REMOTE SERVICE PROVISION FOR CONNECTED HOMES

A thesis submitted to the Faculty of Technology in fulfilment of the requirements for the degree of Doctor of Philosophy at De Montfort University

by

Wei Feng

Mechatronics Research Group

August 2010

Declaration

I hereby declare that no part of material described in this thesis has been submitted for award of any other degree or diploma in this or any other university or college of advanced education

Acknowledgements

Here to address my sincere gratitude to many people without their support this work would not have been possible.

First of all, I would like to express my gratitude to my first supervisor Mr. C.B.Wong who gave me the opportunity for this research study. I also would like to express my gratitude to my supervisor Dr. Xi Chen for sharing his knowledge with me and Dr. S.K. Chong for his great support during my work on the DTI funded research project, Equipment Management Trial. Many thanks also go to my colleagues, Dr. H.T. Pan and Mr. H.B. Jia in the De Montfort University for their constructive discussions.

My special thanks go to my dear friends Mark Ingle, Tony Grayson and lovely Ninicat, who help me to correct my grammar, provide accommodation and continuous friendship during my stay in Leicester.

I would like to express my great appreciation to my partner Aidan McIvor who is always there for me. Thanks for giving me a warm and comfortable “den”. Thanks for being tolerate of my occasional grumpiness.

Finally I would like to thank for my parents who always give me great support, and encourage me to be strong, not give up during the difficult time.

Abstract

The advancement of Information and Communication Technology (ICT), the tremendous pace of miniaturisation and cost reduction in electronics devices in the last decade have encouraged a proliferation of networked home devices, appliances and systems with built-in "intelligence". Furthermore, pervasive home Internet connections in last five years have created abundant opportunities for connected home services ecosystem to foster a wide range of innovative services, which would improve the quality of life. Domestic systems and environments are cost sensitive and "No home is alike", which leads to a set of specific technical challenges in realising remote services provision to connected home environments. Significant research efforts have been carried out to develop service provision models, architectures and tools worldwide. However, the current designs and systems are predominantly one-size-fits-all, which compromises their ability to handle the diversity of service requirements between both service providers and recipients. This research study aims to tackle these challenges via the development of an integrated and open remote service delivery platform which not only promotes interoperability to enable "Choice" and facilitate service "Mass Customization", but also accelerate take-up and create the environment to foster innovative services through third party applications and plug-ins without discrimination.

This research study proposed to view a remote service delivery system from three distinct perspectives: connected home environments (user perspective), remote service delivery platform (service enabler), and remote service providers (service provider perspective); to establish a holistic view on the requirements of remote service provision to connected home environments. A reference architecture for remote service provision based on the proposed views has been devised, which provides built-in support for an "On-Demand" operating model and facilitate "Freedom of Choice" via different levels of interoperability (from device to service levels).

The concept of Remote Service Object (RSO) model pioneered by TAHI, a non-profit organisation, has been adopted to encapsulate the details of physical (e.g. device and

appliance) and logical (e.g. environment information) objects and present with well-defined interfaces for both local and remote access to its service(s). This research study has enriched RSO conceptual specifications with solutions, and converted it into the basic building blocks of an advanced and flexible remote service delivery architecture to realise the vision of "On Demand" service provision with service "Mass Customisation" that lead to "Freedom of Choice".

This research proposes the concept of "adaptive" messages (with well-defined message format and dictionary) to facilitate "Resources Sharing" between different types of services. A lightweight service execution engine, which adopts a well accepted Web service management standard - BPEL, is built upon the home service gateway to manage interoperability aspects such as available service resources between connected homes and remote service providers. The proposed methodologies and associated tools support the aggregation of the relevant home service resources into customised service applications to connected homes, which would meet the needs of individual homes regardless of the differences of their home settings. This aggregation process is the corner stone to realise service "Mass Customisation".

This research study has developed two types of service applications, namely a remote security monitoring service and advanced information service with the aid of a project demonstrator that has been developed to evaluate the proposed methodologies and associated tools and illustrate the proposed concepts for remote service provision. The proposed RSO model has proven to be effective in supporting service interoperability via service-based modelling, regardless of the underlying technology of the home resources. The home service gateway has proven to be an open and flexible platform which can enable various service providers to create different types of service applications without discrimination. The proposed "Resources Sharing" approach has also proved to be efficient in resource consumption regardless the types of service applications. This research introduces new concepts such as "On-Demand", "Mass Customisation" and "Resource Sharing" in remote service provision to connected home environments. These new concepts not only create a new horizon to both service providers and recipients, but also reducing the "Total Cost of Ownership" and could help accelerate the market take-up.

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

AmI:	Ambient Intelligence
APIs:	Application Programming Interfaces
ARP:	Address Resolution Protocol
ASF:	Apache Software Foundation
BPEL:	Business Process Execution Language
CE:	Consumer Electronics
CMS:	Context Management Service
CU:	Communication Unit
DAML:	DARPA Agent Markup Language
DARPA:	Defence Advanced Research Project Agency
DHCP:	Dynamic Host Configuration Protocol
DNS:	Domain Name System
DTI:	Trade and Industry
FXPP:	Flexible XML Processing Profile
GENA:	General Event Notification Architecture
GP:	General Practices
HAN:	Home Automation Network

HAVi: Home Audio Video Interoperability

HTML: HyperText Markup Language

HTTP: Hypertext Transfer Protocol

HTTPMU: HTTP Multicast over UDP

HTTPU: HTTP unicast over UDP

HVAC: Heating, Ventilation & Air-condition

ICANN: Internet Corporation for Assigned Names and Numbers

ICT: Information and communication technology

IHVs: Independent Hardware Vendors

ISP: Internet Service Providers

IT: Information Technology

JSP: JavaServer Page

JVM: Java Virtual Machine

MA: Mobile Agent

MRG: Mechatronics Research Group

NCAP : Network Capable Application Processor

ODE: Orchestation Director Engine

OIL: Ontology Inference Layer

OSGi: Open Services Gateway initiative

PC: Personal Computing

PSA: Pervasive Service Agent

QoS: Quality of Service

RDF: Resource Description Language

RSO: Remote Service Object

SaaS: Software as Service

SDF: Service Delivery Framework

SDP: Service Delivery Platform

SLP: Service Location Protocol

SOAP: Simple Object Access Protocol

SOS: Sensor Operation System

SSDP: Simple Service Discovery Protocol

STB: Set-top box

TAHI: The Application Home Initiative

TEDS: Transducer Electronic Datasheet

TII: Transducer Independent Interface

TIM: .... Smart Transducer Interface Module

TOA: TAHI Open Architecture

UDDI: Universal Description Discovery, and Integration

UPC:	Universal Product Code
UPnP:	Universal Plug and Play
URI:	Uniform Resource Identifier
URL:	Uniform Resource Locator
URN:	Uniform Resource Name
UUID:	Universal Unique Identifiers
XML:	Extensible Markup Language
W3C:	The WWW Consortium
WSDL:	Web Service Description Language

Chapter 1 Introduction

1.1 Research Background

In 2008, about 16.46 million households in the UK had Internet access (National Statistics, 2008). This represented 65 percent of households and an increase from 61 percent in 2007. Of all UK households, 56 percent had broadband Internet access in 2008. Globally, households with Internet connection will grow from 114 million at the end of 2006 to estimated 160 million at the end of 2012 (Parks, 2008). Benefitting from the increasingly competitive market, Europe is in the midst of a home networking boom, at a faster growing rate than North America and the Asia-Pacific region. Meanwhile, 8 billion devices with embedded microprocessors are produced every year, and the number is expected to rise dramatically over the next decade. These devices range in size from a few millimetres to several metres and may be interconnected via wired and wireless technologies into a broad network (Parliamentary Office, 2006). These developments are providing a great opportunity to have control over the surrounding environment with “intelligence”, which can enhance and enrich people’s quality of life.

Many home-based automated systems already exist to provide different types of services to the household, from lighting and residential climate control, to home theatre, audio entertainment systems and domestic security (Farlex, 2005). Traditionally, these systems have been developed independently by different service providers using different standards and built on different types of platform (Wang, 2006). For instance, the smart home health care system typically connects devices directly to the health care service provider; a monitored security service also requires connecting various related devices to the security service provider (Gupta, 2002). These systems currently are developed independently with limited consideration on interoperability, if there is any. Such ‘Closed’ systems will cause unnecessary high cost in system resources (e.g. sensors, home gateway, and network) due to an inability of “sharing resources” (e.g. information from motion detection sensors can be shared between domestic security and home healthcare systems) when operating within the same environment. Furthermore, these ‘Closed’ systems will likely increase the operation burden to their users because of likely inconsistent user interfaces. To take the advantage of the emerging smart home environment and reduce system cost through “resource sharing”, it is necessary to integrate service resources into a general platform and share them between different service systems.

A home may include a range of devices, from very small sensors to networked appliances and computers. These devices and systems are likely developed with different network and protocol standards, which cause complexity in managing them working in a coordinated manner to deliver advanced services. The domestic environment is not like the enterprise and industrial environments; it is unlikely to have a system administrator at home to manage the complexity of the devices configuration, integration and communication issues. Home users are well known for expecting user-friendly interfaces and easy to use functionality without involving any technical details. In addition, the householder should still have the freedom to replace current appliances with new models or relocate them within the house without impacting the existing services. This means that the home could be considered as an unstructured, dynamic and heterogeneous environment, which will require a service delivery platform to manage devices and services automatically without the home users’ attention.

