as appliances increases. Because different appliance subscribes different channel in MOM, plugging in new appliance will not influence the Home Automation currently running in MHAP. Even adding the protocols and appliances of the new Home Automation technology will not influence the system. Furthermore, the durable subscription and persistent message of MOM makes messaging reliable. With MOM core, MHAP support the reliable operation required in Home Automation systems. MHAP uses Java Message Service, the standard MOM interface for Java, to access all function of MOM. Therefore, different MOM providers can be chosen to gain different degrees of performance of MHAP.

1.4.2 Standard Deployment and Service Management Infrastructure

All MHAP components are deployed on the standard deployment and service management infrastructure which is OSGi presently. Through the OSGi, software or service bundles on MHAP can be transferred from service provider through Internet, and the software and hardware provider can developer their own products without influencing each other. Deploying on such infrastructure makes MHAP highly extensible. Furthermore, the services on MHAP such as protocol adaptor can install independently without stop all the system or current automation operation.

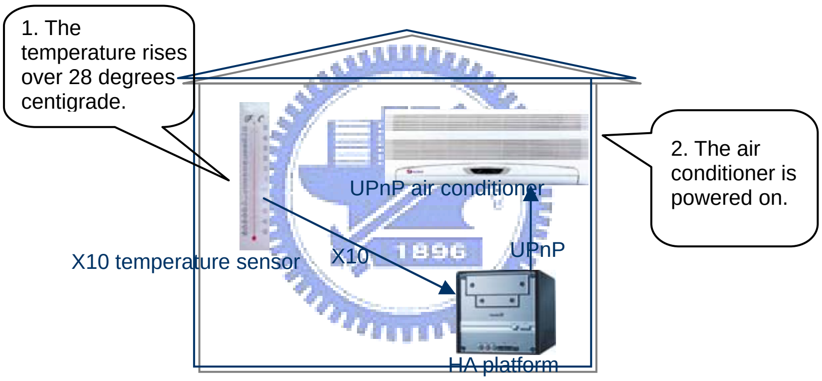

1. The temperature rises over 28 degrees centigrade.
2. The air conditioner is powered on.
UPnP air conditioner
X10 temperature sensor
X10
UPnP
HA platform

Figure 1-6 power on the air conditioner when the temperature rises over 28

1.4.3 Rule Based Home Automation

MHAP provides the rule services instead of the traditional API or program. The Home Automation operation is indicated by rule. For example, a user can set the rule “Power on the air conditioner when the temperature rises over 28” with various kinds of configuration console. Then the temperature becoming over 28 degrees centigrade will trigger the Home Automation operation which is to power on the are conditioner.

Figure 1-7 indicates another MHAP rule which three different appliances participate in. If a fax is coming, the Home Automation that CD player starts to play the song “Unchained Melody” and power on the light will be finished.

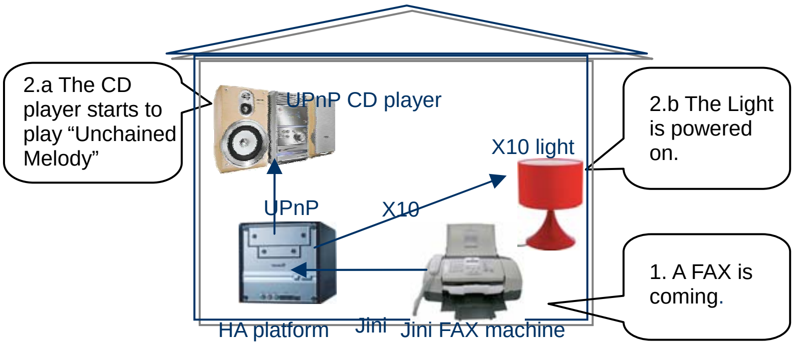


Figure 1-7 power on the light and play a song when a FAX is coming

There is one other thing that is important for HMAP rule. All device and appliances are abstracted by XML descriptors. The HMAP rule which has been set will not need to change even thought the appliances belonging to home technologies added in the future.

Chapter 2 Background and Related Work

2.1 Home Automation

Home Automation (HA) means that the device or appliance at home connects and collaborates with each other to facilitate the human life. A Home Automation system can schedule and automatically operate the appliances in the home network for convenient or secure purpose. Besides, the change of an appliance's state may also cause the change of another appliance's state or trigger some action of another appliance in a Home Automation environment.

Basically, the main components of a Home Automation system include devices with automation programs or services and various appliances being controlled or providing state information. With the vigorous development of Digital Home, the former may be home gateways in the future. The latter are subsystems supporting any home networking technology, for example, the lighting, security system, entertainment, thermostat, irrigation or other appliances.

2.2 Home Networking Technologies

Many home networking technologies [10] recently developed such as UPnP, HAVi, X10, INSTEON, Jini [11] and LonWorks [12] have provided appliances with connectivity. Traditionally, the appliances supporting one of these home networking technologies communicate with each other to finish Home Automation. Because most home networking technologies use different protocols and different kinds of networks usually can't be connected physically, they have low interoperability (Figure 2-1). Therefore, the appliances of a Home Automation system must support the same home networking technology before.

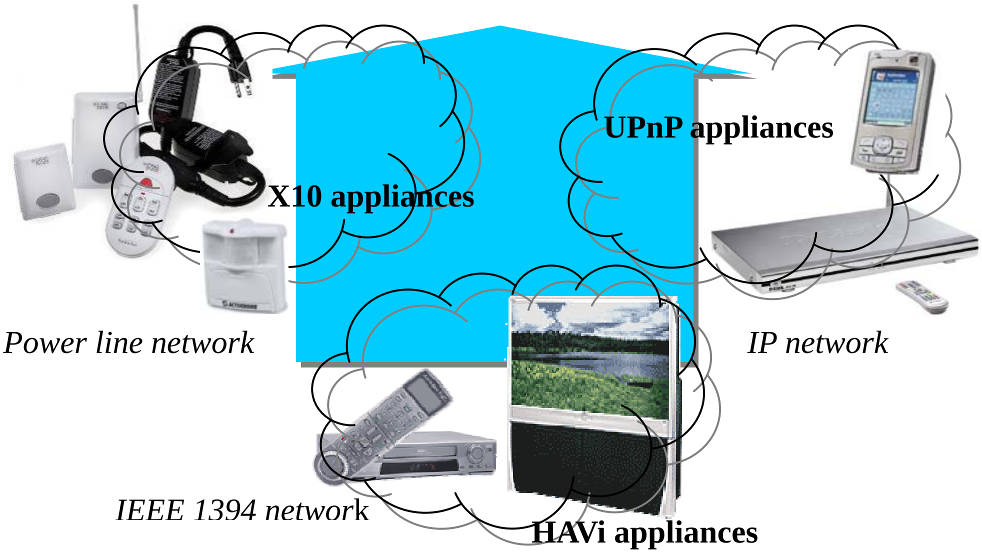


Figure 2-1 X10, IEEE 1394 and UPnP home networks

2.2.1 X10

Pico Electronics Company developed the X10 in 1975. For accomplishing Home Automation, the X10 defines the protocols for communication among devices. It uses the power line as primary physical media of signaling and control and also defined a radio based transport. All appliances and controllers in a network can only transmit one command at a time with X10 singles. The X10 receivers can't decode the command when two signals collide. The traditional X10 protocol is slow compared with other technologies using power line such as LonWorks and INSTEON.

2.2.2 INSTEON

The INSTEON also use electrical wiring and radio frequency to transmit messages like X10. However, INSTEON confirms that every message is received and will automatically resent the message when detecting errors. Each INSTEON device is a repeater to enhance the quality of the network. Although the INSTEON protocol and X10 protocol are not compatible, INSTEON allows manufacturers to develop

INSTEON-compatible products which are X10-ready. Therefore, the kind of INSTEON can work in existing X10 networks. However, the INSTEON devices don’t repeat X10 signals and can’t improve the disadvantage of existing X10 networks.

2.2.3 Universal Plug and Play (UPnP)

UPnP is a distributed, open home networking technology based on existing standards such as Transmission Control Protocol (TCP), Hypertext Protocol (HTTP) and Extensible Markup Language (XML). UPnP offers the architecture to accomplish pervasive peer-to-peer network connectivity of devices such as the personal computer, intelligent and wireless devices [13]. The networking architecture of UPnP is on the top of TCP/IP to control and transfer data among networked devices among the home, office, and public spaces. Therefore, the UPnP are used to discover other devices on the network and remotely control appliances and share information among devices and the World Wide Web. Besides, the UPnP also offers the event subscription mechanism for devices [14]. The UPnP forum is standardizing the device and service with defining and publishing UPnP device and service descriptions. The standardization makes the devices easily connect with each other and simplify the implementation of networks.

2.2.4 Home Audio/Video Interoperability (HAVi)

The home audio/video interoperability (HAVi) is vendor-neutral audio-video standard for home entertainment environment. With IEEE 1394 as the interconnection medium, the architecture of HAVi includes a set of application programming interfaces (APIs) and services. In HAVi environment, different home entertainment devices such as VCRs, security systems connects together and the primary device such as a television can control the others. HAVi associated each physical device with a software proxy [16]. Therefore, adding software proxy can extend the feature of the device. On the other hand, when a user installs a new device, the HAVi system is able to configure itself to accommodate it. The HAVi also offers the function of addressing scheme, resource discovery, posting and receiving events, streaming and controlling isochronous data streams.

2.2.5 Jini

Jini network technology is an open software architecture for building distributed systems which is highly adaptive to change [17]. Jini offers object services to clients and the discovery of the available services [18]. Differing from other distributed object systems, Jini is designed for the dynamic distributed environment, where the services are transient and the network is unreliable. Before the client use the service, the Jini environment downloads the mobile code into the client. The client then communicates with services through the mobile code. The client needn't be re-implement when the protocol changes. The changes of the protocol are private to the service and to the mobile code. Therefore, Jini is the technology suitable for building the evolvable, flexible and adaptive network required in dynamic computing environments such as a home network.

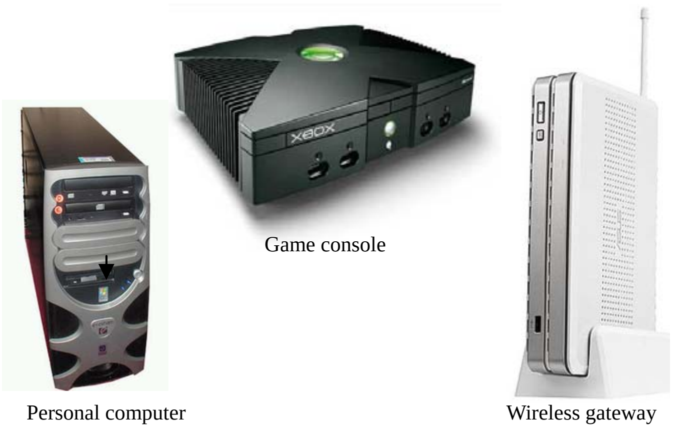


Figure 2-2 different kinds of home gateways devices

2.3 Home Gateway

2.3.1 Home Gateway Device

A home gateway is a device being able to connect to the Internet and to be an entertainment and networking center at home. Therefore it allows communication and control between service providers and the devices within the home network. The communication channels used by the device are seen in home networking technologies such as UPnP and TCP/IP. The communication among different notions of services and service discovery are accomplished by XML-like standard [19]. The hardware of home gateway can be personal computers, set-top-boxes, DSL gateways or game consoles such as Xbox and PS2.

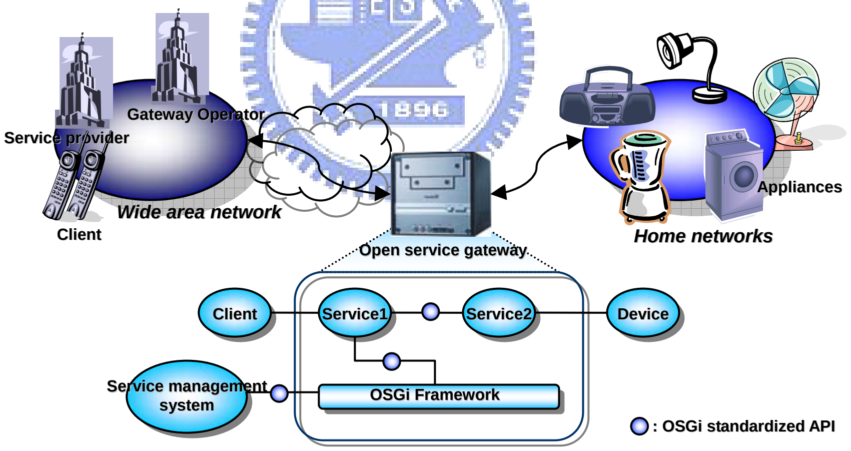


Figure 2-3 OSGi: a case of the service management and deployment infrastructure

2.3.2 Middleware of Home Gateway

Some middleware technologies provide the standard way to connect the home gateway to the Internet. Therefore, some middlewares such as Open Services Gateway initiative (OSGi) provide not only the application programming interface (API), but also the deployment infrastructure and service management platform. In a Home Automation system, the home gateway may physically connect to different kinds of home network and control the appliances within them.

2.3.3 Open Service Gateway Initiative

The OSGi is a set of specifications which defines a standardized and component oriented computing environment for networked services [20]. Executing in a networked device such as an embedded device, an OSGi service platform is capable of managing the life cycle of the software components in the device. The management includes dynamically install, update, or remove the software components without disrupting the operation of the device. The software components in OSGi can dynamically discover and use other components and dynamically integrates into a further application or library [21]. The OSGi contains some standard component interfaces such as HTTP servers, logging, security, XML, and many more. Therefore, an OSGi service platform can use different implementations of the components from different vendors to obtain different optimizations.

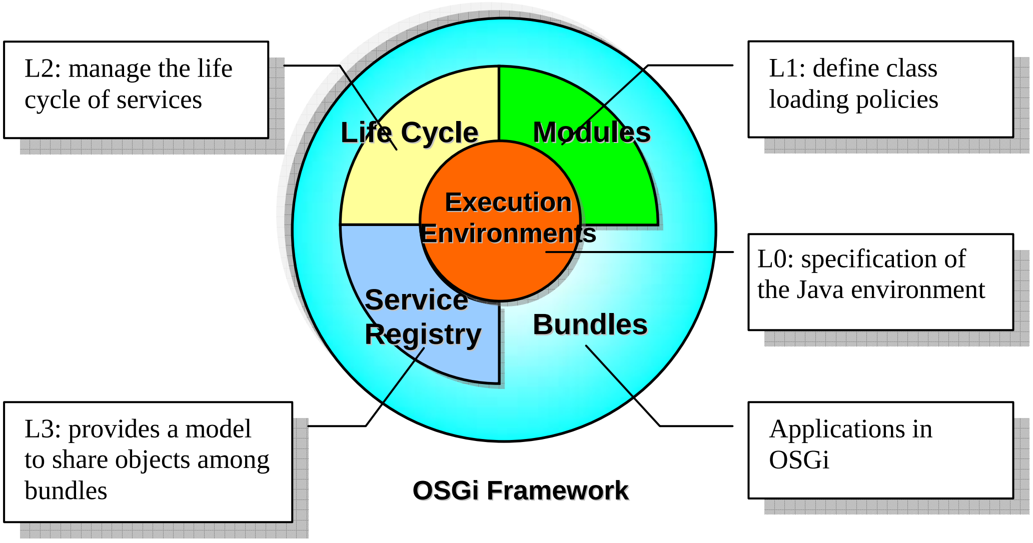

L2: manage the life cycle of services
L1: define class loading policies
Life Cycle
Modules
Execution Environments
L0: specification of the Java environment
Service Registry
Bundles
L3: provides a model to share objects among bundles
Applications in OSGi
OSGi Framework

Figure 2-4 OSGi defines the environment for gateway networked services

Traditionally, developing software component in a service gateway require complex developing and maintaining configurations. However, the OSGi software component architectures can simplify this configuration process [22]. The unit of application software for OSGi is called the bundles. The OSGi framework which is the core of the OSGi specification provides a standardized environment to the bundles. As Figure 2-4 shows, the OSGi framework is divided into four layers which are execution environment, modules, life cycle management and service registry.

Execution environment: Since the OSGi framework is based on Java, the lowest layer is the specification of valid Java environment, for example, J2SE, CDC, or CLDC. In this layer, the OSGi also define the minimum requirements on an execution environment for bundles.