Moreover, the home settings are likely to be influenced by location, age (of the home-user), culture, or household income. Generally speaking, no two homes are alike. Householders might also expect different services customized to their preferences. These possibilities increase the complexity of service delivery to the home environment. Based on the existing service delivery models, the current design of service is predominantly one-size-fits-all, regardless of the difference between home settings and user requirements that allow home users to have little or no choice. For example, all leading domestic security monitoring services are contract based, which typically cost £300-£400 annually (http://www.adt.co.uk; Smarthome, 2009), which few families can afford or are willing to pay. In many cases, it is recognized that burglars are mostly opportunists taking advantage of when the property is empty, such as when the residents are on holiday, away for weekend, shopping, etc. Instead of paying for the expensive annual service, most householders are likely to prefer a “needed to" basis monitoring service even it is slightly more expensive on a daily basis. Obviously, such "On-Demand" services business model requires an advanced and flexible service delivery architecture (or platform) to realize its vision, which is not currently available.

In addition, smart home technologies are undergoing enormous pace of development with emerging new technologies and maturing of new service domains (Helal, 2005; Ricquebourg, 2006; Alison, 2008). The design of a service delivery architecture should not only be considered from a technical point of view, but also from its capability to provide advanced services with great ease and efficiency, which is the main issue for user acceptance (Yamazaki, 2006). To promote the provision of a wide variety of services to the residential market, we need to encourage more companies' to be involved. This requires the service platform to be non-proprietary, or at least non-discriminatory in its use of third party applications and plug-ins.

However, there is no such service delivery platform available for the connected home. Developers face the difficulty of connecting devices to the platform and using it to create services; and services are not so easy to use for the general public. Furthermore, there are not enough attractive services available to entice customers to buy and install the necessary infrastructure support. Usability and attractive services are the main issues in creating and/or accelerating market take-up and elevating company interest for investment. The amount of users is determined by the cost of devices, infrastructure and services, quality of services, as well as the user-friendly interface (OSGi, 2008).

The proposed research will review the enabling technology, available standards and platforms. Domestic users are not like commercial and industrial users. They are comparatively more cost sensitive, which determines the domestic service delivery platform relatively "thin". A good analogy is the current TV set-top box, which runs with the software specifically designed for the specific services with tight hardware constraints. As in many situations, it is not the technical feasibility, but rather the estimated profitability of the services that decided for a "Go" or "No Go" decision. This research proposes a service delivery platform that could support a "Service-on-Demand" business model (Meixell, 2008; http://www.mrg.dmu.ac.uk), which aims to meet various customer requirements with a single service delivery platform and improve service provision through “Choice” and “Mass Customization”. Under this model, the service provider will deliver services regardless of home infrastructure variations, and provide services on an as needed basis. For example, instead of committing into a particular long-term service contract, the householder could order the required service in a preferred time-limited period. This model not only reduces the overall cost of services for the householder, but also opens up the markets for the majority and attracts occasional customers. Also, instead of providing a one-size-fits-all service to all customers, service providers can learn the home settings via advanced information and communication technology (ICT) and customize the service to meet the specific requirements of an individual householder. This research has a specific focus on domestic security and healthcare service applications, as both have a large potential market and could be transformed into real products in the foreseeable future.

1.2 Research aim and objectives

- To study and define an open service architecture to facilitate “On-Demand” remote service delivery to heterogeneous connected homes.
- To design an associated middleware framework, supporting services and user interfaces to realize an “eService-On-Demand” business model with the concepts of “Choice” and service “Mass Customization” for connected homes.
- To prototype a Service-On-Demand service system with self-validation capability using the platform formulated and the associated methodology.
- To define an evaluation methodology against the proposed service systems and methodologies.
- To verify the proposed methodology using the service system and platform prototypes.

1.3 Thesis Organisation

The thesis is organised in 8 chapters. An overview structure of the thesis is displayed as follows:

Chapter 1: (Introduction) introduces the research background and problem statement for the research.

Chapter 2: (Background and Emerging Technologies) reviews the emerging technologies and methodologies for service provision. The review focuses on: (i) the definition for the service and the key characteristics of service delivery to connected homes; (ii) networked devices and applications; (iii) service discovery methodologies; and (iv) service aggregation approaches.

Chapter 3: (Related Works on Service Delivery Systems) reviews the enabling architecture for service delivery and the emerging research in this field. The review also identifies the technical challenges for service provision.

Chapter 4: (Analysis and Design for the Service Delivery System) proposes a scheme of developing a remote service delivery system that would be able to support "Service on Demand" business model with the concept of "Mass Customization". A reference service architecture which encapsulates all the key components and their relationships is proposed and described.

Chapter 5: (Realization of Remote Service Delivery System for the Connected Home) proposes a remote service delivery system with associated middleware framework, supporting services and user interfaces. The proposed system could realize the "eService-On-Demand" business model embedding the concepts of "Choice" and "Mass Customization" for the connected home.

Chapter 6: (Implementation of the Remote Service Delivery System and Supporting Tools) describes implementation of the remote service delivery platform and supporting tools.

Chapter 7: (Test Case) describes how the proposed research methodologies and prototypes are verified with test cases that are based on two domestic service applications.

Chapter 8: (Conclusions and Future Work) concludes this research study, identifies the contributions to knowledge and future work.

Chapter 2 Background and Emerging Technologies

2.1 Service and Service Delivery

As the proposed research is about remote service delivery to connected homes via advanced ICT, the definition of "Services" must be clearly defined to avoid any confusion with other definitions in other research areas (e.g. services in a customer service department).

2.1.1 Definition of Service

Service is a popular word that is used in almost every aspect of life. In computing and communication, a service is a coarse-grained processing unit that consumes and produces sets of objects passed-by-value or message. The service consists of a connection of components that is able to deliver the business functions that service represents (Endrei, 2004). A service may be a computation, storage, and a communication channel to another user, a software object, a hardware device, or another user (Jini, 2005). Services are similar to traditional software components as each service responsible for its own domain, which typically translates into limiting the scope to a specific function or a group of related functions (Erl, 2005).

Moving beyond e-business and e-commerce, Hewlett Packard (HP) first developed a business concept called an e-service. "An e-service is a service available via the Internet that completes tasks, solves problems, or conducts transactions." (Sahai, 2000) Using HP's e-service concept, any information resource or application program is a potential e-service; and Internet Service Providers (ISP) and other companies are logical distributors or access points for such services (Erl, 2004). In HP's vision, Information Technology (IT) departments will increasingly address their needs in a structured model so that individual models have the potential to be part of an e-service.

Conversely, the high cost of creating and maintaining software and hardware infrastructures for delivering services has led to a notable trend toward the use of third-party service providers, which rent out network presence, computing power and data storage space to clients which need such infrastructures. This is called information/application outsourcing or Software as a Service (SaaS) (Selcuk, 2009). More and more companies realize that the most important and valuable Internet resources are those that provide services (salesforce.com; Alibab.com). Here, service means those web sites that do not merely provide static information but allow one to effect some actions or changes, such as the control of physical devices. Leading market research firm IDC recently forecasts that worldwide revenue associated with SaaS would reach nearly $11 billion by 2009; and that it is expected to reach $14.5 billion in 2011, representing a compound annual growth rate of 30% (TenWolde, 2007; Heiman, 2009).

2.1.2 Connected Homes and Potential Services

A connected home environment, also called a smart home, is a dwelling incorporating a communications network that connects the key devices such as sensors and actuators, electrical appliances and services, which allows them to be remotely controlled, monitored or accessed. There are three important elements required to make a home environment smart: both local network and Internet connections; a service delivery platform to manage and control service systems; and devices within the homes that need to be connected and linked to the services and systems outside the home (King, 2003).

The roots of connected homes can be traced back to 1970s home automation technologies (Yamazaki, 2006). Unfortunately, besides the infrared remote controllers for appliances, these technologies failed to make any significant inroads to our lifestyle for several reasons. First, effects brought about by these technologies were insignificant compared to the installation costs, and they tended to increase the total household energy consumption. Also, systems based on these technologies were not flexible enough such that networks needed to be re-constructed for any changes in the systems, and these systems were not interoperable. Moreover, these technologies lacked applications and services that could attract householders.

Since the 1990s, as a result of the Internet boom and advancement of ubiquitous computing, researchers have started connected home projects all around the world (Weiser, 1991; Mitsuhiro, 1997; Cory, 1999; Day, 2002; Noury, 2005; Etter, 2007; Park, 2009). Although different projects had different goals, they all predicted connected homes will be the future. Regardless of the difference between socio demographic characteristics such as region, culture, age, and income, applications and services of the connected home could be categorised into several main areas. These service areas and their potential e-services have a great potential to improve the quality of life and examples include such as those below:

(1) Environment – The Environment aspect includes how efficient we use gas, water and electricity within the domestic environment (Ricquebourg, 2006). The potential e-services in this area could be:

Reduce the energy consumption and its cost through advanced energy management services without impacting the quality of life such as "Demand Side Management", "Load Shifting" and advanced heating and lighting control to deliver their services "On Demand" basis and maintain optimal settings in spite of different environmental conditions.