Modules: Based on the top of Java environments, the OSGi framework offers modularization and defines the class loading policies in this layer. The basic Java environment only has a single shared classpath containing all the classes and resources. Furthermore, the OSGi Modules layer enhances the base environment. The layer which is a class loading module controls linking between modules and allows each module has private classes.

Life Cycle layer: The layer offers the API managing the life cycles of bundles in runtime. Therefore, the bundles can be dynamically installed, started, stopped, updated and uninstalled. The layer also provides the extensive dependency mechanisms assuring the correct operation of the environment.

Service Registry: The traditional Java class sharing is not very compatible with dynamically installing and uninstalling code. Therefore, the service registry layer offers a comprehensive cooperation model to share objects between bundles. OSGi services are represented as Java objects. The layer also defines the events to handle the services in runtime.

Because of the characteristics of OSGi, the companies especially developing embedded system will benefit from adopting the specifications in many aspects. The benefits include development cost, customization, deployment cost and unification.

2.4 Message Oriented Middleware

Message oriented middleware (MOM) technology was used for enterprise applications for many years. Functionally, MOM is a middleware transporting messages from a source component to target components. It decouples the message passing between the client and the server. Therefore, when passing message, the client and server don't have to be online simultaneously. Nowadays, the application of MOM has extended from enterprise to mobile applications [23] [24] [25]. Many different MOM products coexist on market. Nevertheless, the system can move from one to another MOM implementation without large change through the standard API for MOM such as Java Message Service (JMS). [26]

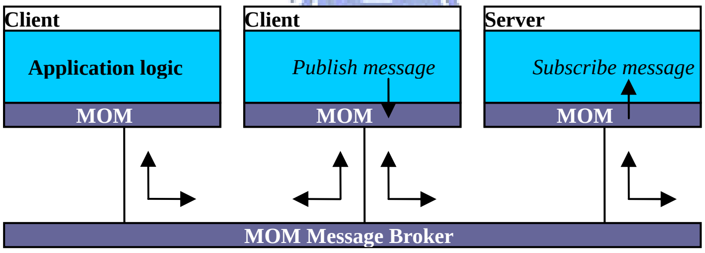


Figure 2-5 MOM decouples message passing between applications

2.4.1 Publish/Subscribe Model

There are two main types of MOM, message queuing and publish/subscribe. Message queuing MOM handles messages through point-to-point approach. The massages should pass the virtual channel which is called "queue," and each message will be received by only one receiver.

By contrast, MOM of publish/subscribe transmits messages by one-to-many way. The message sender, which is called publisher in MOM, produces and publishes messages under a certain topic, such as “upnp.bedroom.tv.” A topic is a virtual channel in publish/subscribe approach. The receivers, which are called subscriber in MOM, subscribe to a topic and then the MOM will push the messages addressed as the certain topic to subscribers. The middleware ensures that all subscribers receive all messages from all publishers on topics. For example, the MOM guarantees all subscribers of “upnp.bedroom.tv” receive all information from all publishers on the topic.

2.4.2 Benefit of MOM

Most present MOM products have the common properties of reliability and scalability. Supporting durable subscription and persistent message of MOM provides the guaranteed delivery of messages. Guaranteed delivery of messages also makes MOM suitable for machine-to-machine communication. The publish/subscribe model let MOM systems scale well when message simultaneously passing to many receivers. Besides, for an event is just considered as a special message, MOM is suitable for building event-driven system.

2.4.3 Persistent Message

The persistent message mechanism of MOM makes sure that a message published will still push to the subscriber after the MOM provider fails. When the publisher publishes a persistent message, the MOM provider will make extra effort to ensure that the message is not lost in transit. For example, the MOM provider usually logs the persistent message to persistent storage when it is sent.

2.4.4 Durable Subscription

The durable topic subscription of MOM is the mechanism to ensure the message is consumer by the subscriber which is offline or inactive when the message arrives. The durable subscriptions offer the reliability of queues to the publish/subscribe model of MOM. When subscriber creating a durable subscription is offline, the MOM provider retains the messages of the subscription until the subscriber consumes them or the subscription expires. Once the subscription is reactivated, the MOM provider delivers the messages published while the subscriber was inactive or offline.

Figure 2-6 and Figure 2-7 show the difference between a nondurable subscription and a durable subscription. In the case of durable subscription, the subscriber and the subscription are coterminous. Therefore, when subscriber is closed or offline, the subscription ends. Between the time when the subscriber is inactive and the time when the subscriber is active again, the subscriber can't consume the message published to the topic. Therefore, in the Figure 2-6, the subscriber consumes messages m1, m2, m5, and m6, but can't consume messages m3 and m4 which will be lost.

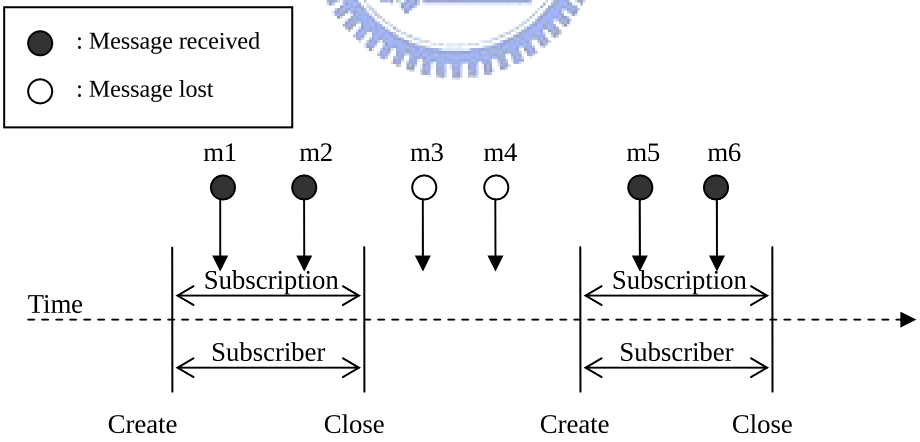


Figure 2-6 the nondurable subscribers and subscriptions

In the case of durable subscription, the subscription continues to exist and to hold messages until the application unsubscribe the topic when the subscriber is offline or close. The MOM provider stores the messages and pushes them to the subscriber when the subscriber is online again. The subscriber will still receive the messages published when the subscriber is inactive or closed. In Figure 2-7, the messages m3, m4 arrive while the subscriber is inactive are not lost and the MOM push them to the subscriber when the subscriber is active later.

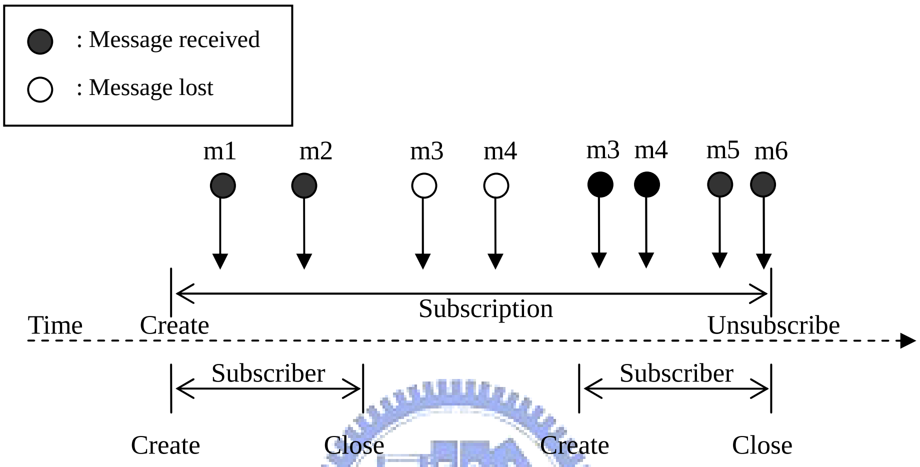


Figure 2-7 the durable subscriber and subscription

2.5 Related Work

2.5.1 Interoperability of Home Networks

The past researches about heterogeneous home network focused on interoperability among different home networking technologies. There are two kinds of approaches to developing the interoperability of heterogeneous home network The first kind is the one-to-one bridge and the second kind is the one-to-any middleware.

One-to-One Bridge: There are many one-to-one bridge middleware such as "Experimental Bridge LONWORKSR/ UPnP" [27] and "Jini Meets UPnP: An Architecture for Jini/UPnP Interoperability." [28] Besides, there has been also the bridge between Jini and the SLP-Jini [29] and the bridge between Bluetooth ESDP and UPNP [30]. The bridge develop the interoperability between two protocols, and the makes the appliances and services in the two networks can cooperate and communicate with each other. However, when there is a new home network added to the environment, the developer should develop bridges between the kind of the new home network and each kind of the existing home networks. Therefore, the (n*(n-1))/2 cost conversion complexity through one-to-one bridges is too high when the number of home networking technologies is increasing.

One-to-Any: The one-to-any approach may be more suitable for Home Automation in heterogeneous environment [31][32][33], for example, “A Framework for Connecting Home Computing Middleware” proposed by Eiji Tokunaga et al.[34] and Universal Middleware Bridge (UMB)” proposed by Kyeong-Deok Moon et al. [35]. With the former, all the converting interfaces should be regenerated via javassist tool when user adds an appliance which is supporting new protocol. Therefore, the former is not convenient for Home Automation. On the other hand, UMB is also not the platform for Home Automation. The research on Home Automation in heterogeneous environment like MHAP is very lacking.

2.5.2 Application in Heterogeneous Environment

There are some studies on application in heterogeneous home networking environments. Universal Remote Console (URC) is a study to offer control function of all device for user [36]. “A Communication Architecture for Spontaneous Systems” proposed by J. Latvakoski et al. [37] uses Session Initiation Protocol (SIP) [38] and OSGi to provide a spontaneous system among personal area networks. However, researches on Home Automation in heterogeneous home networking environment are critically lacking.

Chapter 3 System Architecture

3.1 Layered Architecture

For adapting heterogeneous Home Automation environment and application, MHAP has a four-layered architecture showed by Figure 3-1.

3.1.1 Physical Device and Network

The bottom layer consists of any home network and physical device or appliance supporting of home networking technology. MHAP aims at working over any device and home network environment, even the new one appearing in the future.

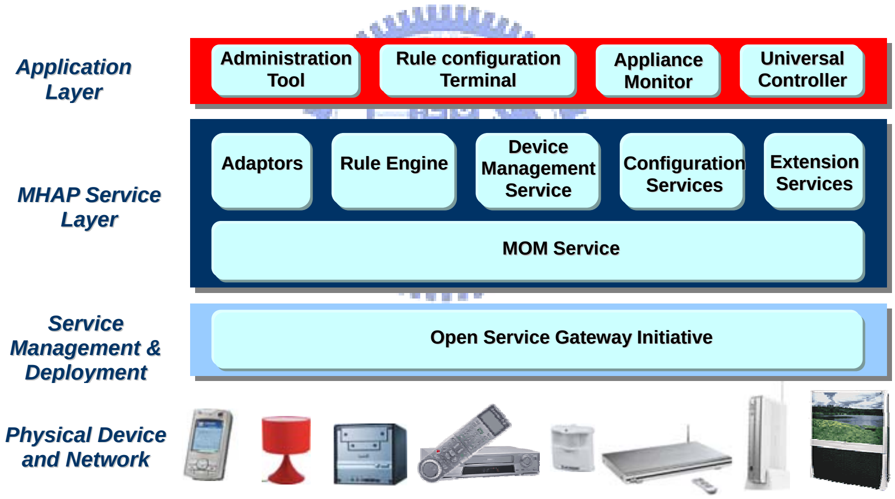


Figure 3-1 architecture of MOM-based Home Automation Platform (MHAP)

3.1.2 Service Infrastructure

The layer provides service management and deployment functions for MHAP services. With increasing on different protocols or appliance manufacturer, MHAP needs to install new service bundles and update the existing service bundles. Infrastructure layer guarantees MHAP service bundles activate and deactivate dynamically with dependency check. It must provide standardized methods to control life cycle of services including downloading and installation from service provider through Internet. In the apparent reference implementation, MHAP choose the Open Service Gateway Initiative as the layer.

3.1.3 MHAP Service

The third layer is MHAP service layer, which is the core of MHAP and consists of many services. In brief, the layer provides all functionalities in event-based Home Automation, which includes event notification, appliance control, Home Automation rule configuration and device management, etc. MHAP service layer is also an adaptation layer, which let program or services operate appliances supporting different Home Automation technologies with the same method. Internally, the layer introduces MOM technology as the method to support event-driven Home Automation in heterogeneous environment. Furthermore, the special application routing in MOM makes the design and implementation of message routing more simple and visual especially when there is more rule added.

3.1.4 Application

The top layer is application layer. Facilitating Home Automation needs many different kinds of applications, for example, the administration tools, rule configuration terminal, appliance monitor or controllers. With assistance of the first three layers in MHAP, developing all of these Home Automation applications is easier. Even through there are many different kinds of home network appliances and technologies, developers needn't consider the complex differences in appliance and network.

3.2 MHAP Messaging

In MHAP service layer, services use messages to communicate with each other.

Furthermore, the message routing in MOM is the main method to finish rule-based Home Automation. There are four main MHAP message types, which are respectively command, query, state and event.

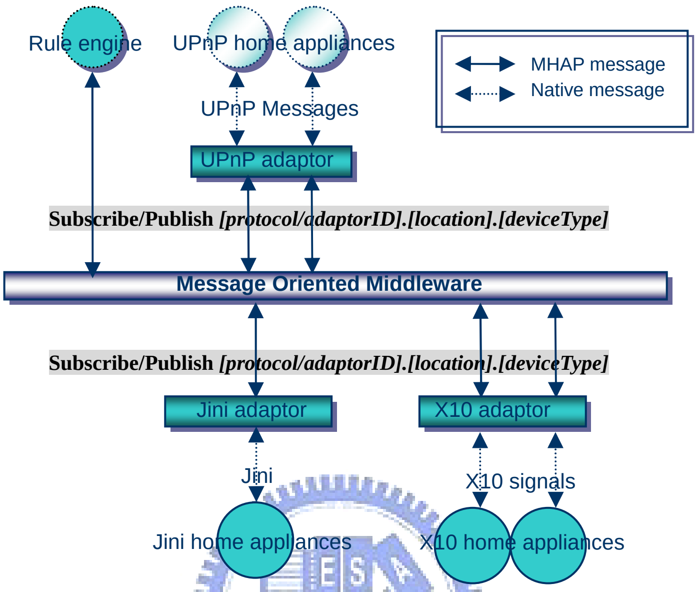


Figure 3-2 message routing in MHAP

As Figure 3-2 shows, two main services communicating with each other are the rule engine and the adaptor. MHAP rule is the basic unit representing automatic Home Automation operations assigned by the user. Rule engine is the application router in MHAP to route the Home Automation message between the adaptors through referring MHAP rules. An adaptor service is the proxy of some kinds of physical home appliances, which creates virtual appliance objects through MHAP descriptors. Different appliance manufacturers or service provider could implement different adaptor to make such appliances join into MHAP.

There are four main MHAP message types, which are respectively command, query, state and event (Table 3-1).

Message	Main Field	Meaning
Command	Device ID	*Indicate the target appliance*
	Source ID	*Indicate the source of the message; for example, the value is the service id of a rule engine when it send a command*
	Service ID	*Assign the service inquired*
	Action name	*Indicate the action demanded to invoke*
	Argument name	*Indicate the arguments needed*
	Argument value	*Carry the value of a argument*
Event	Device ID	*Indicate the appliance sending the event*
	Service ID	*Indicate the service which the event belongs to*
	State variable name	*Indicate the state variable of the service which the event reflects the change of it*
	State variable vale	*Carry the value of the state variable associated with the event*
	Date type	*Indicate the date type of the state variable*
Query	Device ID	*Indicate the target appliance or service*
	Source ID	*Indicate the source of the message*
	Service ID	*Assign the service which is demanded to query the state*
	State variable name	*Indicate the state variable which is queried*
State	Device ID	*Indicate the appliance sending the message*
	Service ID	*Indicate the service which the state belongs to*
	State variable name	*Indicate the state variable of the service*
	State variable vale	*Carry the value of the state variable*
	State variable type	*Indicate the date type of the state variable*

Table 3-1 attributes of MHAP messages

3.2.1 Command Message

The MHAP controls the appliances by command message. For example showed by Figure 3-3, the rule engine sends the “power on” command messages to power on the appliances. The adaptors of the appliances received the control message need not response.

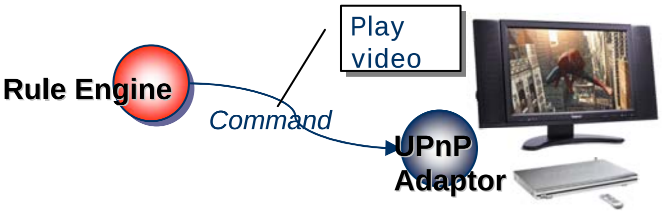


Figure 3-3 the service uses command message to control devices

3.2.2 Query Message

The function of MHAP query message is to attain the state of appliances. MHAP services can publish query message if it wants to know the state of an appliance. Then the MOM provider will push the query message to the adaptor handling the appliance.

3.2.3 State Message

Once the adaptor of the appliances receives the query message, the adaptors respond the MHAP state message containing the state information inquired. Figure 3-4 shows an example of query message and state message. The device management sends a query message to ask the power state of a light and the adaptor of the light replies a state message indicating that it is powered on.

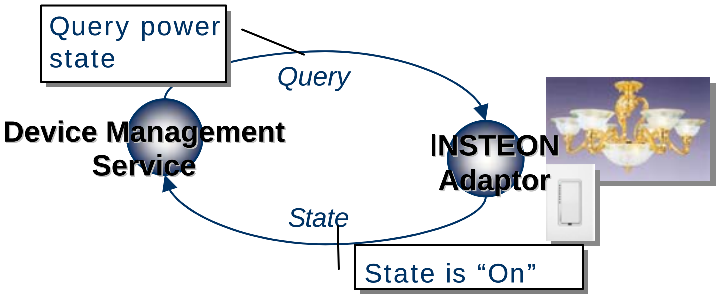


Figure 3-4 the services use query and state messages to get device states

3.2.4 *Event Message*

To notify the services, such as the rule engine, needed to be notified when some state of an appliance changes, the MHAP event message may be sent by the adaptor of the appliance while the state of the appliance changes (Figure 3-5). To send event messages, the event of the state should be subscribed by at least one service.

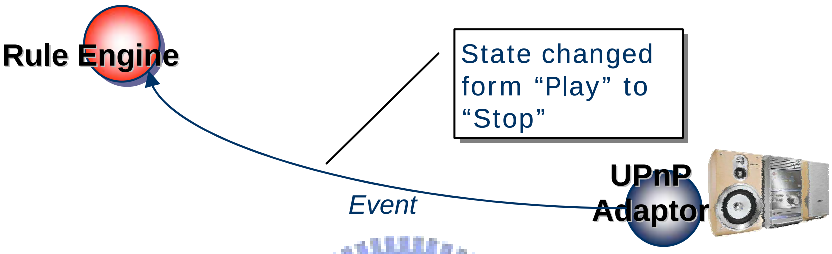


Figure 3-5 a event message reflects the change of the device state

3.2.5 *Routing and Topic Naming*

For introducing the publish/subscribe model of MOM, MHAP accomplishes high extensible Home Automation through message flow. In MHAP, every component subscribes and publishes different topics. As Figure 3-2 shows, a topic is an independent channel from a component to the MOM provider. Because each virtual appliance subscribes and publishes messages through independent channel, the joining of new appliance does not affect the existing Home Automation completely. There are some components in the figure. Rule engine is the application router in MHAP dispatching the message among the components according to MHAP rules. MHAP rule is the basic unit representing automatic Home Automation operations assigned by the user. An adaptor service is a proxy of certain kind of physical home appliances, which creates virtual appliance objects through MHAP descriptors.

The handling of MHAP message flow could be simplified through topic naming. The application router such as rule engine just let different service or virtual appliance subscribe different topic to finish message dispatching. The naming rule of topics is [protocol/adaptorID].[location].[deviceType]. When a user set a Home Automation rule, the rule engine will subscribe the event indicated in the rule. For example, A rule engine is handling a rule which will turn on the X10 light in room1 when the UPnP TV in room2 is powered off. The rule engine just subscribes the topic "upnp.room2.tv" to receive the event. When receiving the event, the rule engine will publish a command message to the topic "x10.room1.light," and then the MOM will push the message to the adaptor handling the command. Furthermore, the components in MHAP can use "*" to subscribe or publish messages of the same property at a time. For example, a component in MHAP can subscribe all x10 message from any appliance in every room through subscribing "X10.*." The same is true of other properties in topics.

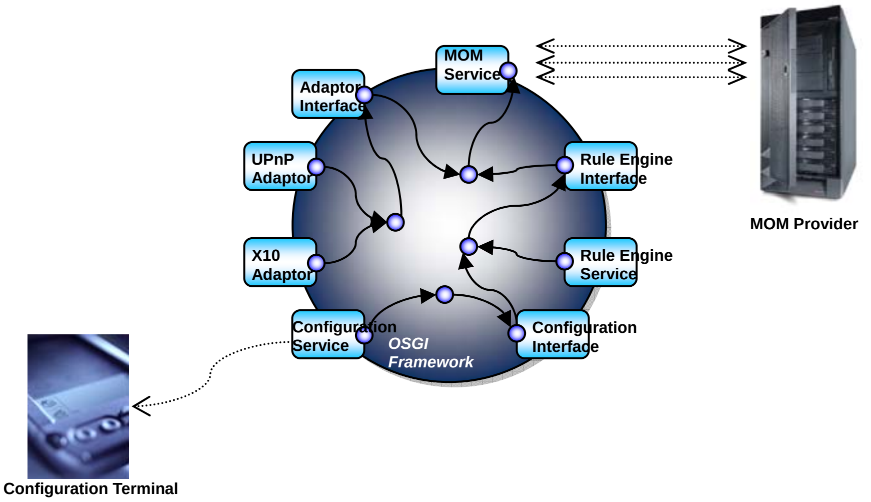


Figure 3-6 dependencies among services in MHAP

3.3 MOM Service

As Figure 3-6 shows, the MOM service in MHAP provides the JMS standard interface to publish and subscribe MOM messages. The current MHAP implementation used ActiveMQ as default JMS provider. Because JMS doesn't define load balancing, fault tolerance, security, and administration functionality, such mechanisms are provided by JMS providers and optional. Therefore, a MHAP system can choose different JMS provider software to achieve different degree of guarantee of such properties.

3.4 Rule Engine

The rule engine is the service handling MHAP message flow according to the rules set by the user with the rule configuration console. The rule engine is designed as an application router. By publishing and subscribing messages, the rule engine controls the message flow which finishing Home Automation according to the rules.

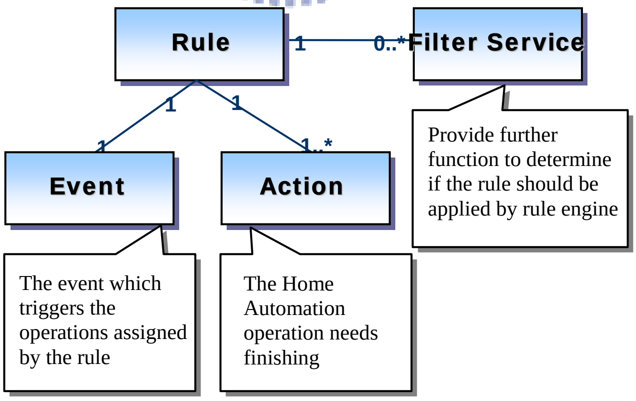

Rule
1
0..*
Filter Service
1
1
1
1..*
Event
Action
Provide further function to determine if the rule should be applied by rule engine
The event which triggers the operations assigned by the rule
The Home Automation operation needs finishing

Figure 3-7 the components of MHAP rules

Once the user sets a rule, the rule engine subscribes the event messages the rule needs. When receiving event messages, the engine refers the rules to check if any operations should be triggered by the event. Then the rule engine will publish the command messages if the operation should be executed. A MHAP rule engine is an

OSGi service. Therefore, it could be updated and activated dynamically. There is one thing to emphasize, which the MHAP consists of services and deployed on the service deployment and management infrastructure such as OSGi.

3.5 Adaptor

In MHAP, the adaptor is the proxy services of appliances (Figure 3-8). The adaptor is responsible for subscribing/publishing MHAP messages and bridging the MHAP message and native messages such as UPnP messages (Figure 3-9). The appliances belonging to a protocol or designed by a manufacturer may use the same adaptor. The adaptor needn't be re-implemented through the MHAP descriptors such as the device descriptor, service descriptor and protocol mapping descriptor.

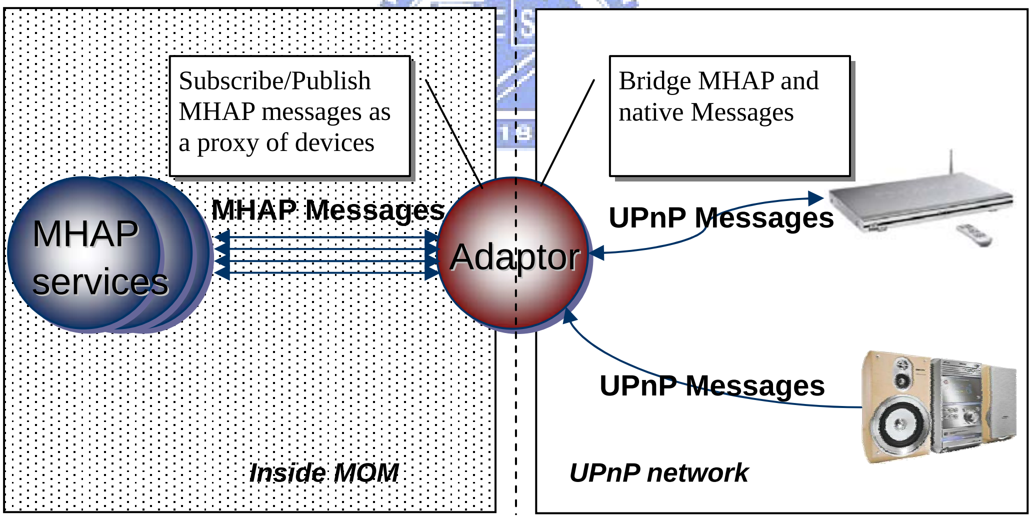

Subscribe/Publish MHAP messages as a proxy of devices
Bridge MHAP and native Messages
MHAP Messages
MHAP services
Adaptor
UPnP Messages
UPnP Messages
Inside MOM
UPnP network

Figure 3-8 an example of the UPnP adaptor

3.5.1 *Publishing/Subscribing MHAP Message*

The adaptor subscribes and publishes MHAP messages as the proxy of the appliances. For the low interoperability of present home network technologies, the appliances only understand the native messages such as UPnP messages. Therefore, MHAP used the platform independent messages, MHAP message, to communicate with each appliance and device.

The adaptor transmits or receives messages through publishing or subscribing messages. The adaptor publishes the state message when receiving the query message which requires the numbers of the appliances. The adaptor also publishes events messages when there are changes of the state occurring in appliances. Furthermore, the adaptor subscribes the query message and command messages belonging it manufacturer or home networking technology.

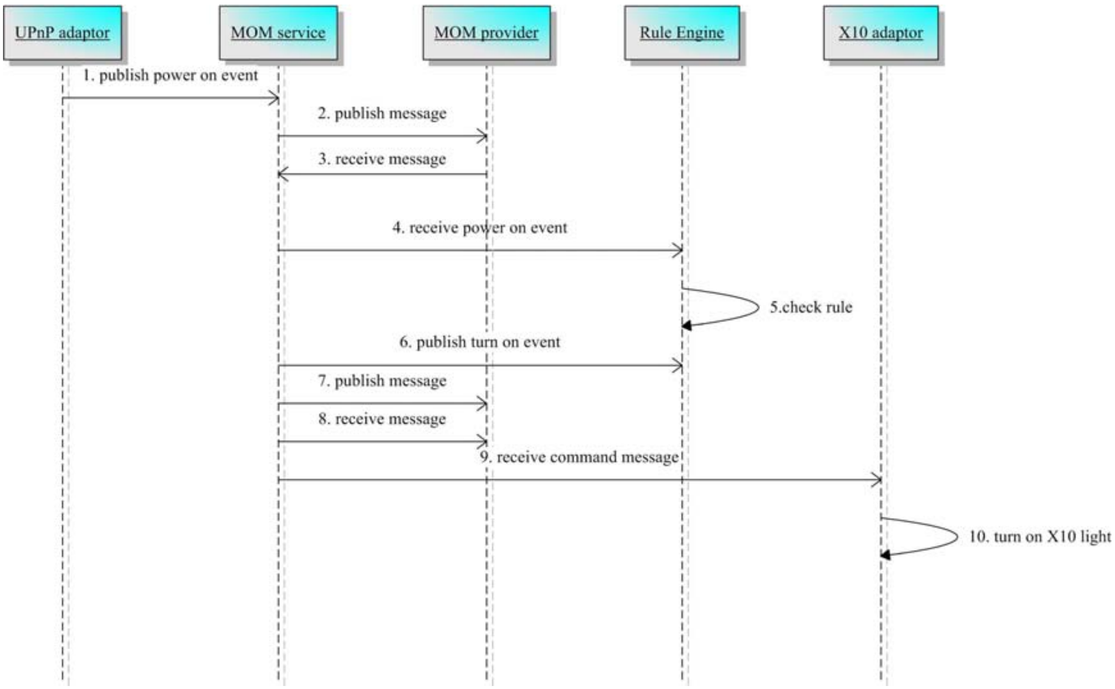


Figure 3-9 the message follow of “turn on X10 light when UPnP TV is powered on”

3.5.2 Bridging MHAP and Native Messages

According to the protocol mapping descriptor, one kind of the MHAP descriptors, the adaptor bridges the MHAP messages and the native message such as UPnP, X10 or INSTEON messages. For example, when receiving a MHAP command message to an INSTEON light, the adaptor sends the counterpart of native INSTEON command to the INSTEON light through the INSTEON network. When an adaptor receives the native UPnP event from an UPnP television, the adaptor publishes the MHAP event message if the event was subscribed.

3.6 Configuration Service

The configuration service is responsible for configuring the services of MHAP. In MHAP, there are two basic kinds of configuration, which are rule configuration and device configuration. A rule consists of event and action part. Through rule configuration service and terminal, the user can add, remove or update the rules of Home Automation. Besides, device configuration provides the interface to assign the device descriptor. The adaptor service can automatically set the URL or content of descriptors or the user can set the URL of descriptors manually with a configuration terminal.

3.7 Device Management Service

The device management maintains and provides the information such as device Id, descriptors and the adaptor of the devices. The other services, for example, the configuration service, may use device management service to maintain these data.

When the user plugs in or remove an appliance to MHAP, the adaptor handling the protocol or manufacturer of the appliance will generate an event and publish it to MOM provider. The event notifies the device management service and rule engine the id and states of the device. The device management service records and maintains all the states of the device before through the content of the event. The event may also trigger some Home Automation operation if there is a rule indicating it.

Chapter 4 MHAP Descriptors

4.1 MHAP Descriptors Overview

The MHAP descriptors provide profiles of the devices and abstract devices. Through the MHAP descriptors, the adaptor being the proxy of native devices attained the required information such as the device ID, device type, services of devices and protocol mapping.

4.1.1 High-level Abstraction

There are three kinds of descriptors in MHAP and they are the device descriptor, the service and the protocol mapping descriptor respectively.]

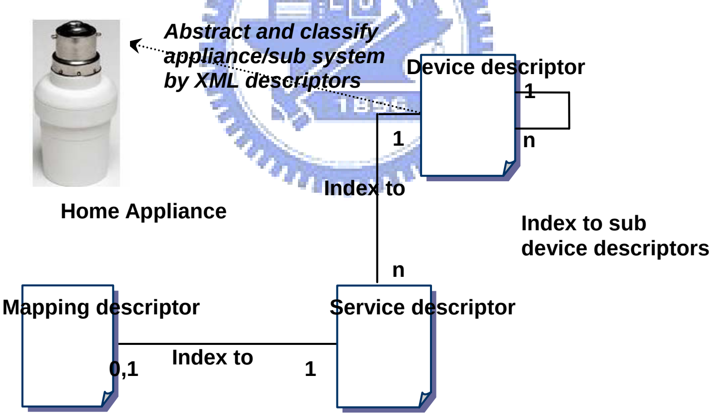


Figure 4-1 the XML-based MHAP descriptors

4.1.2 Flexibility

Descriptors make MHAP service such as adaptors more flexible. When there is new appliances added to the Home Automation system, the program of original adaptor need not to be re-implemented. In fact, the adaptors could get all the information to drive the appliance added from MHAP descriptors.