(2) Security and Safety – The security and safety aspects include alarm and access control (Nicholl & Perry, 2007). The potential e-services in this area could be:

- Advanced security alarm system which supports "On-Demand" monitored security services, enables "Mass Customization" and offers choice to householders that never appeared previously.
- Advanced remote care services for disadvantaged people (e.g. disabled or elderly) that live independently without safety concerns.

(3) Healthcare – Significant amount of research effort has been directed towards remote healthcare service delivery to the home environment and the results have clearly shown

its benefits to healthcare service providers and recipients. Potential healthcare services can be (Inclusion, 2006):

- Remote healthcare support from medical professionals such as medication reminders, activity monitoring related to safety (e.g. falls detection) and physiological measurements for health monitoring (e.g. blood pressure, temperature and blood glucose) (Weerasinghe, 2009).
- Advanced healthcare services using audio/video media for remote health consultation and monitoring (Istepanian & Laxminarayan, 2007).

(4) Equipment Management – Services related to domestic equipment such as maintenance services for domestic appliances and heating systems.

Remote maintenance services such as providing remote check-up for those appliances under contracted service or guaranteed periodically. Customers would able to benefit from advanced services such as predictive maintenance.

(5) Home entertainment – These services could potentially be more customized to personal preferences regarding interest, space and time. For example, Real-time interactive entertainment services such as games and chat shows.

Nowadays, when discussing connected home devices, predominantly are media related, including digital TV receivers, game consoles digital media service and personal video recorders, etc. (www.connected-home-news.com) TV/PC convergence is expected to be the next honey pot for telecoms companies (iptvWorldforum, 2008). Other types of potential services to the connected homes are much less apparent to the public. In fact, a lot of potential services for the connected homes are highly valuable and also potentially help to tackle some tough social problems. Using domestic security service as an example: domestic burglary is a key crime for any neighbourhood and community, and a major concern for the general public. Although the police have invested resources in attempting to reduce the incidence of burglary, with the credit crunch and consequent rise of unemployment, figures from the British Crime Survey showed a 25% rise in thefts comparing October to December 2008 with the same period in 2007 (www.homeoffice.gov.uk). Furthermore, existing burglar alarms are still primitive technically with a high level of false alarms, which lead to annual maintenance and monitoring contracts being mandatory to qualify for police response (ACPO, 2006). However, currently monitored security services are mainly based on annual contracts, which are relatively expensive for the majority. Connected homes with appropriate infrastructure support would facilitate the monitored security service “On-Demand” basis, which could lower the entry cost for the majority and open up the market for the service providers.

In addition, life expectancy has continually increased (UN, 2008). In 2008, 16% of U.K. population is aged 65 or over (National Statistics, 2008). Particularly in developed countries, the declines of both mortality and fertility rates will lead to an increase in the percentage of the aging population. In the United States, the population of Americans aged 65 and older during the next 20 years is expected to double (Merck, 2008). By 2030, it is estimated that there will be 71 million Americans who are senior, roughly 20% of the U.S. population. The aging population together with the change of life style (e.g. poor diet and insufficient exercise) causes an increasing demand for healthcare, particularly with chronic diseases (e.g. obesity, diabetes and cancer). With the help of connected home facility and appropriated care services, elderly can live independently without impairing their quality of life in their familiar home environments (www.freedomeldercare.com).

It is clear that there are numerous opportunities and huge market potential for connected homes. The proposed research focuses on formulating an integrated service delivery platform that would not only support remote services delivery, but also enable service “Mass Customisation” and “Choice” for both service providers and service recipients.

2.1.3 Service Supply Chain for Connected Homes

For any service or service application whether provided remotely or locally in the home environment, there are a set of entities that relate to the creation of service data or information, its delivery, service aggregation, and operation based on the received information from the connected home, and presented to a user or service provider (Gilbert, 2004). These entities can be categorised into four major functional groups:

- Definition and creation of the service
- Management, aggregation and presentation of the service
- Delivery and distribution of the service
- Consume of the service

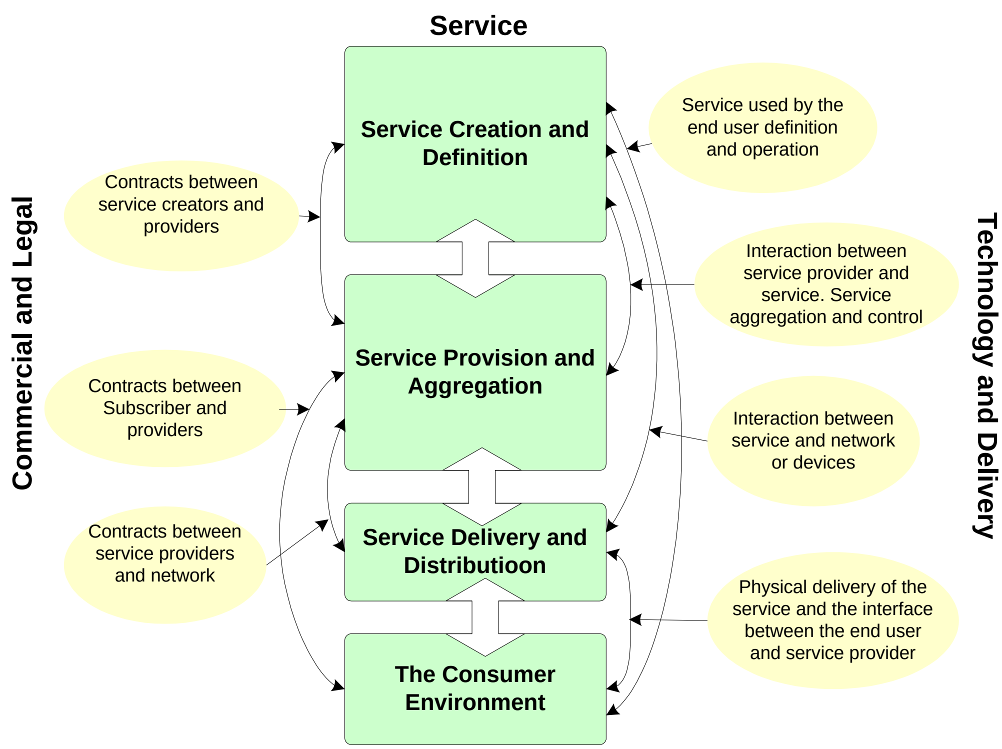

Figure 2.1 Service Supply Chain Model

At such, the service here is the gathering of those service applications that are defined in the service provision by a service provider to home users. Figure 2.1 shows the interrelationships and possible information flow within the service supply chain. Home users have a financial relationship with a service provider and service delivery provider, although sometimes the service provision and delivery come from the same company. Technically, to deliver this service, it is necessary to clarify the requirement specifications and description of properties of any device, user interface, protocol and network involved in the service supply. One of the objectives of this research study is how to complete the service supply chain with quality.

2.1.4 Service Delivery Platform and Home Service Gateway

“A platform can be a physical construction or a set of ideas, principles, agreements, and rules that provide the basis or the outline for something that is more fully developed at a later stage.” (Nokia, 2005). As with service, there is no standard definition, or the components that constitute a service delivery platform (SDP). The definitions of service delivery platform adopted by the proposed research are as follows:

- A SDP provides a system to facilitate the service provision, deployment, management, and execution in connected home environments.
- A SDP supports the delivery of services and contents in a way that is both network and device-independent.
- A SDP enables “Resources sharing” between services.
- A SDP aggregates different service resources and allows service providers to access them in a uniform and standardized way.

In a connected home environment, the SDP is usually realised as a home (or residential) service gateway. Home service gateway is a software/hardware middleware that connects both wide-area network and home network with automatic management and data security, and is a bridge that supports interoperability between home resources and remote service applications in an integrated and uniform manner (Royon, 2007). The home gateway also provides an execution environment that will allow service providers to manage and consume the connected home resources such as devices and information, to deliver services remotely (OSGi, 2008).

2.1.5 Service Delivery Environment for Connected Homes

In principal, the service delivery environment for connected homes comprised of three distinct layers:

- *Home Environment*

The *connected home environment* consists of various home entities, e.g. devices, householder and environment information, which can be directly or indirectly manipulated and communicated via electronic means between them and outside world. For example, room temperature sensor would able to communicate with heating service applications and/or householders. It also provides an opportunity for the householders to extend their monitoring and control beyond the home environment.