4.1.3 Based on XML

The main characteristic of the MHAP descriptors is XML-based. Therefore the descriptors are independent of any platform and extensible. The service provider developing service such as adaptors could easily parses the descriptors, attended the information needed to be processed.

4.2 Device Descriptor

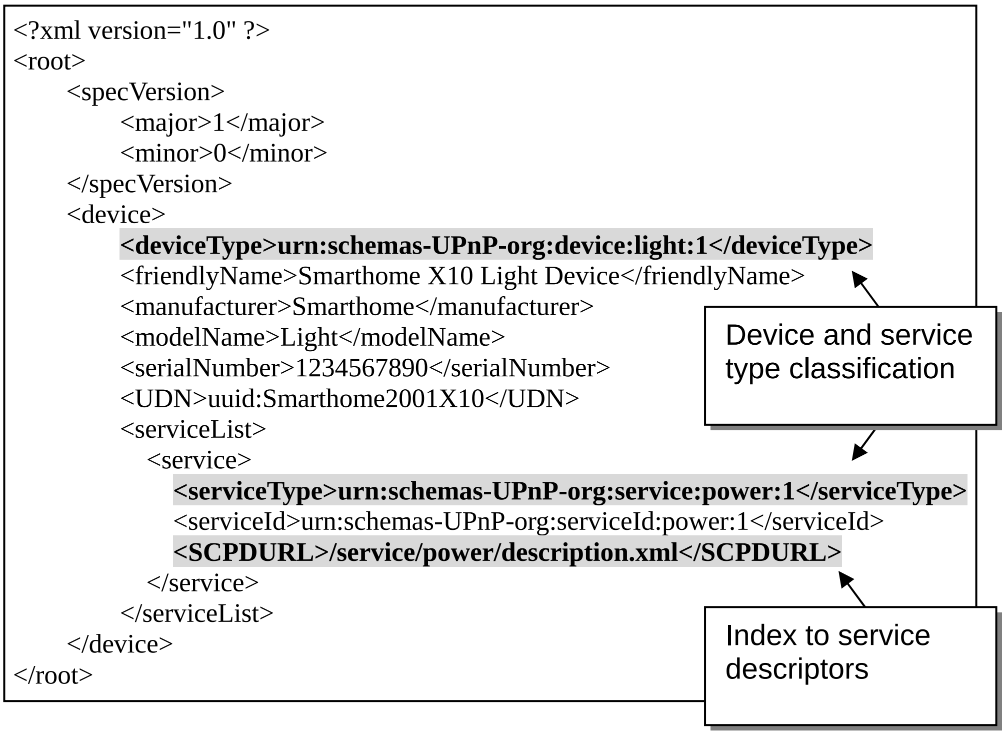
4.2.1 Root of Descriptors

Each MHAP appliance has a root device descriptor which indexes to the sub devices and services. Therefore, the appliance can be two or more types of standard device simultaneously through the hierarchical architecture. The SCPDURL tag indicates the URL of the service descriptor of the device. The service descriptor could provide the further information of invoking the services of the device. The appliance vendor can assign the UDN in MHAP. If there are two appliances in MHAP are the same, the MHAP will generate new UDN for them. The SCPDURL tag indicates the URL of the service descriptor of the device. The service descriptor could provide the further information of invoking the services of the device.

4.2.2 Device and Service Type Classification

The device descriptor abstracts ands provide MHAP devices descriptors the device and service type (Figure 4-2). The device and service types in device descriptors can be the device types defined by UPnP Forum working committee or vender specific type. The UPnP devotes to the complete device standardization, for example, Internet gateway device, lighting control, etc. MHAP takes the advantage of the standardization, although the MHAP device may not be the UPnP appliance and could be any devices support any home networking technology.

The content of the tag “dataType” can be the standard data type defined by UPnP or the vender specific type. The service type indicated by “serviceType” tag simplified the implementation of the services inside MHAP like device type. With the use of standardized types, the rule engine and adaptors can quickly subscribe and publish the required channels. If the type is not the standardized one, the MHAP components get the further channel properties from the service descriptors.



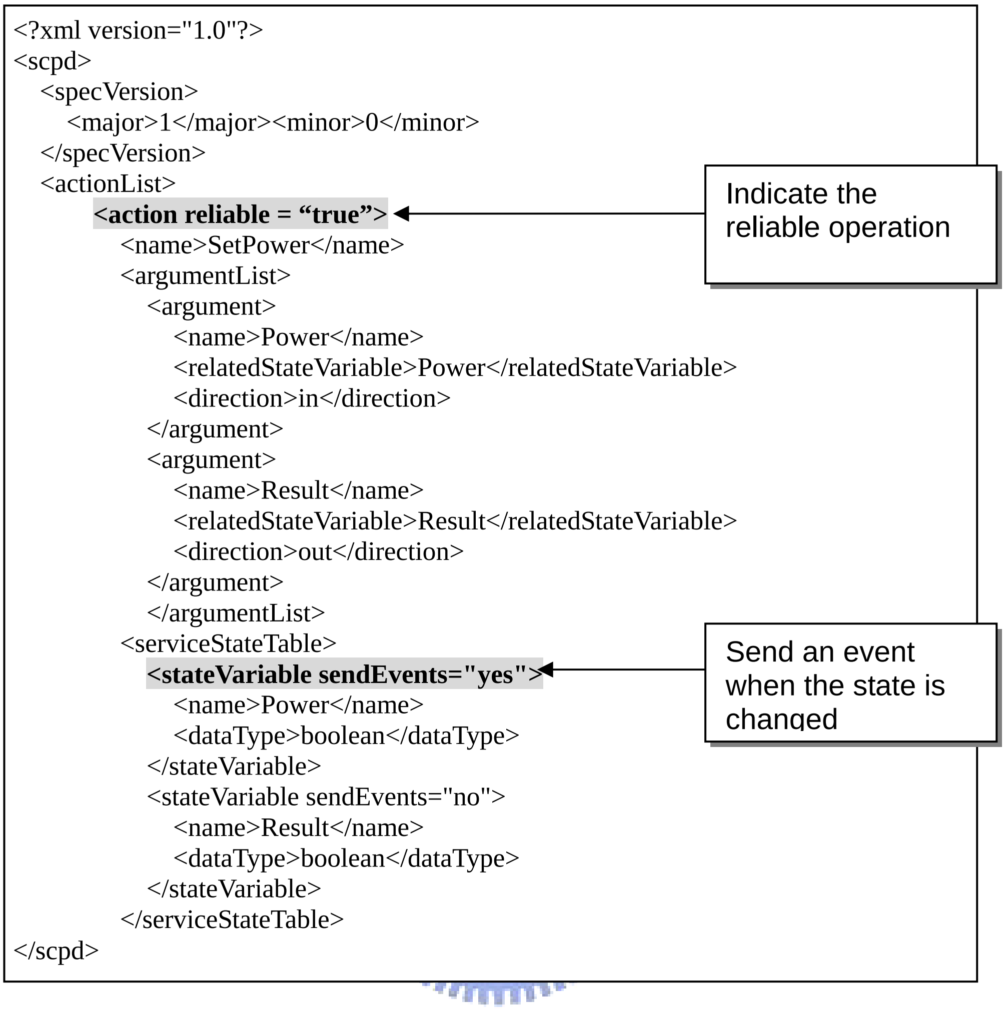
```
<?xml version="1.0" ?>
<root>
        <specVersion>
                <major>1</major>
                <minor>0</minor>
        </specVersion>
        <device>
                <deviceType>urn:schemas-UPnP-org:device:light:1</deviceType>
                <friendlyName>Smarthome X10 Light Device</friendlyName>
                <manufacturer>Smarthome</manufacturer>
                <modelName>Light</modelName>
                <serialNumber>1234567890</serialNumber>
                <UDN>uuid:Smarthome2001X10</UDN>
                <serviceList>
                    <service>
                        <serviceType>urn:schemas-UPnP-org:service:power:1</serviceType>
                        <serviceId>urn:schemas-UPnP-org:serviceId:power:1</serviceId>
                        <SCPDURL>/service/power/description.xml</SCPDURL>
                    </service>
                </serviceList>
        </device>
</root>
```

Device and service type classification

Index to service descriptors

Figure 4-2 an example of the device descriptor

The classification and standardization of device simplified the configuration of Home Automation. Most services inside MAHP such as rule configuration need not to realize what the native properties of the appliances. Processing the X10 lighting action is the same as processing UPnP light inside MHAP. The actual action of native networking is finished by adaptor services according to protocol mapping descriptors.



```
<?xml version="1.0"?>
<scpd>
   <specVersion>
      <major>1</major><minor>0</minor>
   </specVersion>
   <actionList>
         <action reliable = “true”>
            <name>SetPower</name>
            <argumentList>
               <argument>
                  <name>Power</name>
                  <relatedStateVariable>Power</relatedStateVariable>
                  <direction>in</direction>
               </argument>
               <argument>
                  <name>Result</name>
                  <relatedStateVariable>Result</relatedStateVariable>
                  <direction>out</direction>
               </argument>
               </argumentList>
            <serviceStateTable>
               <stateVariable sendEvents="yes">
                  <name>Power</name>
                  <dataType>boolean</dataType>
               </stateVariable>
               <stateVariable sendEvents="no">
                  <name>Result</name>
                  <dataType>boolean</dataType>
               </stateVariable>
            </serviceStateTable>
</scpd>
```

Indicate the reliable operation

Send an event when the state is changed

Figure 4-3 an example of the service descriptor

4.3 Service Descriptor

4.3.1 Actions, Services and Events

The service descriptor indicated the capability of control and event which an appliance supports. The service consists of action and state variable. The action is like method of Java language or operation of XML web services. An action associates with a state variable. State variables show the state of the service, and the event may be send when the state changes. The attribute “seneEvents” of “stateVariable” tag indicate whether the adaptor of MHAP sends events when the state changes.

4.3.2 *Values of Sate Variables*

The values of state variable may need constraints and the tags “allowedValueList” and “allowedValueRange” provide the capability. The tag “allowValueList” restricts the value of the state value to one in a list. The tag “allowedValueRange” and its sub tags can indicate the maximum and minimum value, and the size of an increment operation of a state variable.

```
<allowedValueList>
        <allowedValue>enumerated value 1</allowedValue>
        <allowedValue>enumerated value 2</allowedValue>
          ……
</allowedValueList>
```

Figure 4-4 allowed value list in service descriptor

```
<allowedValueRange>
        <minimum>minimum value</minimum>
        <maximum>maximum value</maximum>
        <step>increment value</step>
 </allowedValueRange>
```

Figure 4-5 allowed values range in service descriptor

4.4 Protocol mapping descriptor

The protocol mapping descriptor provides the information used for mapping MHAP messages to native messages such as UPnP messages and vice versa.

4.4.1 *Flexible Mapping*

Protocol mapping descriptor provides MHAP the capability of flexible mapping. Through protocol mapping descriptor, the adaptor need not be re-implemented when an added appliance is designed. The vendor needn’t provide new adaptor every time when they develop new devices. They can just provide the mapping descriptor with the service and device descriptors to add the new operation of the devices.

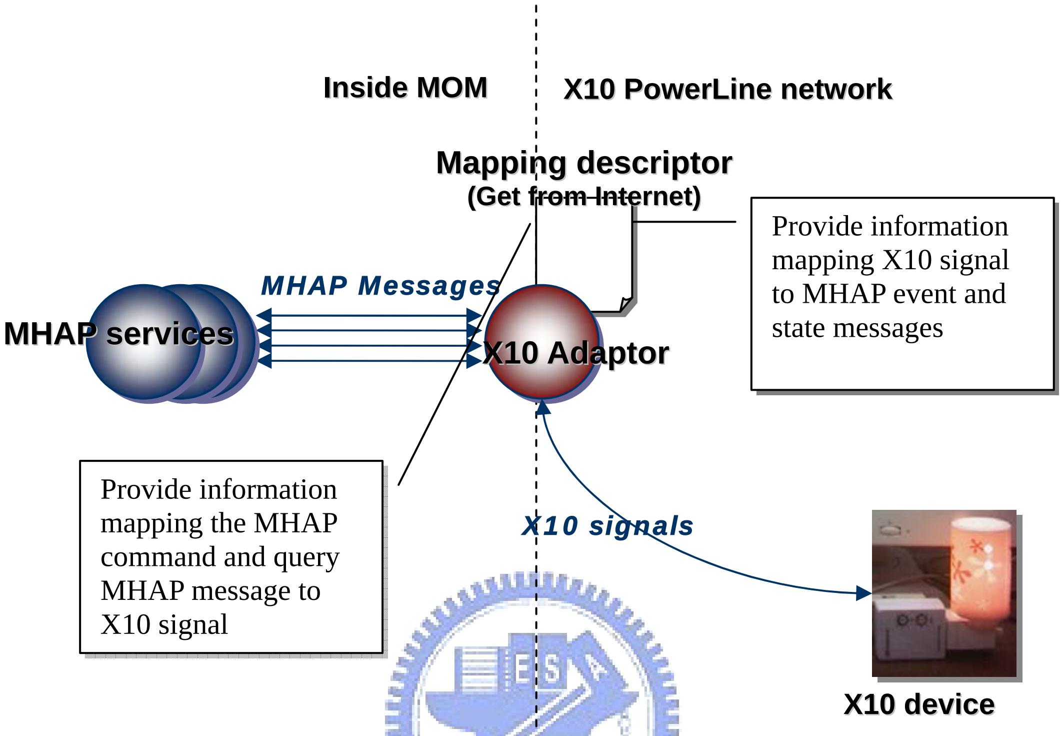
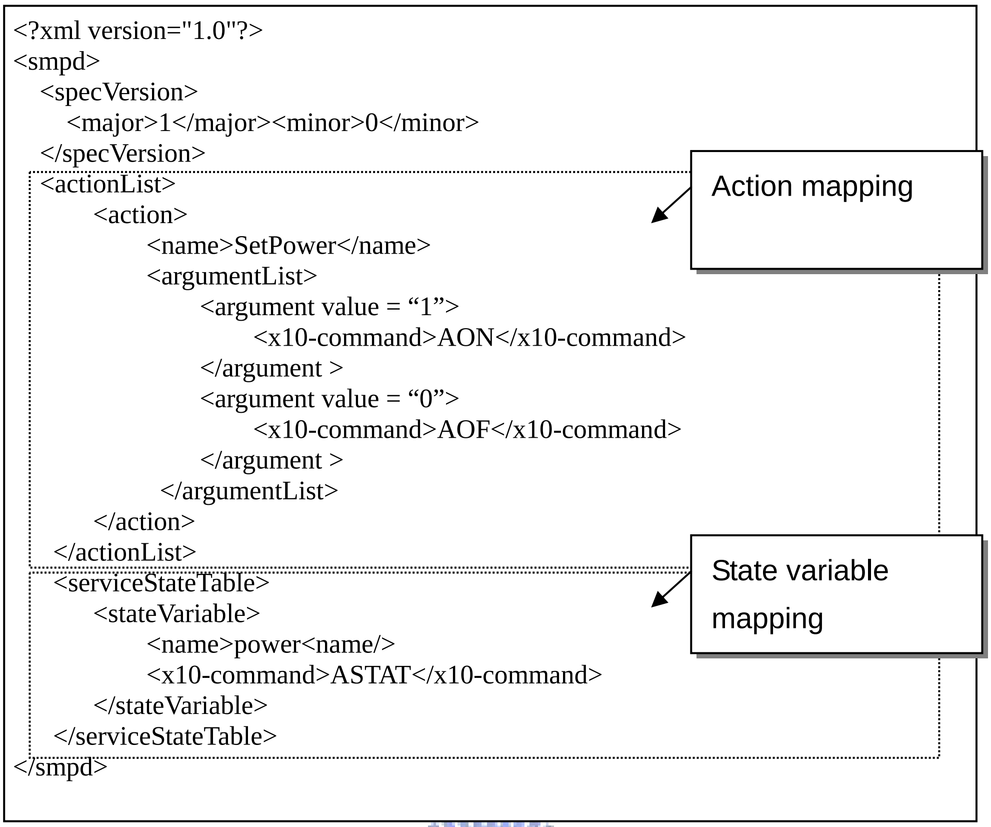

Inside MOM
X10 PowerLine network
Mapping descriptor
(Get from Internet)
Provide information mapping X10 signal to MHAP event and state messages
MHAP Messages
MHAP services
X10 Adaptor
Provide information mapping the MHAP command and query MHAP message to X10 signal
X10 signals
X10 device

Figure 4-6 the mapping descriptor offers two-way protocol mapping information

4.4.2 Association with Service Descriptors

A protocol mapping descriptor is associates to a service descriptor. Figure 4-7 shows the main portion of a protocol mapping descriptor whose native protocol is X10. When invoking the “PowerOn” action, the adaptor sends the “AON” X10 command indicated by x10-command tag. When retrieving the value of the state variable “power”, the adaptor may send the” ATST” X10 command and receive the response from X10 network.



```
<?xml version="1.0"?>
<smpd>
   <specVersion>
      <major>1</major><minor>0</minor>
   </specVersion>
   <actionList>
         <action>
               <name>SetPower</name>
               <argumentList>
                     <argument value = “1”>
                           <x10-command>AON</x10-command>
                     </argument >
                     <argument value = “0”>
                           <x10-command>AOF</x10-command>
                     </argument >
                </argumentList>
         </action>
    </actionList>
    <serviceStateTable>
         <stateVariable>
               <name>power<name/>
               <x10-command>ASTAT</x10-command>
         </stateVariable>
    </serviceStateTable>
</smpd>
```

Figure 4-7 a piece of the mapping descriptor for a X10 device

Chapter 5 Implementation

5.1 Target Environment

5.1.1 Hardware Environment for MHAP

The target execution environment of MHAP is a home gateway at home. The hardware of the MHAP gateway can be any device able to execute the infrastructure layer framework, which is the OSGi middleware for present MHAP implementation. Therefore, the MHAP gateway may be a personal computer, a game console or a wireless gateway. The hardware of the present MHAP implementation is a personal computer.