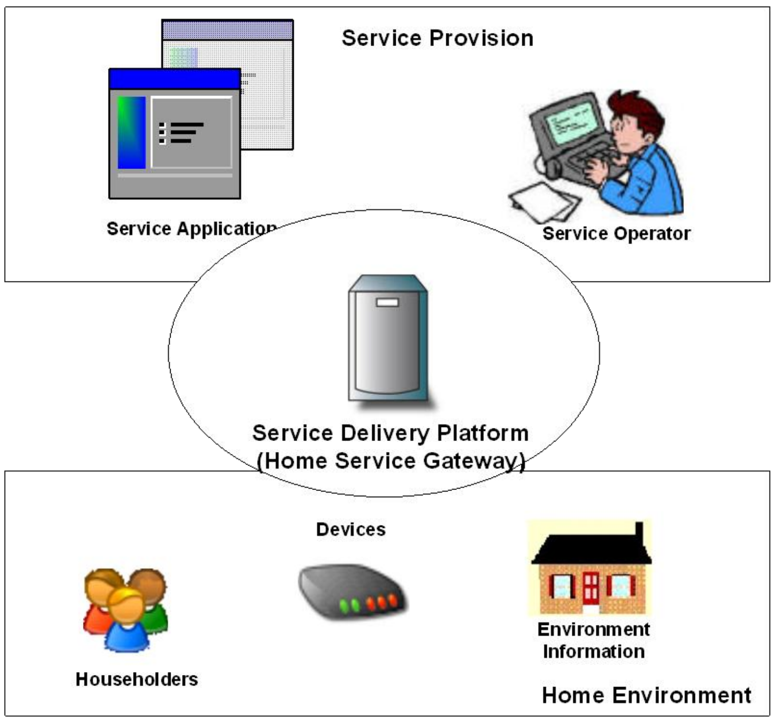

Figure 2.2 Service Delivery Environment for Connected Homes

- *Home Service Gateway*

Within the context of services defined in Section 2.1.1, the home service gateway is not only a home networking device, used as a gateway to connect devices in the home to the Internet or other wide-area network (en.wikipedia.org/wiki/Home_gateway), but also an integrated platform that consists of a set of housekeeping services through which designated service resources within the connected home would allow to manage, and interact with remote service applications automatically for supporting service delivery. Ideally, the home service gateway should realise as an open platform that based on open standards to allow interoperability between service resources in the connected homes and service applications from different service providers.

- *Service Provision*

  The *Service Provision* layer contains the service provision facility and staff to fulfil the service level agreements, provide and delivery and agreed services to individual home environment.

  The following sections will review the existing technologies and potential knowledge gaps for service delivery to a connected home environment.

2.2 Home Appliances, Devices and Environment Information

A connected home is an environment where real world information is captured into a digital or a virtual world, a place where information content is received, consumed, stored, moved between devices and, increasingly, to other networks with which can provide features and services that are not otherwise possible (Laerhoven, 2005). A connected home is likely to contain many different types of devices. However, each device is required to be networked and interoperable with other devices and networks.

In the connected home, sensors are the most influential components that are capable of detecting and responding to the physical stimuli such as temperature, movement, pressure or light. The types of sensors typically used at home include motion sensors, glass break detectors, carbon monoxide detectors, smoke sensors, and temperature and humidity sensors.

2.2.1 Networked Sensors and Devices

Traditional sensors primarily require users to closely interact in their processing, interpret the information, and take action based on what was sensed (Laerhoven, 2005). As ordinary home users are mostly only interested with the service facility, but do not want the technology to complicate their daily life, it requires devices and applications in a connected home to be able to interpret sensor data and act autonomously without human intervention, making the system unnoticeable and in the background. The existing solution is to connect sensors and devices into a common network (e.g. IEEE 802.15 (www.ieee802.org/15/), Bluetooth (www.bluetooth.com), and ZigBee (www.zigbee.org)) that provide standards for data communication and support life cycle management (e.g. addition and removal) of sensors and devices. Each network standard has a mechanism to manage and integrate underline sensors and devices, but the exchange of data between different network protocols requires special support from the home gateway.

The market for sensors has grown up to $43 billion at 2008. (www.Marketresearch.com) With the continuous demands for devices and systems with better capabilities, enhanced functionality and easy to use, sensors expect to bring more value into the systems through additional and more accurate data for operational support. Smart sensors and smart sensor networks are emerging to provide better information and knowledge, which can significantly improve the overall system performance and system reliability (Zhang, 2004).

There is no common agreement about what additional features a primary sensor has to exhibit to become smart. IEEE defines a smart sensor as a sensor "that provides functions beyond those necessary for generating a correct representation of a sensed or controlled quantity. This function typically simplifies the integration of the transducer into applications in a networked environment" (IEEE1451, 2000). The key feature of a smart sensor is the combination of the traditional sensing functionality together with information processing and communication capability. Add-on microchips with communication capability provide the possibility to programme within the sensor for advanced performance.

IEEE 1451 is a family of smart sensor standards that defines architecture and a set of interfaces for connecting traditional sensing elements to microprocessors and allowing sensors to be connected into control networks in a network-independent fashion. Whilst the

IEEE 1451 protocols are powerful as a general interfacing mechanism, they are too complicated and also too expensive to adopt in home environments. A mechanism or new approach is required to support the remote service delivery to connected homes.

2.2.2 Sensors and Devices Management Scheme

Although smart sensors can improve both sensors & devices functionality and flexibility, they require significant amount of effort in programming during the application development stage which cause significant burden to the developers. To alleviate this situation, various sensor management schemes are introduced to ease the development efforts and promote interoperability between sensors from different manufacturers and developers.

As sensor technologies evolve and become diverse, there is no simple solution for sensor design and control. Here are some representative device management schemes:

- Linux Device Driver System

The Linux device driver system is very powerful and widely used scheme. It consists of four features. The first is the data abstraction, which categorizes all devices into different types, like character device (data container accessed by byte units like keyboard or mouse), block device (accessed by block units like hardware or CD), and stream device (transmitted in the form of stream of bytes without buffering). The second is the user Application Programming Interfaces (APIs) related to each data abstraction. The third is an integrated management scheme for each device so that the device manufacturer can easily attach the device driver into operating system dynamically. The last is naming devices with which users can give a name to each device, such as “/dev/rr01” to associate the file system to connect with them (Rubini, 1998). The main advantage of Linux model is that the Independent Hardware Vendors (IHVs) or developers can get the source code for devices accepted into the mainline kernel. Once a device is working on a given version of Linux, interested developers will support it continuously work in all future versions and cross-architectures. However, this scheme is very hard to apply to smart sensors directly, since it requires a sophisticated operating system (Kohn, 2008).

- Sensor Network Operating System

In general, a sensor network operating system includes task management, power management, RF communication, event management, and timing synchronization. Among these systems, TinyOS provides component-based programming paradigm with low power consumption (Levis, 2004). However, it does not contain programmable process model, therefore do not support new application development on top of it. Han et al. introduced a Sensor Operation System (SOS) that features dynamic module loading (Han, 2005). However, update software become very difficult when the development has completed. MANTIS (Abrach, 2003) and Nano-Qplus (Park, 2006) are based on threads, supporting pre-emptive multithreading like Linux. Unfortunately, lack of dedicated management support causes inconvenient for users and developers. For example, more than a hundred APIs are required to support about 20 sensors. Until now, there is no sensor network operating system can met the efficient and convenient requirements to manage sensor network in connected home environments.

- IEEE 1451

The architecture of IEEE 1451 is depicted in Figure 2.1. Sensor can be connected into network via Network Capable Application Processor (NCAP), which is a programmable microprocessor. One smart sensor could attach more than one sensing element, also known as transducer. The software architecture of IEEE 1451 consists of four modules: smart transducer interface module (TIM), transducer electronic datasheet (TEDS), transducer independent interface (TII), and communication protocol (Woods, 1999). Manufacturers are only responsible for TIM and then system developer can provide corresponding NCAP to link the transducer into the network. Each transducer associated with the corresponding TIM and guaranteed its transparency via TEDS. TEDS is a pre-defined data structure used to describe the transducer to the network capable application processor (NCAP), and make the self-identification of the transducer to the system.

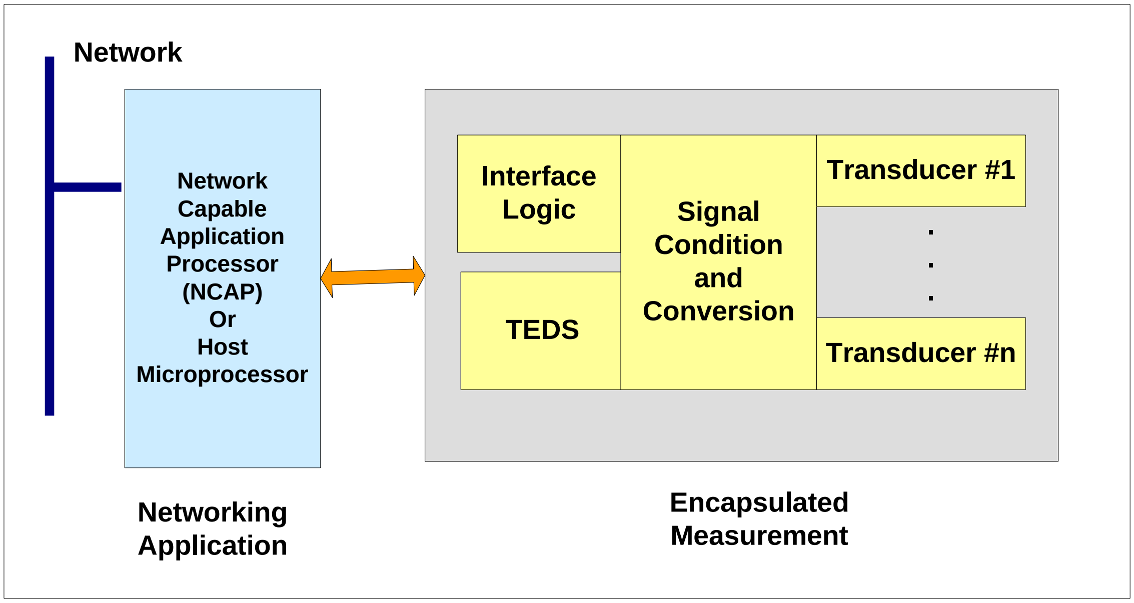

Figure 2.3 IEEE1451 Architecture

Compard with previous technologies, IEEE 1451 standards are more mature and practical. However, the existing standards are too complicated and none of them are specified for connected home environments. For example, IEEE 1451 Sensor Web Enablement Standards refers to Web-accessible sensor networks that can be discovered and accessed through Internet (Percivil, 2006). It is not house owners' interest to expose private information in such public manner; also the cost and technical complexity of these standards are beyond for the majority of householders.

2.2.3 Networked Appliances

In early 1998, Mechatronics Research Group (MRG) at De Montfort University had conducted a research project, called “Whitebox” (Xie, 2001) and subsequently led to an EU funded project called “ELIMA” (Yang & Moore, 2009), which successfully demonstrated the benefits of potential services such as maintenance, re-use and re-cycling via networked home appliances and devices.

Although the benefits of networked home appliances are well understood for the appliances manufacturers and researchers, it has little attraction for householders to purchase networked appliances mainly because lack of public awareness on the potential benefits of the advanced services such as energy management and remote maintenance services. Furthermore, the infrastructure support for the remote service delivery to connected homes still in its embryonic stage and “Killer Service” to attract householders are simply not existed.

2.3 Service Management in Connected Home Environments

Connected home environments are expected to equip with networked devices, appliances and systems. The issues of how to manage these devices and systems without manually configuring them and spending time actively looking for services are a big challenge. Due to the unique characteristics of connected home environments such as lack of technical knowledge locally, the householders expect these connected devices and systems to be automatic organised without needing their active attention.

2.3.1 Service Discovery

Broadly, service discovery is framework that would support the development of self-organisation in an ICT based systems. Such framework is a collection of protocols using to discover hardware components or software entities, and determining how to invoke or utilize these services, is essential for device automatic management (Chakraborty,2004). Over the past decade, many different service delivery protocols were proposed in research communities and industry consortia, each of them has address a different mix of issues with its advantages and disadvantages (Zhu, 2005). It is unlikely that any of these protocols could or would dominate the connected home environments. Furthermore, the existing service discovery protocols are only able to discovery each other if clients and services are using the same protocol. How to support the interaction between different protocols remains as a challenge.

This section is going to review the existing service delivery protocols and their major components regardless their difference in taxonomy. In order to facilitate the discovery process, service discovery protocols are required to support the following functions:

- Advertise when new services come to the network to make them available to consumers. The advertisement is also required to cover their state change and expiration of lifetime.
- Automatic discovery of local and remote service regardless of its network type.
- Services are able to describe their capabilities as well as query and understand other services.
- Automatic configuration without other party's intervention.

2.3.1.1 Service discovery architecture

Among the existing service discovery mechanisms, Jini (www.jini.org), Service Location protocol (SLP) (Guttman 1999), Salutation Protocol (http://salutation.org) and UPnP/SSDP (www.upnp.org) are most popular implementation for fixed network structure, which do not fit very well for dynamically connected home environments.

- Partially centralized architecture

Partially centralized architectures are realized by implementing a set of service coordinators to cache descriptions of related services. Within this architecture, service providers will have to find a service coordinator before register its services with, and service requestor will have to find a service coordinator in order to search for a service. Examples of discovery protocols include UPnP (www.upnp.org), JXTA (jxta.dev.java.net), and SLP (Guttman, 1999).

Although the service coordinators approach does provide service visibility, the benefit dose not come without cost and complexity. For example, service providers need to broadcast a discovery message periodically each time to find a service coordinator and register with it to maintain the accuracy of their repositories. Improvements are required to reduce the management overhead and its complexity (Engelstad, 2003).

- Fully distributed architecture

An alternative to the partially centralized approach is the fully distributed architecture, where no centralized coordinator is necessary. Each service device that wants to provide service to others runs itself as a service discovery server and responds to service discovery requests for its own services. Jini, and SLP are the examples of this type of architecture.

The fully distributed mechanism can be easily realized by running a name resolver on each service device. Since the responding name server is located on the same device that provides the service, a direct route is already in place for subsequent communication between the two devices. However, how to manage distributed object without centralized coordinators is a continual challenge in distributed systems.

- Hybrid architecture

Hybrid approach allows fully distributed service discovery servers co-exist with some centralized coordinators. Service providers and service requestors will utilize a centralized coordinator if anyone is available in their surroundings. Otherwise, they will use fully distributed approach. Twine (Louati, 2005) supports this type of approach. This hybrid approach is mainly adopted by sophisticated wireless systems because of their unique operating environments.

2.3.1.2 Service and attribute name resolution

In connected home environments, various networked devices that offer different services may enter and leave the network at anytime. Efficient and timely service discovery is a pre-requisite for good utilization of distributed service resources in the connected home.

Although there is no particular service discovery protocol that is likely to be supported by majority, it is in sharp contrast to name resolution, where but nearly every Internet host support DNS (Mockapetris, 1987) and is equipped with a DNS Resolver. It is because those most current user-applications include web browsing, e-mail, telnet and ftp rely on DNS for communication (Engelstad, 2003). Due to the success of DNS, DNS SRV records (Gulbrandsen, 2000) have been standardized to bundle simple service discovery into DNS. They allow a DNS resolver to resolve a generic service name (e.g. printer) into an IP address or a port number. By using the existing DNS architecture, simple service can be discovered without considerably add-on complexity. The latest development on service discovery protocols supports the integration of service discovery and name solution.

To provide the interoperability among service discovery protocols and expendability for new services, many discovery protocols define a format for services and attributes, which called template-based approach. For example, Apple's Rendezvous (Zaliva, 2004) gives detail for how service names and attributes should be composed. In addition, several protocols offer pre-defined names and attributes for common services. For instance, Bluetooth SDP (www.bluetooth.com) has pre-defined a set of 128-bit Universal Unique Identifiers (UUID) for frequently used services and a set of attributes IDs. This approach can eliminate the ambiguity during naming process. More over, some protocols, like Jini and Bluetooth SDP offer user-friendly names and attributes instead of conventional machine friendly versions (Zhu, 2005).

However, the real time connected home environments may involve more than one discovery protocol, how to let services using various templates and pre-defined names remain as a challenge.

2.3.1.3 Communication Initialisation Method

When a new service joins the network, it has to initiate communication and then register itself with a directory if present.

Unicast is the most efficient initiate communication method as it involves only related parties, INS, Twine, SLP, and P2P adopt this method for their implementation. Its drawback is it requires the directory resides in a known location. This method will not working well in the connected home environments because of its dynamism.

User Datagram Protocol multicast is an elegant solution among the current service discovery protocols. Under this scheme, newly connected service send out a few multicast address initially, after receive the reply, the received the address will be saved and switch the communication to unicast. This method adopted by Jini, SLP, and Splendor for their implementation.

An alternative method is link layer broadcast, which usually limited within a single hop of wired and wireless network. This method is also adopted by SLP and Splendor for their implementation.

2.3.1.4 Service directory architecture

Current service discovery protocols designed are either based on directory or non-directory model.

The directory-based model has a dedicated component called directory that manage service information and process query. The directory architecture adopted by different protocols can be generally classified as centralized or decentralized model (Ahmed, 2005). Centralized model is not suit for large system as it is fault sensitive. Decentralized architecture is scalable and provides better level of fault-tolerant.

The non-directory based model is suitable for simple environment where services are relatively few. Some protocols like SLP and Salutation provide more flexible solution by supporting both directory-based and non-directory based approaches.

2.3.1.5 Service state information

Most service discovery protocols maintain service statues as a “soft” state, which use a service announcement to specify the service’s life span. Before the service expires, a client or directory can poll the service for validity or the service can announce itself again to renew the state. Otherwise, the service will become invalidate after expiration and removed from the directory. This mechanism simplified the life cycle management of the registered services and can keep service state information up-to-date. Alternatively, clients and directories can maintain the service statues as a “hard” state in which services are polled periodically to make sure their information is up-to-date.

2.4 Service Aggregation Technologies

The key difference between a smart home and a traditional home is its capability to integrate appropriate resources into service applications, and ability to communicate with service providers remotely. Majority of current remote service business models rely on remote service providers to employ and control every aspect of remote service development and its delivery. Thus, it is difficult for small business companies that do not have a remote service delivery infrastructure to provide remote services to the user because of the horrendous overhead cost of building and maintaining such infrastructure. In order to open up the competitive and encourage the development of innovative services; provide both service providers and recipients with more choice and to rapid introduce new services, an open service aggregation platform (or common gateway) is expected on which services from various service providers can be created and delivered (Moyer, 2009).

Numerous research and development effort have been carried out to develop a service aggregation platform. Some research projects are focus on commercial environments, such as smart conference rooms (Potamianos & Huang, 2008) and smart offices (Wang & Chen, 2009) that based on the concept of "Smart Space" with certain degree of success. Although the targeted environments of these research projects are not same as connected home environments, their requirements on service aggregation have high degree of similarity. The review in this section is going to cover the use of service aggregation technologies used at other smart spaces.