5.1.2 Software Environment for MHAP

The basic software environment for MHAP includes the infrastructure layer framework and MOM provider software, which are OSGi and JMS compliant middleware respectively.

OSGi Framework: The present OSGi framework for the present MHAP implementation is Oscar[39], which is an open source implementation of the OSGi framework specification [40]. The OSGi framework in MHAP can be changed to any OSGi compliant products or open source solution, for example, ProSyst's mBedded Server [41], IBM’s SMF [42], or Knopflerfish [43].

MOM Provider Software: The MOM provider software used for the present MHAP implementation is Active MQ [44], which offers JMS interface, the persistent message and durable subscription of messages. The MOM provider software can be also change to any JMS compliant MOM products or open source solution. The examples of the JMS compliant MOM produces include IBM’s MQSeries [45], Softwired's iBus [46], SwiftMQ Router[47], Tibco's EMS[48] and Fiorano's FioranoMQ[49]. The will-known open source JMS implementation includes JBoss Messaging [50] and OpenJMS [51].

5.1.3 Environment for End-User Application

Since MHAP integrates all the home networks, any ubiquitous programmable device can be the environment for end-user application. The target devices of the present implemented end-user applications of MHAP include the MHAP gateway, the J2ME mobile phone and the PDA phone with Window Mobile environment.

5.2 MHAP Implementation

MHAP consists of MHAP services. The infrastructure layer of MHAP uses the OSGi framework to offer the complete management and deployment for MHAP services. Two important ones of the functions are lift cycle control and dependency check. As the cooperating MHAP service resides in different house and network when the system scales to Internet, the communication model of MHAP services are publish/subscribe messaging of MOM.

5.2.1 Life Cycle Control of MHAP Services

Implementing the life cycle control interface of OSGi [52] makes the MHAP service are manageable and deployable to the MHAP infrastructure layer. Each deployable software to the OSGi framework is called a bundle. As Figure 5-1 shows, there are six states in the life cycle of a bundle.

Table 5-1 describes the six states of a bundle. After MHAP installs a service bundle from Internet or local storage device, the bundle is in installed state. The OSGi framework then resolves the dependency of the bundle. After the bundle passes the dependency resolving, it OSGi framework will move it to the resolved state. If the bundle doesn't pass the dependency resolving, the OSGi framework will automatically download and install required bundles according to the dependency of the bundle. Once the OSGi framework has successfully resolved the dependencies of the bundle, the MHAP can start it. When starting the bundle, it is in the starting state. A MHAP bundle in the state means that the start method of the activator of the bundle is active. If the start method of the activator completes, the bundle has been successfully started and then the OSGi framework change the bundle to the active state. The bundle in active state is executing it task and cooperating with other active bundles.

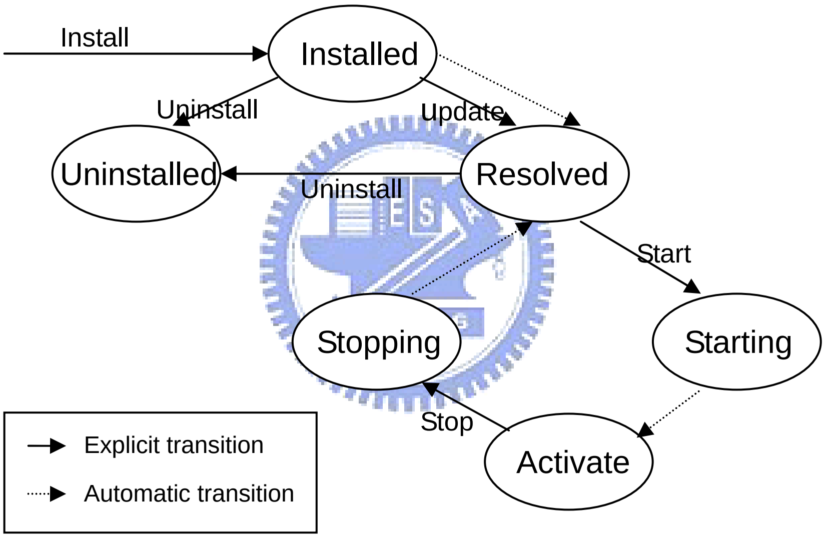


Figure 5-1 the bundle cycle of MHAP services

When MAHP stop a bundle, the bundle is in the stopping state. In the state, the stop method of the bundle activator is active. After the stop method completes, the bundle is stopped and will move back to the resolved state. The MHAP may uninstall a bundle. When a bundle is uninstalled, it is in an unusable state which is called uninstalled state. If bundle is in the uninstalled state, all references to the object of the bundle should be released immediately in order to avoid the unexpected exception about invoke unusable service of the bundle.

Table 5-1 the bundle states of MHAP service bundle

BUNDLE STATE	DETAIL DESCRIPTION
Installed state	A MHAP bundle is in this state when it has been installed but cannot be executed by OSGi framework yet. The bundle in this state of which dependencies are not resolved. The OSGi framework will attempt to resolve the dependencies of the bundle and move it to the next resolved state.
Resolved state	A MHAP bundle is in this state when the OSGi framework has successfully resolved the dependencies of the bundle.
Starting state	A MHAP bundle is in this state when the start method of the activator of the bundle is active. The bundle starts successfully if the start method completes without unexpected exception. Once the bundle starts successfully, it will move to the next active state.
Active state	A MHAP bundle is active in MHAP. That a bundle is in this state means it has been successfully started.
Stopping state	A MHAP bundle is in the state when the MHAP stopping it. The stop method of the activator of the bundle is active. After the stop method completes, the bundle is stopped and will move back to the resolved state.
Uninstalled state	A MHAP bundle is in this state when it is uninstalled. The bundle is unusable state and all references to the object of the bundle should be released immediately.

An activator is responsible for starting and stopping a bundle and executing required jobs during state transition such as releasing resource. To build an activator of a bundle, the Java class of the activator needs to implement the *org.osgi.framework.BundleActivator* interface, which declares the start and stop method of the bundle. Figure 5-2 shows the code example of an activator. An activator registers and un-registers services so that another service can find and use the service through the service registry of OSGi framework.

```
public class MyBundleActivator implements BundleActivator{

   private BundleContext bundleContext;

   public void start (BundleContext bc){
         setBundleContext (bc);
         // ...
         BundleListener bl = new RuleEngineBundleListener();
         bc.addBundleListener(bl);
   }

   public void stop (BundleContext bc){
         // ...
         setBundleContext (null);
   }
   // Other methods...
}
```

Figure 5-2 code example implementing BundleActivator

The activator often registers the listener of “BundleEvent.” The listener of “BundleEvent” implements required jobs during state transition. The “BundleEvent” contains bundle state-change information and the OSGi framework will fire it to the registered listener which implements *org.osgi.framework.BundleListener* interface.

Table 5-2 listener interfaces about bundle, service and framework events

Listener	Event	Type
BundleListener	BundleEvent	INSTALLED, RESOLVED, STARTED, STARTING, STOPPED, STOPPING, UNINSTALLED, UNRESOLVED, UPDATED
ServiceListener	ServiceEvents	MODIFIED, REGISTERED, UNREGISTERING
FrameworkListener	FrameworkEvent	ERROR, INFO, PACKAGES_REFRESHED, STARTED, STARTLEVEL_CHANGED,WARNING

Besides the “BundleEvent,” the developer of MHAP service could implement and register the other kinds of handler defined in OSGi to monitor and handle the service and framework event. Table 5-2 shows the listener and event developer can use when implementing the MHAP services.

```
public class RuleEngineBundleListener implements BundleListener{
        void bundleChanged(BundleEvent event){
                switch(event.getType()){
                        case BundleEvent.INSTALLED :
                                //The bundle has been installed.
                                break;
                        case BundleEvent.RESOLVED:
                                //The bundle has been resolved.
                                break;
                        case BundleEvent.STARTED:
                                //The bundle has been started.
                                break;
                        case BundleEvent.STARTING
                                //The bundle is about to start.
                                break;
                        case BundleEvent.STOPPED:
                                //The bundle has been stopped.
                                break;
                        case BundleEvent.STOPPING:
                                //The bundle is about to stop.
                                break;
                        case BundleEvent.UNINSTALLED:
                                //The bundle has been uninstalled.
                                break;
                        case BundleEvent.UNRESOLVED:
                                //The bundle has been unresolved.
                                break;
                        case BundleEvent.UPDATED:
                                //The bundle has been updated.
                                break;
                }
        }
 }
```

Figure 5-4 code example implementing BundleListener

Figure 5-3 shows an example of the *BundleListener* implementation. The *bundleChanged* method of the listener will receive the object of *BundleEvent* and handle the event. The type of the *BundleEvent* includes “INSTALLED,” “RESOLVED,” “STARTED,” “STARTING,”, “STOPPED,” “STOPPING,” “UNINSTALLED,” “UNRESOLVED” and “UPDATED.” Different types of BundlEvent reflects different situation in state transition of a bundle. The listener implements different handling codes depending on each types of BundleEvent bject received.

5.2.2 *Indicating Dependency between Services*

The service dependency is indicated in a manifest file. After implementing the classes for the MHAP bundle, the developer must define a manifest file that contains metadata for the OSGi framework. The OSGi framework manipulates the bundle according to the metadata. Figure 5-5 shows the manifest file of the MHAP rule engine. The Import-Service portion indicates the service that the rule engine bundle will use. The Export-Service portion indicates the service of the rule engine bundle that other bundle will use. Beside the service level dependency, the API level dependency must be indicated in “Import-Package” and “Export-Package” portion. The OSGi framework resolves the dependencies of a bundle by the four portions of the manifest file.

```
Manifest-Version: 1.0
Bundle-Name: Home Automation Rule Engine
Bundle-Classpath: .,mahprule.jar
Created-By: 1.4.2_11 (Sun Microsystems Inc.)
Bundle-NativeCode: /wincomlib.dll ; processor = x86 ; osname = Windows
   NT, /wincomlib.dll ; processor = x86 ; osname = Windows 98 ,/wincoml
 ib.dll ; processor = x86 ; osname = Windows 95, /wincomlib.dll ; proc
 essor = x86 ; osname = Windows 2000; processor = x86 ; osname = Windo
 ws XP
Bundle-Activator: mhap.RuleEngineActivator
Import-Package: org.osgi.framework, mhap.mom
Export-Package: mhap.rule
Import-Service: mhap.mom.MOMService
Export-Service: mhap.rule.RuleEngine
Bundle-Version: 1.0.0
```

Figure 5-5 the manifest file of the MHAP rule engine bundle

A final MAHP bundle refers to a JAR [53] file. The JAR file packages the manifest file and the Java class and library file associated with the bundle.

5.2.3 *Publishing/Subscribing Topic*

The MHAP services communicate with messages through MOM core. The MOM service bundle exports MOM service with the JMS package. The other MHAP service can send and receive messages through the MHAP MOM service. The MHAP use the publish/subscribe model of MOM. The MHAP MOM service uses the standard JMS package javax.jms to manipulate the messages. When publishing messages on a topic, the MHAP MOM service uses a *javax.jms.TopicPublisher* object, which is the publish/subscribe form of a message producer. The publish method of the *javax.jms.TopicPublisher* object is responsible for publishing a message to the topic. To receive messages that have been published to a topic, the MHAP MOM service uses a *javax.jms.TopicSubscriber* object which is the publish/subscribe form of a message consumer. Since the implementation of MOM service is based on JMS, the MHAP can work with any JMS compliant products or open source JMS solution.

5.2.4 Reliable Message

The MHAP supports the reliable operation, which will be finished even if the MOM provider fails and restarts or the operated appliance is offline temporarily at the time when the operation is occur. MHAP introduces both the persistent message and durable subscription mechanism to provide the reliable operation with achieving reliable message delivery.

Persistent Message: The event message that triggers the reliable operation and the command message that indicates the reliable operation are published in the persistent mode of JMS. In this case, the MHAP pass the parameter of deliveryMode is *DeliveryMode.PERSISTENT* when invoking the publish method of the TopicPublisher object. The JMS API supports two message delivery modes which indicates whether messages are lost if the JMS provider fails. The delivery modes are *DeliveryMode.NON_PERSISTENT* and *DeliveryMode.PERSISTENT* fields of the *DeliveryMode* interface. In persistent delivery mode, the JMS provider which is

ActiveMQ in present MHAP environment take extra care to ensure that a message is not lost in transit in case the JMS provider fails. A MHAP message sent with this delivery mode is kept in persistent storage when it is sent. If the operation is not marked reliable, MHAP sends the message related to the operation with the non-persistent mode. In the case, the JMS provider doesn’t store the message or otherwise guarantee that it is not lost, which improves the performance of MHAP and reduce storage overhead of the MOM provider.

Durable Subscriptions: Besides persistent message, MHAP uses durable subscriptions for the subscribers to ensure that the appliance executing reliable operation receives all published messages. The durable subscription of MOM provides the reliability of queues to the publish/subscribe message domain. Therefore, the MOM provider will still push the message to the appliance even if it is offline when the message is published. The MHAP uses the *createDurableSubscriber* method of *javax.jms.Session* to create a durable subscriber object. The JMS provider then stores the messages sent or published to the topic, as it would store messages sent to a queue. The JMS provider delivers the messages that were published while the subscriber was inactive. When handling the messages which are not related to any reliable operation, the MHAP use the *Session.createConsumer* method to create a nondurable subscriber. Although a nondurable subscriber can receive only messages that are published while it is active, nondurable subscription improves the performance and reduces storage overhead.

5.3 End-user Application Implementation

As Figure 5-6 shows, this study implements two end-user applications for demonstration. The first one is the universal remote controller and the other is the Home Automation rule configuration tool. The universal remote controller is installed and executes in a mobile phone supporting J2ME and the remote rule configuration tool is installed and executes in a PocketPC.

5.3.1 Universal Remote Controller

The universal remote controller is a Java program and can execute in any J2ME/MIDP environments. The universal remote controller sends messages to the universal controller service in MHAP gateway through TCP/IP. The universal controller service in MHAP gateway in the application layer of MHAP controls devices through device management service.