2.4.1 Service Aggregation in Context-aware Systems

Context aware computing was first discussed in 1994 as software that "adapts according to its location of use, the collection of nearby people and objects, as well as changes to those objects over time." (Schilit &Theimer, 1994). However, the exact definition of context aware computing is dependent on its building environment and specific requirements of the system. In general, context aware computing is regarded as a system that uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task (Kiani, 2005).

As the interaction between computer-equipped devices and human beings is one of the key elements of a smart space, many smart space systems are also called as context-aware systems. Among them, Anind Dey et al (Dey, 2001) have built a smart conference room, which used a Context Toolkit middleware to support rapid prototyping of certain types of context-aware applications by providing a number of reusable components. The aggregation process of the Java based Toolkit is shown in Figure 2.4 (Dey, 2000).

Widgets are used to acquire context from sensors and provide a uniform interface to applications that want to use the context. Aggregators are similar in functionality to widgets but different in scope. A widget represents the entire context from a particular sensor whereas an aggregator represents the entire context about a particular entity such as person, place or object. An aggregator can subscribe to a widget then waiting for the call back for the notification of the changes of the widgets context information, or can query the widget to get the latest context.

The BaseObject is a Java class that provides the basic communication infrastructure needed to communicate with the distributed components in the Context Toolkit. Discover is used to locate context components that are of interest to them, as a form of resource discovery. Service applications get aggregator location from discover and propose a subscription request to interested aggregator. If aggregator approves the request, it will send a response to the application. In this way, several networked devices (sensors in this project) can work together to fulfil an advanced service object with the coordination of the service application.

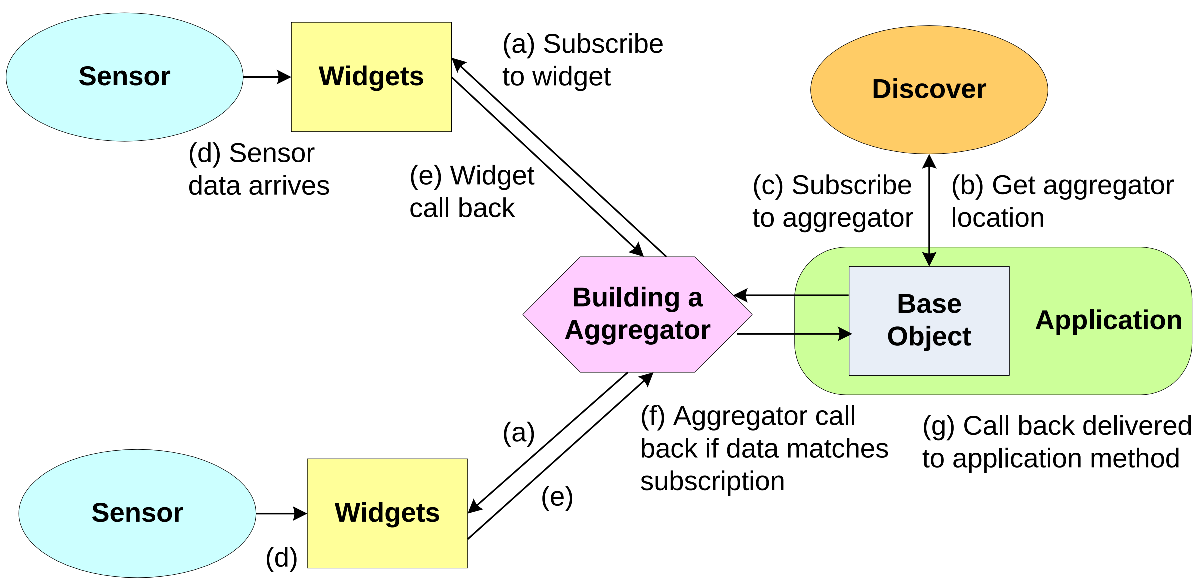

Figure 2.4 Aggregation Process of Context Toolkit Middleware

Even though the Context Toolkit only provides a preliminary test-bed solution for building context-aware system, it identifies design processes that help to gain a better understanding for building context-aware applications. In addition to the complexity and inflexibility of this system, it also requires applications and sensors to be able to understand each other; consequently, application programmers need to use the same terminology as context-sensing architecture. However, such dictionary is not well-defined and varying from system to system.

2.4.2 Centralised Context-aware architecture and Agent technologies

Other context-aware systems like MavHome (Cook, 2003) and EasyMeeting (Chen, 2004) explored a centralized Context Broker Architecture to maintain a shared context model for computing entities in the space and enforce user-defined privacy policies

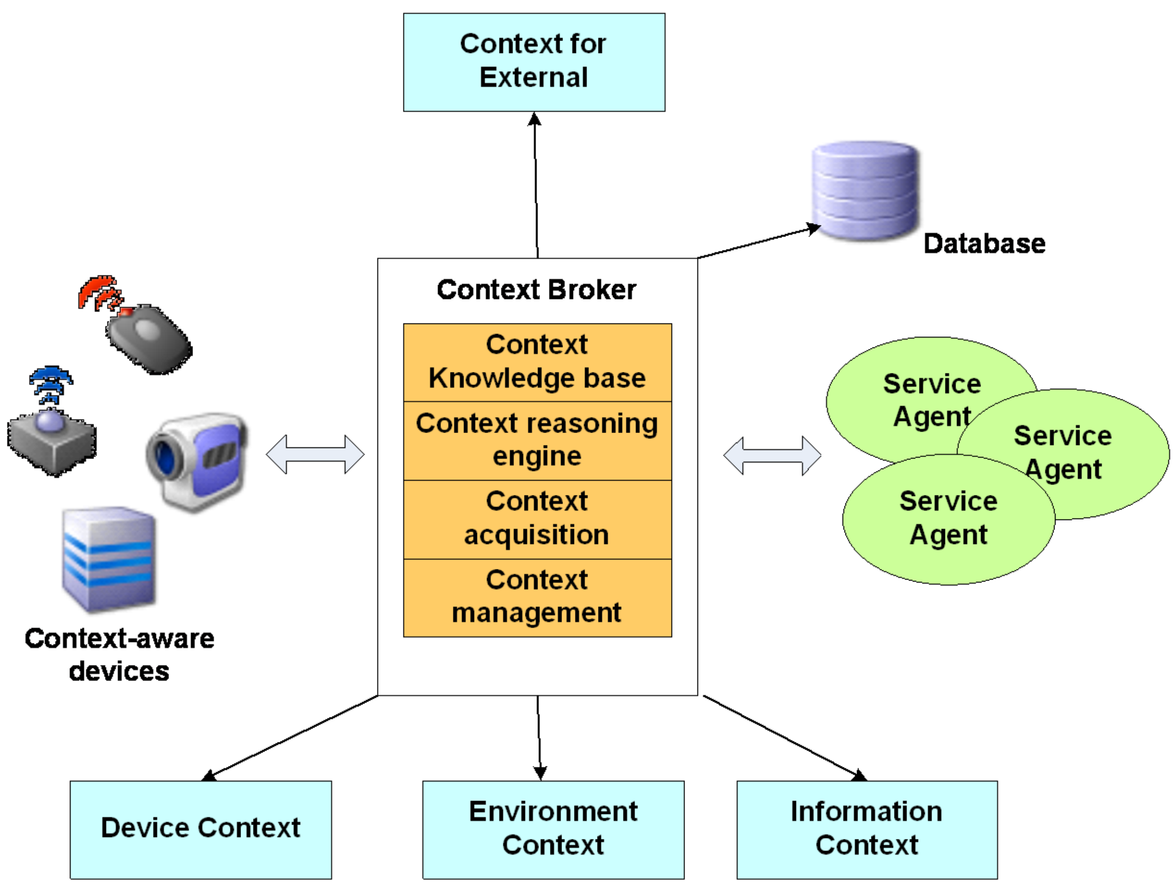

Figure 2.5 Major Components for Context Broker System

As illustrated in Figure 2.5 (Cook, 2003), the central element in context broker architecture is the presence of a context broker - a centralized model of context through which all devices and service in the space can share. The context broker is responsible for:

- Providing a centralized model of context that all devices, services, and agents in the space can share.
- Acquiring contextual information from sources that are unreachable by the resource-limited devices.
- Reasoning about contextual information that can't be directly acquired from the sensors.
- Detecting and resolving inconsistent knowledge stored in the shared context model, and
- Protecting privacy by enforcing policies that users have defined to control the sharing and use of their contextual information.

Centralised context broker architecture provides the possibility for developing more complex context-aware systems with richer context representation. For example, the embedded pressure sensors in the MavHome (Cook, 2003) capture inhabitants' footfalls, and the system uses this data for position tracking and pedestrian recognition. To improve the autonomous and pro-active capability of the system, agent technologies were used for automatic execution of service applications.