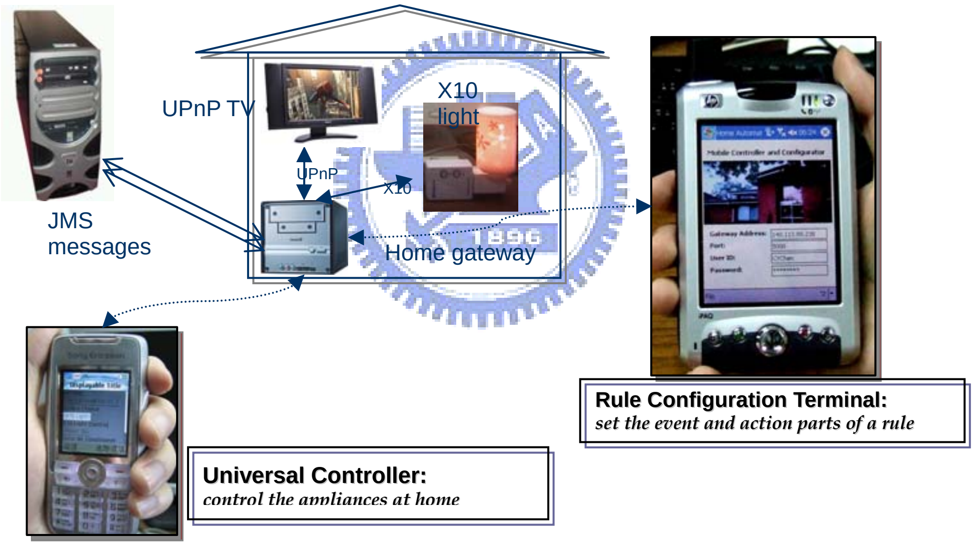

UPnP TV
X10 light
UPnP
X10
JMS messages
Home gateway
Rule Configuration Terminal:
set the event and action parts of a rule
Universal Controller:
control the appliances at home

Figure 5-6 various Home Automation applications in MHAP

5.3.2 Remote Rule Configuration tool

The remote rule configuration tool consists of two parts. The first part is the terminal executing on mobile devices. The second part is the portal service in MHAP gateway. The configuration terminal is a .NET application which can executes on any mobile device providing Windows Mobile environments. The configuration terminal implements the user friendly GUI adding, editing and removing rules, therefore the user can configure Home Automation at home easily through handheld devices. The configuration terminal sends and receives configuration messages to and form the configuration portal. The configuration portal uses the configuration service to manage and set the Home Automation rules in MHAP. The present implementation of rule configuration tool also integrates the universal control and appliance monitor function offered by device management service.

Chapter 6 Application Demonstration

6.1 Demonstration Environment

6.1.1 MHAP Gateway

MHAP could be deployed on any home gateway device with any OSGi compliant software environment. As Figure 6-1 shows, the hardware of the MHAP gateway in the demonstration is an IBM X 31 notebook. The demonstration gateway connects to X10 and INSTEON networks through X10 and INSTEON computer interfaces respectively. The X10 power line computer interface sends and receives native X10 messages in X10 network whose medium is the power line. Similarly, the INSTEON computer interface sends and receives native INSTEON messages in INSTEON network whose medium is also the power line. The demonstration gateway also connected to LAN at home through IEEE 802.11b wireless LAN.

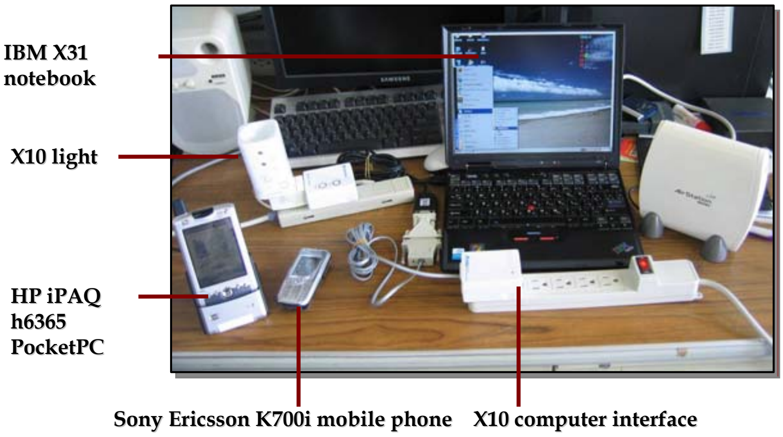


Figure 6-1 the basic demonstration hardware

The notebook acts as both the home gateway and MOM provider. The OSGi framework of the demonstration gateway is Oscar and the software of MOM provider is ActiveMQ.

Table 6-1 appliances in the demonstration environment

Appliance	Native Protocol	Device Type	Function
D-link DSM 320 media player	UPnP	AV renderer	*1. Browse, choose and play audio/video files from UPnP media gateway* *2. Further control such as volume adjustment function*
D-Link media server on PC	UPnP	Media server	*1. Provide audio/video files for AV renderers* *2. Manage audio/video files and connections to AV renderers*
Dimmable light with INSTEON LAMPLINC adaptor	INSTEON	Lightening control	*1. Dimmable lighting* *2. Reply to status request*
Wireless X10 occupancy sensor with X10 RF Transceiver	X10	Sensor	*Detect motion*
Electronic fan with X10 two-way appliance adaptor	X10	HVAC	*1. be controlled to power on/off* *2. Reply to status request*
Light with X10 two-way appliance adaptor	X10	Lightening control	*1. be controlled to power on/off* *2. Reply to status request*

6.1.2 Home Appliances and Sub Systems

Table 6-1 describes the appliances in the demonstration of the paper. The smart appliances supporting different kinds of home networking technologies connects to

MHAP and works together to accomplish Home Automation. The home gateway connects to the UPnP appliances such as the DSM 320 media player and the media server through IEEE 802.11b wireless LAN. The DSM 320 media player is an UPnP entertainment appliance at home which can browse and play the multimedia files form UPnP media server, which is another UPnP appliance. The X10 smart appliances uses power line signals for communication and INSTEON appliances do, too. Figure 6-2 shows the physical appliances in the demonstration.

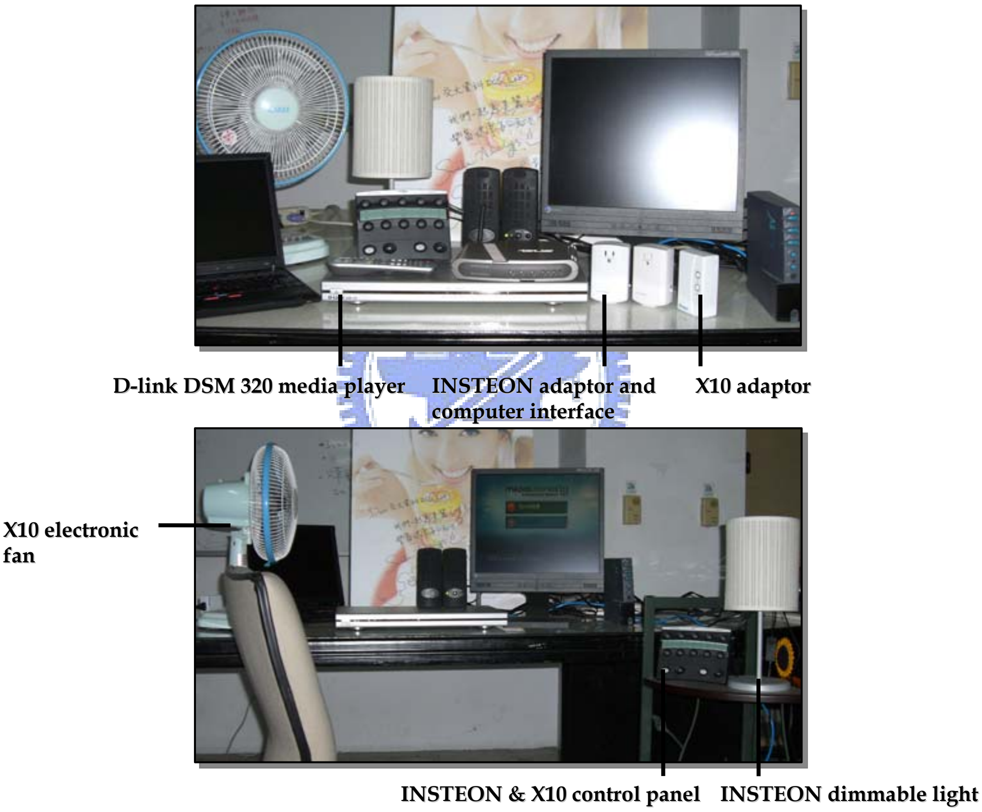

D-link DSM 320 media player
INSTEON adaptor and computer interface
X10 adaptor
X10 electronic fan
INSTEON & X10 control panel
INSTEON dimmable light

Figure 6-2 appliances and sub systems of demonstration environment

Beside the physical appliances, the simulated devices of the project CyberLink for Java [57] are also the demonstration appliances. CyberLink for Java is a development package for UPnP application. As Figure 6-3 shows, the simulated devices include televisions, clocks, air conditioners, binary lights, remote controllers and washers. The simulated devices all support UPnP control protocols and offer various UPnP services such as services controlling power states and services adjusting temperature. They also offer the UPnP event function to notify the change of the device states.

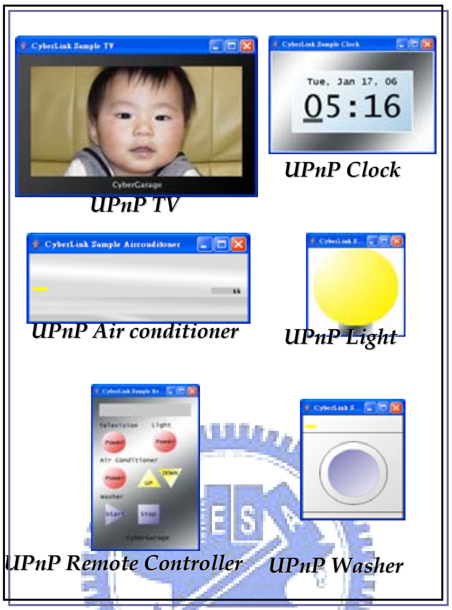


Figure 6-3 simulated devices of project CyberLink for Java

6.1.3 *Devices for Configuration and Control*

Some appliances or device are used for control or configuration purpose. In the demonstration, a Sony Ericsson K700i mobile phone supporting J2ME can control the appliances in MHAP as a universal remote controller. The mobile phone connects to the MHAP gateway through GPRS. Besides, another HP iPAQ h6365 PocketPC is running the application which is both the universal remote controller and the Home Automation rule configuration terminal. The PokcetPC connects to the MHAP gateway through IEEE 802.11b wireless LAN.

6.2 Deployment Architecture

Figure 6-4 indicates the deployment architecture of the MHAP. Inside home, the smart appliances connect to the home gateway through several home networks such as X10, INSTEON and UPnP. The MHAP with MHAP services deployed such as rule engine and adaptors is executes in the home gateway. The MOM provider software can executed in the same home gateway or in another machine. The MHAP gateways can connects to each other though Internet and publish/subscribe messages among each other.

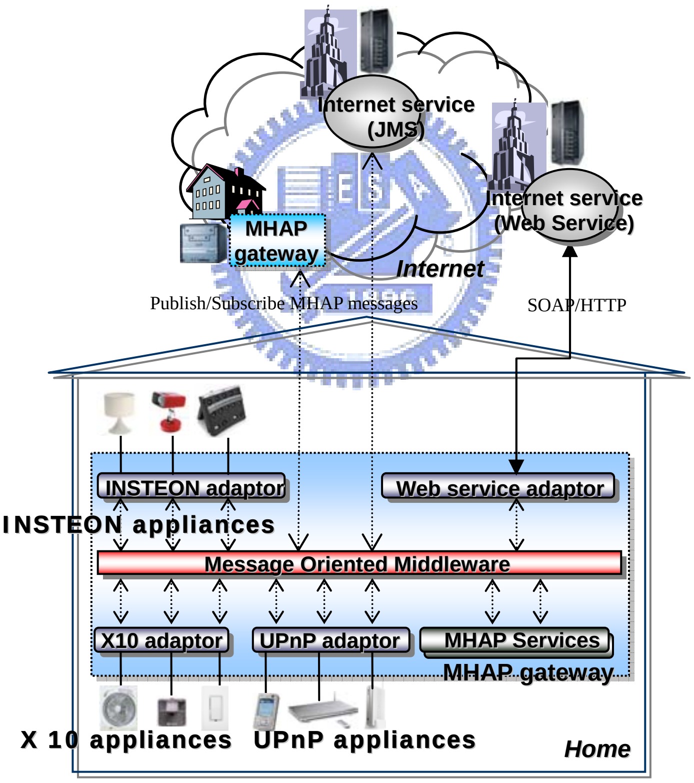


Figure 6-4 the deployment architecture of MHAP

Outside home, the Internet services such as JMS services and Web services are deployed on the server of the service provider. The JMS Internet services can directly subscribe and publish MHAP messages. The other Internet services such as Web services [54] subscribe and publish MHAP messages through service adaptors which are similar to appliance adaptors. Therefore, the legacy services outside home needn't change or be deploy to another place. Through the deployment architecture, MHAP can push message outside home to the service of a service provider such as a home security company or subscribe the messages form the service.

6.3 Universal Control

The user can also control the appliances at home with the MHAP universal controller. Figure 6-5 shows the J2ME version of the universal controller. The J2ME version of universal controller connects to MHAP home gateway through TCP/IP. It display the appliances at home and the user select an appliance to control.

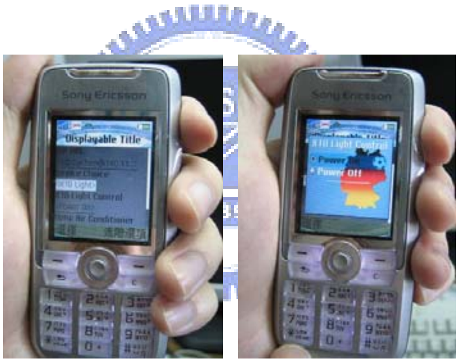


Figure 6-5 turn on a X10 light with a universal control application

While the user powered on the X10 light through the universal controller, the controlling MHAP services in gateway uses the device management service to finish the task. As Figure 6-6 shows, the device management service published a "power on" command message to topic "x10.room1.light." The adaptor of the X10 light received the control message for X10 light and transmitted the native X10 command through the power line interface. Finally, the X10 light was turned on after receiving the command. While the user powered off the UPnP TV through the universal controller, the device management service published a "power off" command message to topic "upnp.room1.tv." The adaptor of the UPnP TV then received the MHAP command message and transmits the native UPnP power off message. The UPnP TV was powered off after receiving the UPnP message. Therefore, the MHAP can control all the appliances at home through message flow.

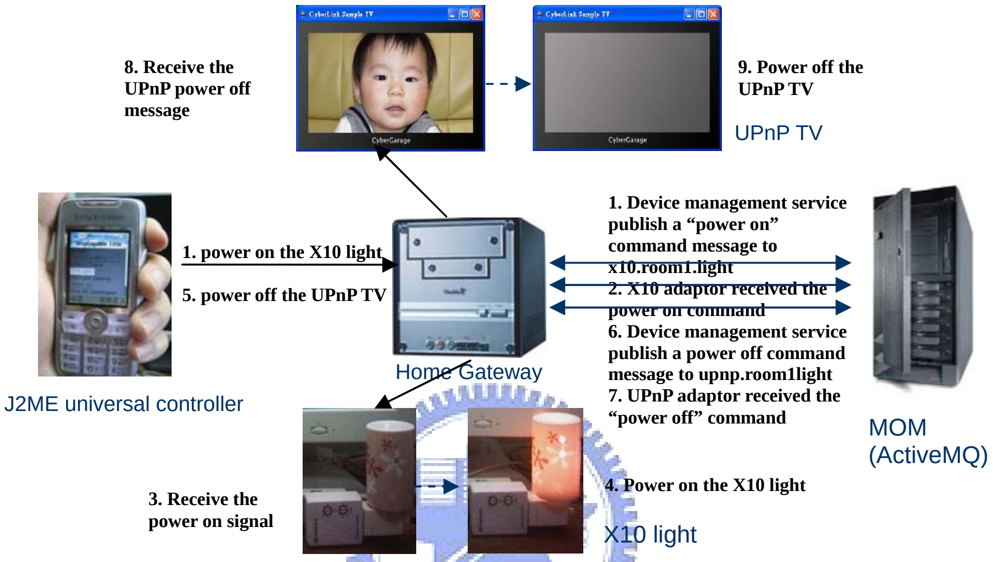

8. Receive the UPnP power off message
9. Power off the UPnP TV
UPnP TV
1. power on the X10 light
5. power off the UPnP TV
1. Device management service publish a "power on" command message to x10.room1.light
2. X10 adaptor received the power on command
6. Device management service publish a power off command message to upnp.room1light
7. UPnP adaptor received the "power off" command
Home Gateway
J2ME universal controller
MOM (ActiveMQ)
3. Receive the power on signal
4. Power on the X10 light
X10 light

Figure 6-6 the demonstration of universal control and the message flow

6.4 Rule-based Home Automation

6.4.1 Rule Management and Configuration

The user can add, edit or remove the Home Automation rule in MHAP gateway with the rules configuration terminal. The user can also control the appliances at home with the MHAP universal controller. The PocketPC version of the rule configuration terminal connects to MHAP home gateway through TCP/IP. After secure login, the user can manage and set the rules (Figure 6-7). The rule configuration tool retrieves all the appliances which are online in MHAP and displays the action and event supported by each appliance. The rule consists of event and actions and the user can set each part of the rule by click. The J2ME version of universal controller can also connect to MHAP home gateway through TCP/IP. However, the universal controller can be only used for controlling appliances and users cannot set the rules with it.

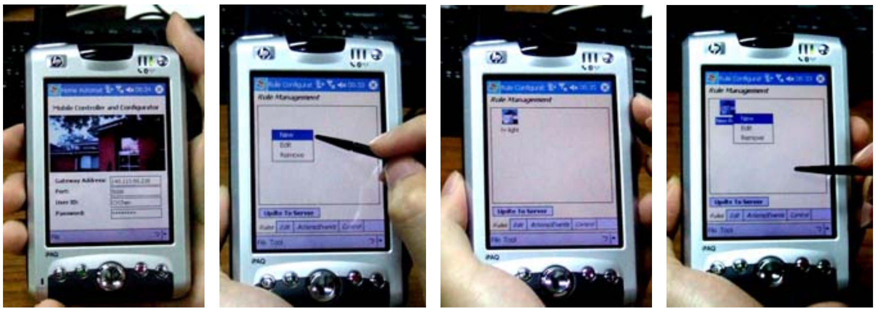


Figure 6-7 rule management with configuration console

In the demonstration, the user set a Home Automation scenario which consists of three rules (Figure 6-8). The first rule indicates the X10 electronic fan should be powered on when the state of the UPnP media player changes to “playing.” The second rule indicates the INSTEON dimmable light should be turned on when the state of the UPnP media player changes to “pause.” The third rule indicates the UPnP media player should resume playing when the state of the INSTEON dimmable light changes to “off.”

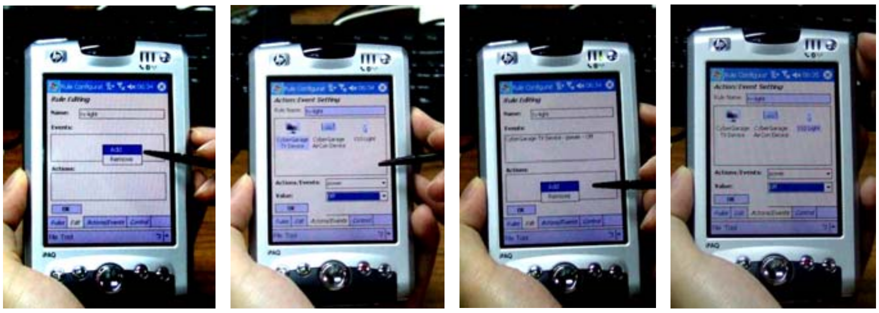
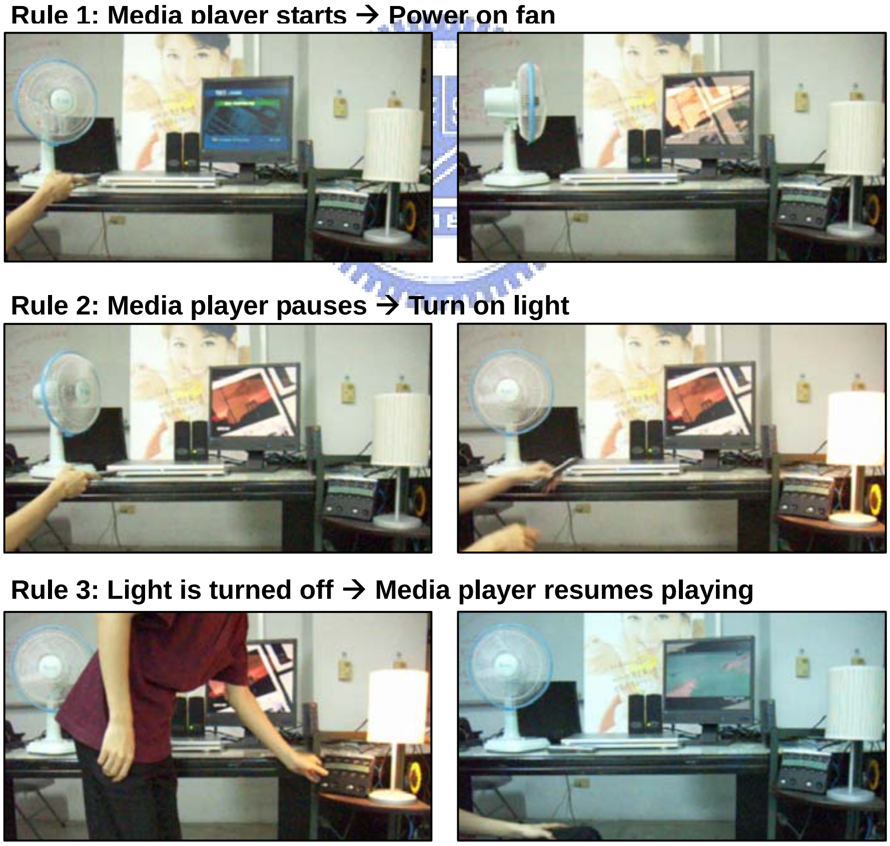


Figure 6-8 set the demonstration rule with configuration console

6.4.2 *Home Automation Executing*

After adding Home Automation rules, the MHAP will perform Home Automation operations according to the rules. Figure 6-9 shows the demonstration after setting the three Home Automation rules. When the UPnP media player had started to play, the X10 electronic fan was powered on automatically. When the person had paused the UPnP media player for walking out of the room, the state change of the UPnP media player triggered the INSTEON dimmable light to turn on. When the person had been back to the room and turned off the light, the UPnP media player resumed playing again.



Rule 1: Media player starts → Power on fan

Rule 2: Media player pauses → Turn on light

Rule 3: Light is turned off → Media player resumes playing

Figure 6-9 a demonstration of Home Automation accomplished by MHAP

6.4.3 Message Flow

Figure 6-10 indicates the message flow in rule 2 showed in Figure 6-9. The rule indicated that the INSTEON dimmer light should be turned on if the UPnP media player is paused. While the user had added the rule to MHAP with configuration tool, the rule engine subscribed the required topic “upnp.room1.avrenderer” to receive the event message. After the user paused the UPnP media player, the media player sent a native UPnP event message indicating the change of the state. The UPnP adaptor receiving the event message then published a MHAP event message to the topic “upnp.room1.avrenderer.” After the MOM provider had pushed the message to rule engine, the rule engine checked if any operation should be executed according to Home Automation rules. To finish the operation assigned in rule 2, the rule engine published the MHAP command message to the topic “insteon.room1.light” to turn on the light. The INSTEON adaptor received the message and bridged the message to INSTEON power line network. Therefore the dimmable light assigned by the device id in command message was turned on finally.

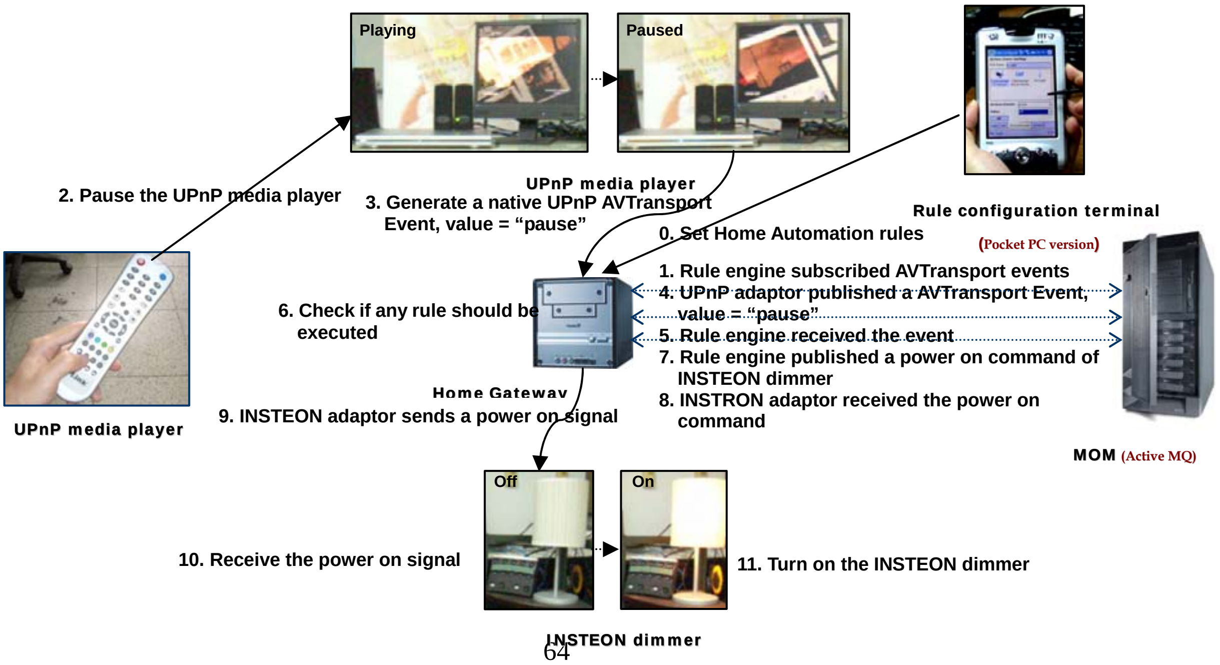

Playing
Paused
UPnP media player
Rule configuration terminal
(Pocket PC version)
2. Pause the UPnP media player
3. Generate a native UPnP AVTransport Event, value = “pause”
0. Set Home Automation rules
1. Rule engine subscribed AVTransport events
4. UPnP adaptor published a AVTransport Event, value = “pause”
5. Rule engine received the event
7. Rule engine published a power on command of INSTEON dimmer
8. INSTRON adaptor received the power on command
6. Check if any rule should be executed
Home Gateway
UPnP media player
9. INSTEON adaptor sends a power on signal
MOM (Active MQ)
Off
On
10. Receive the power on signal
11. Turn on the INSTEON dimmer
INSTEON dimmer

Figure 6-10 message flow in Rule 2 of demonstration showed in Figure 6-9

6.5 Scaling to Internet and Other Network

The services outside the home can also join to Home Automation through MHAP. Any legacy service which is implemented by any distributed technology such as MOM, Web service, CORBA [55] or RMI [56] can cooperate with the appliances at home. For example, the system of a home security company can subscribe the event of security sub system. Then when the security sub system at home detects a fire or a home burglar, the message will publish outside the home to the company. The MOM guarantees the delivery of the messages from the home to the company. The MHAP can also scales to Internet. For example, if there is an Internet service counting the most popular song playing at each home, the service can subscribe the event of media player at home. Then the playing event messages will be pushed to the Internet service. On the other hand, the event of Internet service can also trigger another Home Automation operation. For example, the UPnP media player can automatically play the most popular song every time when the Internet service publishes the event notifying that the hit parade is updated.

Chapter 7 Discussion and Conclusion

7.1 Benefit of MHAP

7.1.1 Event-driven Home Automation in Heterogeneous Environment

The MHAP support event-driven Home Automation in heterogeneous environment through MOM. MHAP not only schedules operations of different appliances but also let the state change of one appliance trigger operations of the other appliances which belongs to any kind of home network. The MOM-based core of MHAP makes Home Automation event-driven even in Heterogeneous. An event is a type of MHAP message. Through independent channels from each adaptor and application router to MOM, messages can flow form one home network to another incompatible home network. Therefore, an event form an appliance can trigger another automatic operation of another appliance in anther home network. MHAP makes every smart appliance cooperate together, even the appliances are using incompatible home networking technologies.

7.1.2 Device and Protocol Independent Platform

MHAP is completely device and protocol independent. MHAP import every smart appliance as a corresponding abstract MHAP appliance through XML-based descriptors. In message flow of MOM, the appliances communicate with each other through device and protocol independent MHAP messages.

7.1.3 Reliable Fault Tolerant Home Automation

MHAP offers the reliable operation and the fault tolerant function. Reliable operation meets the requirements for automatic operations of security. MHAP ensure that the operation remarked as “reliable” in descriptors will be done though the guaranteed message delivery introduced by MOM. The durable subscription and the persistent message mechanism of MOM guarantee the delivery of message involved in reliable operation.

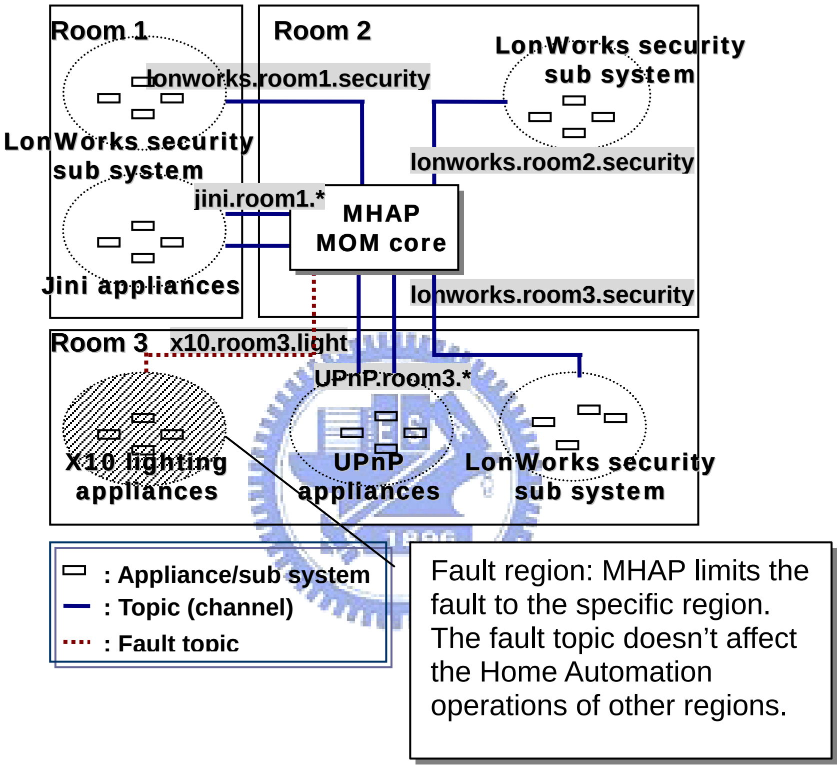


Figure 7-1 region division provides the fault tolerant and reconfigurable Home Automation

On the other hand, when appliances or channels fails, the fault tolerant function of MHAP keeps the system effective uninterruptedly. MHAP’s region division according to topic naming limits the damages into certain region, and therefore the damages don’t affect the other parts of Home Automation. In MHAP, the appliances belong to a set of properties are viewed as in a region and the property includes the location, the device type and the adaptor defined in topic naming. Since each region subscribe/publish the independent topic, the fail involved in a region will not affect the message flow of other regions. As Table 7-1 shows, when the channel of the X10 lighting appliances fails, the Home Automation operations of the other appliances still be finished. The damaged region uses channel “x10.room3.light,” and the other regions use the other independent channels to carry message flow. Therefore, MHAP limited the damage into the region of “x10.room3.light,” and the messages of Home Automation were still flowing freely. To take another example, when a channel of the fire alarm in room1 fails, the fire alarm in the other room can still publish event messages to the home security company though MHAP.

7.1.4 High Scalability

The MOM core makes MHAP scales well when the numbers of appliance increasing through one-to-many publish/subscribe messaging approach. On the other hand, a MHAP system can scale from home domain to Internet domain. Because the MOMs are able to cross sub net and connect to each other. The messages in one MHAP gateway can pass to other gateway outside home and push to Internet services. Such scalable characteristic makes the automation scope of MHAP extends from a home to a larger scope such as a building or campus. The components participate the automation can not only be appliances or sub systems at home but also Internet services.