First proposed in 1994 (White, 1996), the concept of Agent was implemented as the Telescript, while a runtime environment was also constructed for the execution and migration of Agents. Computers generated and dispatched Agents to execute some management task over the network (White & Helgeson, 1997). The Agent platform consists of the runtime environment and development application program interface (APIs), and many different platforms were designed after 1997 (Baumann, 1998; Johansen, 1998; Braun, 2005)

Agents can have different names under derived concepts and usability (e.g. intelligent agents, mobile agents, autonomous agents, and distributed agents). In principle, an Agent is described as a computer system that is situated in some environment that is able to react independently in order to meet its design objective (Othman, 2007). Agent concept has been used in designing many smart home projects, but most of the today agent systems are very resource demanding both for the client and the server (Wu, 2007; Hsu, 2008; Saravanan & Reuter, 2008).

The difficulties of developing Agent systems can be described as:

- Consider acquired data about the environment to be “knowledge” that is uncertain and subject to revision.
- Use implementations of logical and mathematical algorithms to evaluate or “reason” about data
- Treat the end state of these processes as “desired goals”
- Characterize actions that realize these goals as “plans” intended, by the Agent, to realize those goals (Holmes, 2009).
- How to build and how to integrate in a heterogeneous platform

2.4.3 Service Oriented Architecture Composition and Web Services

2.4.3.1 Service Oriented Architecture

Among the emerging technologies, Service Oriented Architecture (SoA) is often used for building smart homes. SoA is “an application architecture within which all functions are well defined as services with well defined interfaces which can be called in defined sequences to form business processes” (Channabasavaiah, 2003). In SoA , service is defined as “an application function packaged as a reusable component for use in a business process.”

SoA as an architecture is modelled around three basic components: the service requestor, the service provider and the service registry. As shown in Figure 2.6 (Erl, 2004), the service provider describes its services with WSDL and publishes the description at service registry, therefore, services can be located and invocated by the service requestor (Erl, 2004).

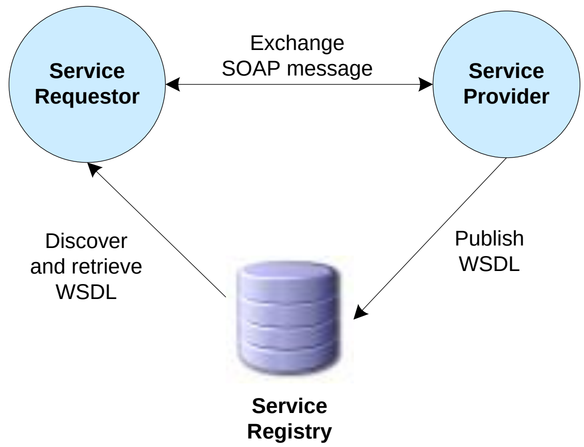

Figure 2.6 Service Oriented Architecture basic components

In essence, services within SoA are self-contained, modular, interoperable, loosely coupled, location-transparent, composite entities that can share contracts with other services in the shape of service level agreements (Erl, 2005). The major benefit of SoA is to facilitate the construction of flexible service applications.

The realization of SoA solution should at least solve the following four challenges:

- How to specify services in a formal and expressive language?
- How to automatically discover and manage services?
- How to automatically composite services into service applications?
- How to ensure their correction?

2.4.3.2 Web Service Technology

In 2000, the W3C received the submission of Simple Object Access Protocol (SOAP) specification that was originally designed to unify and replace Remote Process Call (RPC) communication. The key idea of SOAP was for data transmitted between components to be serialised into XML, transported, and then de-serialised back into its normal format (Erl, 2004).

SOAP has quickly showed its potential in building a non-proprietary free Internet communication platform for advancing e-Business systems. This led to the idea of creating pure web-based distributed technology, called Web services. A web service is a software system designed to support interoperable machine-to-machine interaction over different network platforms, managed by different organisations (Booth, 2003). Two fundamental requirements for this type of service are: communicates via Internet protocols, and data presented in XML format. Web services are building blocks for creating open distributed systems, and allow companies and individuals to efficiently and effectively make their digital resources available worldwide. Moreover, it provides a standard format for interoperating between software applications running on heterogeneous platforms.

The broad acceptance of Web service approach has benefited from a set of supplementary technologies that have subsequently become de facto standards. However, Web services and XML are only building blocks for constructing distributed Web-based systems. Importantly, adopting XML and Web service technology in the correct way requires not only IT skills but also knowledge and understanding of business logic involved. Moreover, Web Service is an integration middleware but not a platform middleware. It won't replace existing middleware, like CORBA, J2EE, and .NET (Baker, 2003).

2.4.3.3 SoA Service Composition Approach

Service composition is an essential component of SoA , as it is concerned with aggregating interoperable services to fulfil a collaborating service task. In recent years, many individual service composition approaches have been proposed and developed. (Pessoa, 2008) These approaches range from industrial standards, such as BPEL (IBM & SAP, 2005) and OWL-S (Ankolekar, 2002), to formal methods, like Petri nets (Hamadi, 2003) and process algebras (Salaun, 2004; Beek, 2007).

Several organizations are developing languages for service composition, the most widely accepted one being the Business Process Execution Language (BPEL) and the Web services Choreography Description Language WS-CDL (Kavantzas, 2004). However, many of these languages have limited ability to support automatic service composition, mainly due to the absence of ontology representation of the available services. Even though Semantic Web community and others have proposed several solutions, such as the Web Ontology Language for Web Services OWL-3 (Ankolekar, 2002) and the Web Service Modelling Ontology WSMO (Dumitru, 2005), there is no well-accepted ontology dictionary for the connected homes (http://sioc-project.org) (http://www.foaf-project.org), which means a semantic solution is still fairly remote for a smart home service aggregation approach.

Life-cycle Phase	Evaluation Criterion
Service Discovery	Service description
	Service Matching and selection
Process model creation	Behaviour specification
	Information specification
	Level of automation
	Composition time
	Coordination distribution
Process model verification	Composition correctness
Execution	Service binding

Table 2.1 Service Composition Life-cycle Phase Evaluation Criterion

As shown in Table 2.1, a service composition process can be briefly categorized into service discovery, process model creation, process model verification, and execution phases. For each considered phases, a set of criterions can be used to evaluate different service composition approaches (Pessoa, 2008).

2.4.3.4 BPEL and BPEL Engines

First released in 2002, BPEL is a XML based language designed to enable the coordination and composition of a set of services (Barreto, 2007). It is based on Web Service Description Language WSDL (Christensen, 2001), which is basically an interface description language for web service providers. BPEL is a behavioral extension of WSDL using a workflow-based approach. It expresses relationships between multiple invocations by controlling data flow link and employing a concurrent computation model (Beek, 2007). The main construct to model the flow of services is called process, which is a concurrent description that connects activities for sending/receiving message with external service providers. BPEL is the most important existing service coordinating technology in the business-to-business (B-to-B) area.

However, BPEL is only able to coordinate for the semi-permanent web services system. It is unsuitable for binding unknown web services according to the service description(Yamato, 2008). BPEL requires a rigid interface description with port type names and operation name of web services, so it can only be used to apply pre-known web services whose port type and operations exactly match.

After BPEL and WSDL are deployed, a BPEL engine is required to interpret BPEL description and execute activities such as the service invoking or sending message. There are many BPEL engines available, both form commercial or open source. Table 2.2 shown some well-known BPEL engines.

Product	Vendor	Framework	Compatibility	License
Apache Agila	ASF	Servlet	WS-BPEL	Apache v2.0
Apache ODE	ASF	Apache Axis J2EE	BPEL4WS 1.1 WS-BPEL	Apache
BizTalk Server	Microsoft	.NET	BPEL, BPMN, RFID	Commercial
Oracle BPM	Oracle	J2EE	BPEL	Commercial
Open ESB	Sun	J2EE, JBI	WS-BPEL	Open Source

Table 2.2 Comparison of BPEL Engines

Among them, all commercial BPEL engines are specifically designed for enterprise business process orchestration (Riou, 2008). The three major open source BPEL-based development are Sun's NetBeans Enterprise Pack (http://www.netbeans.org/products/enterprise), The Eclipse SOA Tools Platform Project (http://www.eclipse.org/stp), and JBoss jBPM (http://jboss.com/products/jbpm). Although the Sun tools is the most mature and easy to use, but to run NetBean consumes quite a bit computer resource.

Nowadays, BPEL engines are predominantly deployed in centralised architecture, which requires all interactions to go through one central server. This approach is mainly developed for enterprise systems, which could be too complicated and expensive for the connected home user. Alternatively, some research projects provide a decentralised process management architecture, which allows distributed BPEL engines to manage the execution of relevant sub-processes, and to interact with the underlying service resources directly. This architecture naturally reflects the distributed and dynamic features of the connected home environments and should be able to offer better quality of coordination support (Yang & Pidgeon, 2007).

2.5 Summary

Services in a connected home environment have benefitted from the boom in Internet connections and advancement of networked devices, appliances and systems. With the support of a service delivery platform, householders can receive services provided by remote service providers based on the service resources available in connected homes.

The potential services in a connected home could be categorized into environment, security and safety, healthcare, domestic appliances, and home entertainment. Of these, home entertainment systems require large investment in components from specialised network cables to high end networked devices. Therefore the proposed research will not focus on the delivery of entertainment services although their market potential is huge. Alternatively, it aims to provide a thin and low cost remote service delivery platform through which it is possible to facilitate the development and delivery of remote services in an integrated and flexible manner.