7.1.5 High Extensibility

The infrastructure layer and MOM core guarantee the high extensibility of MHAP. MAHP consists of services over the infrastructure layer. The infrastructure uses OSGi to manage the life cycles of services, to update or install service through Internet and to check dependencies among service bundles. For example, when a new appliance is plugged in, the MHAP will check if there is appropriate adaptor for it. If there is no appropriate adaptor in MHAP gateway, the MHAP can get it through Internet. The user may also assign the URL of the adaptor service by himself and then the MHAP will install and activate the service dynamically. During all the processes such as plugging in the appliance, installing services or activating services, the current automation is not affected. The MHAP infrastructure layer guarantees that the service is activated or deactivated dynamically. Furthermore, when the user add a new appliance and configure a new rule related to both the appliance and old appliances, the operations of existing Home Automation doesn't need stopping or suspending. In MAHP, an appliance communicates with each other through message flow between each independent channel and MOM provider. The existing appliances subscribe and publish it own topics through adaptor services, therefore the new appliance does, too.

7.1.6 Easy to Use

MHAP offers an easy-to-use way to configure Home Automation. MHAP is device and protocol independent. Therefore, the user can configure the appliances with the same way even if the appliances supports different home networking technology. The MHAP separates Home Automation into rules and centralize the settings and configurations. Therefore, the user can configure Home Automation through editing rules of MHAP without multiple processes loading each macro program to each smart appliance.

7.2 Comparison between MHAP and UMB

The Universal Middleware Bridge (UMB) proposed by Kyeong-Deok Moon et al. is a close research to MHAP. But there are some differences between these two systems. Table 7-1 shows the comparison between MHAP and UMB.

7.2.1 Home Automation Support

MHAP is a complete platform for Home Automation while UMB focuses on interoperability problems of home networks. Although both MHAP and UMB integrate home networks dynamically, the UMB doesn't support any easy-to-use method to finish Home Automation. By contrast, MHAP is supporting rule-based method to configure Home Automation. Therefore, the Home Automation services for UMB should be re-implemented when the user want to change Home Automation regulations. But the MHAP's Home Automation is reconfigurable and the user can simply add or change the rules to reconfigure the scenario when the requirement of Home Automation changes.

7.2.2 Messaging

MHAP uses MOM with JMS interface as messaging component while UMB design uses proprietary routing table to finish routing. Therefore, MHAP introduces the high scalability and reliability of MOM, which is required in a Home Automation system. Although both UMB and MHAP use platform impendent messages to overcome the interoperability problem, the MHAP introduces MOM technology. Therefore the design and implementation of MHAP is simpler. Because of MOM, the routing action of MHAP is accomplished just by making different services such as rule engine or adaptor subscribe different channel. By contrast, the message routing in UMB needs complex operations such as looking into tables and processing packets, and requires complicated implementation.

7.2.3 Extensibility

On the other hand, MHAP consists of OSGi services, and all services can be downloaded, installed, updated, activated, deactivated and uninstalled through standardized method. Even the MHAP infrastructure layer can check the service dependencies among services. But in UMB, only the adaptor components can be changed.

Table 7-1 compare MHAP with UMB in Home Automation

Property	MOM-based Home Automation Platform (MHAP)	Universal Middleware Bridge (UMB)
Home Automation Supporting	Rule-based services	Only programming API
Messaging	MOM (JMS)	Proprietary routing table
Scalability	High	NA
Extensibility	High (All Services can be updated or extend)	Only adopters can be changed
Deployment mechanism	OSGi(with dependency check and life cycle control)	NA
Reliability	Reliable Operation	NA

7.2.4 Scalability

The scalability of MHAP depends on the MOM provider, and most MOM providers on market have well scalability. However, there is no experiment result showing the degree of the scalability of UMB. The MHAP works on deployment and service infrastructure, all the services including rule engine and adopters can be downloaded from Internet and be automatically installed and activated. By contrast, UMB provides no method to update its adaptor or other component through internet. Further more, the MHAP can cross subnet, connect to each other and scale to Internet. Therefore, the MHAP has high extensibility and could connect each home gateway to become a building automation platform and cooperate with Internet service with scalable MOM messaging.

7.2.5 Reliability

MHAP can gain stability by the durable subscription and persistent messages through MOM. But the UMB doesn’t focus on it and provide no mechanism or architectures to increase stability.

7.3 Conclusion

In conclusion, MHAP provide people a platform integrating every home network technology and human can do anything which software can do through home appliances. For example, if there is an OSGi service on home gateway which can transmit Morse codes to a certain place, through MHAP, the user can send Morse codes by turning on a light. All he has to do is to provide the MHAP descriptors.

MHAP introduces MOM to offer a device and protocol independent platform which supports event-driven Home Automation. MHAP's Home Automation is reliable and fault tolerant. The reliable operation which consists of delivery-guaranteed messages meets the security application requirement. MHAP's region division according to topic naming provides the fault tolerant and reconfigurable Home Automation. MHAP can limit fault in specific region which belongs to a topic because the messages in MHAP flow among independent channels connected to MOM respectively. The independent channel design also offers user a complete reconfigurable Home Automation. The user can dynamically change a certain part of the Home Automation scenario and the other parts are still effective. New protocol, device and Internet services can dynamically join to the current Home Automation without any affection such as suspending operations. Through pushing cross-subnet messages outside home, MHAP can scale to Internet or other network and can cooperate with any legacy Internet services or security systems of home security companies. The Internet service can be a service supporting any distributed technology such as JMS, Web services or CORBA.

Moreover, MHAP has high extensibility through the infrastructure layer. All components of MHAP are services and can be automatically deployed from Internet.

The infrastructure layer introduces OSGi to install, update, activate, deactivate and uninstall correct services according to the dependency among the services. For user, MHAP provide the easy-to-use way to configure complex Home Automation. A user can simply add, delete or edit some rules to configure or reconfigure the complex scenario of the Home Automation.

7.4 Future work

7.4.1 Security

MHAP will make Home Automation more convenient through the scalability and extensibility introduced by MOM. Some events may across the range of a home and the MOM can work cross subnet indeed. Therefore, the security mechanism will be discussed more in such situation [58]. MHAP will provide the MOM core the filtering function to retain the security of home when the MHAP scales to Internet and may push some messages to Internet services or other MHAP gateways. Filtering mechanism can also protect a MHAP gateway against the dangerous messages from other MHAP gateways or Internet services.

7.4.2 Integrating Sensor Network

Nowadays, sensors networks are famous in domains of everyday life such as health monitoring [59]. MHAP will integrate the home networks with sensors networks to offer more convenient application for human life. The Home Automation will be close to people through sensor networking technologies.

7.4.3 MHP Support

The infrastructure layer that MHAP supports presently is OSGi. However, some other gateway middleware such as Multimedia Home Platform (MHP) [60] may be chosen. Implementing the MHP version of MHAP is another future work. In the MHP version of MHAP, the MHAP services can be updated through terrestrial broadcast. Furthermore, the interactive of Digital TV services can join the Home Automation at home.

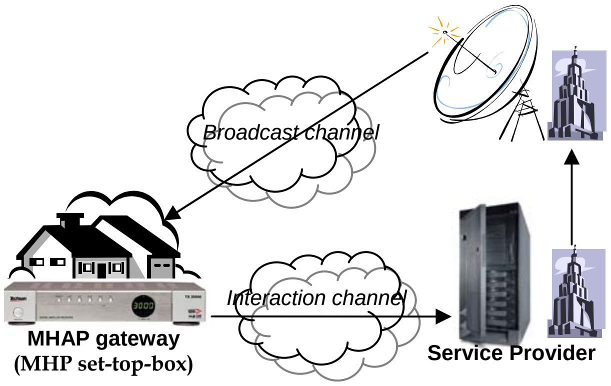


Figure 7-2 Digital TV services join the automation through MHAP in the future

Bibliography

[1] J.L. Ryan, Home Automation, Electronics & Communication Engineering Journal, Vol.1, Issue 4, pp. 185-192, July 1989

[2] J.G. Turnbull, Introducing home area networks, BT Technology Journal, Vol. 20, No 2, pp. 33-38, April 2002

[3] X10 web site,
http://www.x10.com

[4] INSTEON web site,
http://www.insteon.net/

[5] UPnP Forum,
http://www.UPnP.org/

[6] HAVi- Home Audio Video Interoperability,
http://www.havi.org/

[7] G. Banavar, T. Chandra, R. Strom, and D. Sturman, A Case for Message Oriented Middleware, In 13th Int. Symp. on Distributed Computing (DISC), 1999.

[8] Open Services Gateway Initiative alliance,
http://www.osgi.org

[9] B. Rose, Home networks: a standards perspective, IEEE Communications Magazine, Vol. 39, Issue 12, pp. 78-85, Dec. 2001

[10] Chris Adams, Home area network technologies, BT Technology Journal, Vol. 20, No 2, pp 53–72, April 2002

[11] Jini.org,
http://www.jini.org

[12] Echelon Co., “LonTalk Protocol Specification ver 3.0,” 1994

[13] Dong-Sung Kim, Jae-Min Lee, Wook Hyun Kwon, In Kwan Yuh, Design and implementation of home network systems using UPnP middleware for networked appliances, IEEE Transactions on Consumer Electronics, Vol.48, Issue 4, pp. 963-972, Nov. 2002

[14] B.A. Miller, T. Nixon, C. Tai, M.D. Wood, Home networking with Universal Plug and Play, IEEE Communications Magazine, Vol. 39, Issue. 12, pp.104-109, Dec 2001

[15] Choonhwa Lee and Sumi Helal, Protocols for service discovery in dynamic and mobile netwroks, International Journal of Computer Research, Volume 11, No. 1, pp. 1-12, 2002

[16] R. Lea, S. Gibbs, A. Dara-Abrams, E. Eytchison, Networking home entertainment devices with HAVi, Computer, Vol. 33, Issue 9, pp. 35-43, Sep. 2000

[17] J.M. Dricot, De Doncker, M. Dierickx, F. Grenez and H. Bersini, Development of distributed self-adaptive instrumentation networks using Jini technology, IEEE International Workshop on. VIMS 2001, May 2001

[18] Jim Waldo, The Jini architecture for network-centric computing, Communications of the ACM, Vol. 42, Issue 7, pp. 76-82, July 1999

[19] R.E. Filman, Embedded internet systems come home, IEEE Internet Computing, Vol. 5, Issue 1, pp. 52-53, Jan. 2001

[20] D. Marples and P. Kriens, The Open Services Gateway Initiative: An Introductory Overview, IEEE Commun. Mag., vol. 39, no. 12, Dec. 2001

[21] D. Valtchev and I. Frankov, Service Gateway Architecture for a Smart Home, IEEE Communications Magazine, Vol. 40, Issue 4, pp. 126-132, April 2002

[22] P. Dobrev, D. Famolari, C. Kurzke, B.A. Miller, Device and service discovery in home networks with OSGi, IEEE Commun. Mag., Vol. 40, Issue 8, pp. 86-92, Aug 2002

[23] Do-Guen Jung, Kwang-Jin Paek and Tai-Yun Kim, Design of MOBILE MOM: Message oriented middleware service for mobile computing, International Workshops on Parallel Processing, pp. 434-439, 1999

[24] M. Caporuscio; A. Carzaniga, A.L. Wolf, Design and Evaluation of a Support Service for Mobile, Wireless Publish/Subscribe Applications, IEEE Transactions on Software Engineering, Vol. 29, Issue 12, pp. 1059-1071, Dec. 2003

[25] Eiko Yoneki and Jean Bacon, Gateway: A Message Hub with Store-and-Forward Messaging in Mobile Networks, International Conference on Distributed Computing Systems Workshops, pp. 348- 353, 2003

[26] Sun Microsystems, Java Message Service Specification Version 1.1, April 2002

[27] S. Chemishkian, J. Lund, Experimental Bridge LONWORKS / UPnP, CCNC 2004

[28] J. Allard, V. Chinta, S. Gundala, G. G. Richard III, J. Allard, V. Chinta, S. Gundala, G. G. Richard III, Jini Meets UPnP: An Architecture for Jini/UPnP Interoperability, SAINT'03, 2003

[29] E. Guttman and J. Kempf, Automatic Discovery of Thin Servers: SLP, Jini and the SLP-Jini bridge, IECON'99, 1999

[30] Bluetooth ESDP for UPNP, http://www.bluetooth.com/pdf/ESDP_UPnP_0_95a.pdf

[31] D. Ueno, T. Nakajima, I. Satoh, K. Soejima, Web-Based Middleware for Home Entertainment, ASIAN'02, 2002

[32] Kyeong-Deok Moon, Young-Hee Lee, and Young-Sung Son, Chae-Kyu Kim, Universal Home Network Middleware Guaranteeing Seamless Interoperability

among the Heterogeneous Home Network Middleware, IEEE Transactions on Consumer Electronics, Vol. 49, No. 3, August 2003

[33] Kyeong-Deok Moon, Young-Hee Lee, Chae-Kyu Kim, Context-aware and adaptive universal home network middleware for pervasive digital home environment, CCNC 2004

[34] E. Tokunaga, H. Ishikawa, M. Kurahashi, Y. Morimoto, T. Nakajima, A Framework for Connecting Home Computing Middleware, ICDCSW'02, 2002

[35] Kyeong-Deok Moon, Young-Hee Lee, Chang-Eun Lee, Young-Sung Son, Design of a universal middleware bridge for device interoperability in heterogeneous home network middleware, IEEE Transactions on Consumer Electronics Vol. 51, Feb. 2005

[36] Bill LaPlant, Shari Trewin, Gottfried Zimmermann, and Gregg Vanderheiden, The Universal Remote Console: A Universal Access Bus for Pervasive Computing, IEEE Pervasive Computing, Vol. 3, Issue 1, pp. 76-80, Jan. 2004

[37] J. Latvakoski, D.Pakkala, P. Paakkonen, A Communication Architecture for Spontaneous Systems, IEEE Wireless Communications Magazine, Vol. 11, Issue 3, pp. 36 – 42, Jun. 2004

[38] M. H. Handley, E. Schulzrinne, and J. Rosenberg, SIP: Session Initiation Protocol, RFC 2543, Mar. 1999.

[39] Oscar - An OSGi framework implementation http://oscar.objectweb.org/

[40] R.S. Hall, H. Cervantes, An OSGi implementation and experience report, CCNC 2004, Jan. 2004

[41] mBedded Server OSGi Framework http://www.prosyst.com/products/osgi_framework.html

[42] IBM Service Management Framework

http://www-306.ibm.com/software/wireless/smf/

[43] Knopflerfish - Open Source OSGi

http://www.knopflerfish.org/

[44] ActiveMQ,

http://www.activemq.org/

[45] IBM Software - WebSphere MQ

http://www-306.ibm.com/software/integration/wmq/

[46] Softwired iBus,

http://www.softwired-inc.com/

[47] SwiftMQ Router,

http://www.swiftmq.com/products/index.html

[48] TIBCO Enterprise Message Service,

http://www.tibco.com/software/messaging/enterprisemessageservice.jsp

[49] Fiorano - FioranoMQ Home Page,

http://www.fiorano.com/products/fmq/overview.htm

[50] JBoss.com - JBoss Messaging,

http://www.jboss.com/products/messaging

[51] OpenJMS

http://openjms.sourceforge.net/

[52] The Open Services Gateway Initiative, "OSGi Service Platform Release 2 Specification ," IOS Press, March 2003

[53] Sun Microsystems, JAR File Specification, 2003

http://java.sun.com/j2se/1.5.0/docs/guide/jar/jar.html

[54] Web Services

http://www.w3.org/2002/ws/

[55] S. Vinoski, CORBA: integrating diverse applications within distributed heterogeneous environments, IEEE Communications Magazine, Vol. 35, Issue 2, pp. 46-55, Feb. 1997

[56] Java Remote Method Invocation (Java RMI)
http://java.sun.com/products/jdk/rmi/

[57] Project CyberLink for Java
http://sourceforge.net/projects/cgUPnPjava/

[58] P. Bergstrom, K. Driscoll, J. Kimball, Making Home Automation communications secure, Computer, Vol. 34, Issue 10, pp. 50-56, Oct. 2001

[59] V. Tsetsos, G. Alyfantis, T. Hasiotis, O. Sekkas, S. Hadjiefthymiades, Commercial wireless sensor networks: technical and business issues, Second Annual Conference on Wireless On-demand Network Systems and Services, Jan. 2005

[60] Multimedia Home Platform,
http://www.mhp.org/