The lack of attractive services has so far impeded the prevalence of networked appliances. In fact, sensors are the most influential components for the connected home. However, traditional sensors need add-on microchips with data processing and communication capabilities to provide advanced functions and connectivity. A smart sensor is expected to provide interfaces that conformed to the open standards, which enable them to communicate and offer their service to other devices, appliances and systems.

The nature of the connected home environment expects these networked devices, appliances and systems to be able to organise automatically without the active attention from the home users. Therefore, the concept of service discovery mechanisms is vital in managing device discovery, installation, configuration, and life-cycle control. Currently, there is no universal acceptable service discovery protocol for all types of networks, which brings complexity to connected homes because heterogeneous networks are likely exist in the majority of connected home environments. An integrated platform is expected to provide support for connected devices, appliances and systems using different protocols to interact with each other.

A valuable aspect of the connected home is its capability to integrate appropriate service resources into service applications and communicate with remote service providers. This capability is vital in advanced services because they are likely to require more than one service resources to fulfil the service task. Majority of current business models rely on service providers to deploy and control every step of service development and delivery. This model usually employs proprietary standards and designs, which restricts other service providers in sharing the same infrastructure and consequently limits the choice of the householders and prolongs the introduction of new services. An open service aggregation platform is required to support remote service delivery that would provide supports for “Mass Customisation” in services and provide freedom of “Choice” to both service providers and recipients.

Chapter 3 Existing Service Delivery Systems and Related Works

3.1 Introduction

Enabling technologies in the area of networked devices, communication networks, and software architectures have been reviewed in Chapter 2. Although there is no platform existing that would support remote services delivery with the capability of “Mass Customization” and freedom of “Choice” for both services providers and householders, many existing platform hint of a partial solution.

This chapter establishes a set of fundamental challenges for building a remote services delivery platform that would support “Mass Customisation” and freedom of “Choice”. Current representative architectures that are adopted by leading service providers to facilitate remote services delivery will be reviewed. Although these architecture may not conceived with connected homes in their mind, their designs and approaches provide useful references for establishing the required architecture. State of the art research projects related to remote services delivery to the connected home environments will also be examined.

3.2 Technical Challenges for Remote Services Delivery to Connected Homes

A major technical problem hinders the realisation of remote services delivery platform that would support “Mass Customisation” and provide freedom of “Choice” to both services providers and householders is to achieve seamless integration and interaction among heterogeneous devices, appliances and systems within connected home environments. This problem leads to numbers of challenges, including both technical and socio-economic perspectives. This research project is mainly focus on technical perspectives and their main technical challenges are described in the following sub-sections.

3.2.1 Service Resource Visualisation

Remote service providers must have the knowledge and information on available service resources in connected homes before able to assess and decide whether the required service(s) can be delivered with acceptable quality. The current practice to acquire this knowledge and information is either conducting on-site survey or through some form of questionnaire, however, none of these practices would able to support the “On-Demand” business model in remote service delivery effectively and efficiently. To overcome this problem, it is desirable to have a method and/or tool to enable the remote service providers to acquire such knowledge and information remotely and automatically. The key challenge for achieving such desirable outcomes is how to present the required information such as available service resources, their locations and method(s) of invocation electronically that would accept by majority of manufacturers and remote service providers for connected homes. Although some obvious standard-based descriptive language such as Extensible Mark-up Language (XML) did exist to encode such information electronically, currently there is no standard in defining the necessary semantics for the service resources within connected home environments to allow such information exchange between connected home environments and the remote service providers.

3.2.2 Service Resources Discovery and Life Cycle Management

Although remote service providers requiring to have the knowledge and information of available service resources in connected home environments for supporting their service(s) delivery, it would be unacceptable to allow them roaming connected home environments to interrogate all connected home devices, appliances and systems to gain such understanding because of the security concern. An alternative way is to employ a central repository where connected service resources are registered and home gateway is always the preferable choice for such purpose.

The requirements of service resources discovery in the connected home environment is significantly more complex than both commercial and industrial environments as there are three different types of discoveries coexisting in a single space (Zhu, 2005): firstly, an individual device, appliance or system needs to be recognized within a heterogeneous network environment: secondly, a service object needs to register with the home gateway (i.e. central repository) and manage underlying devices and related information resources; and thirdly, a service provider needs to discover service resources over the Internet for supporting the required service(s) delivery. Although there are many service discovery protocols have developed in the past decade, none of them can cope with the complexity of service discovery in connected home environments.

There is also a challenge to carry out life cycle management of connected home service resources automatically because connected home environments are highly dynamic and constantly under change when compared with commercial and industrial environments. It is highly unlikely householders willing to spend significant amount of time and efforts to update and maintain their home central repositories to enable remote service providers to acquire the knowledge and information of available service resources. The reason is partly because majority of householders are lack of required knowledge and partly treats them as an operational burden. In order to overcome such management issue, automatic service resources discovery and management become essential for the success of remote services delivery to connected homes (Lv & Zhou, 2009).

3.2.3 Address and Relation Management

An address mechanism must able to identify and locate all connected devices, appliances and systems to deliver the appropriate messages to rightful recipients. Within a collaborative networked environment, a connected device could be presented as a service object on its won or part of service resources within a service contract that using different naming conventions. For example, a connected device such as occupancy sensor can be identified with its unique manufacturer ID when it presented as a standalone service resource or identified with a "name" that given by the service platform when it assigns as part of the monitored security service contract. Although different identifiers are using to identify the same connected device the service platform must able to manage such complex naming convention without any confusion. The existing solutions such as identifier alias using look-up table might offer an answer but with limitation in the connected home environments (e.g. resources hungry).

Heterogeneous networking is part of the nature of connected home environments. For example, addresses of WiFi Ethernet and ZigBee using different addressing schemes. Thus the addressing mechanism needs to be able to translate network addresses and associated protocols between different network domains. Thus, connected devices, appliances and systems that residing in different networks is able to interact with each other.

One of the features of connected devices, appliances and systems in a connected home environment is that they have physical location limitations. For instance, a sensor placed by the front door may have different service tasks and separate service meaning to one installed in a bed-room. In addition, devices in the same environment could be required to provide their service(s) to different service recipients each having their own preference. However, addressing mechanisms are not responsible to present the location related information. Nowadays, location tracking requires specific devices (i.e. expensive special devices and complicated protocols, which will significantly increase the cost of the connected home (Baek & Choi, 2007).

3.2.4 Message Interaction Mechanism

The service delivered in the connected home not only encounters the interaction between dynamic and heterogeneous devices, appliances and systems, but also involves service processes provided and maintained by different platforms and organisations (Zhang &

Gracanin, 2008). There are several technical challenges in building the connected home message interaction mechanism:

Firstly, most existing systems restrict message subscriptions to pre-defined destinations. However, during on-Demand remote service provision, it is the remote service provider's responsibility to generate the service contract. It means that service resources such as sensors within the connected home environment might not have the knowledge about which party is going to consume its message (i.e. information resource). How to monitor and manage the state of change of underlying connected devices and forward messages to the desired party remains as a challenge.

Secondly, data and messages from connected devices, appliances and systems are largely simple and in various formats. There is no existing remote service delivery platform allows remote service providers to understand and use them automatically in such dynamic and unstructured environments (no two homes are alike). The problem is how to aggregate and consolidate these remote service resources into a meaningful content and use them for supporting the required remote services delivery.

Thirdly, as connected devices, appliances and systems involved in the service provision are located on different platforms, it creates a challenge in service integration and communication regarding workflow and network traffic management within dynamic and unstructured connected home environments.

3.2.5 Service Aggregation

Advanced integrated services such as care services to elderly or disadvantage people are likely to require a collection of different types of care services such as social and health care. These different services are also likely having some form of interdependence in order to maximize their benefits to the care recipients. Thus service aggregation is an essential part of service delivery. The current practices in service aggregation are not only static in nature (e.g. once it forms, any minor change will be a mammoth task), but also an intense "labour task";

are not able to support remote service delivery to connected homes that would provide "Choice" and supporting "Mass Customization" to service recipients. A much more dynamic service aggregation approach with adaptation capability is required to overcome deficiencies of existing practices. Currently, the adaptive service aggregation approach has not been investigated sufficiently and hence it is still an open issue to research communities.

One of the challenges is how to establish a communication service between authorized service providers and the connected home in an autonomous way. In general terms, a communication service is a functionality that is made within local networks to exploit and access service resources (Quitadamo, 2007). Such services have always been implemented by static software and/or hardware modules and managed by dedicated network administrators. This approach is becoming too "primitive" for remote service delivery to connected home environments. Developing a dynamic communication service that could support remote service provision by "On-Demand" aggregation with relevant service resources is a major challenge. Another challenge is that each party has its own security mechanism and policies to protect its local resources, and the service process across multi-platforms must be able to operate within various security realms. "A security realm is a group of principals (e.g. people, devices and computers) that are registered with a specific authority and managed through a consistent set of security processes and policies." (Zhang & Xiu, 2007).

As domestic service resources are no longer expected to be packaged and published in pre-defined service task, it is the remote service providers' responsibility to orchestrate relevant service resources into a required service task. The process of allowing remote service providers to gain the knowledge and information of available service resources functionalities and their relationships to enable them generating a service process accordingly remains as a challenge. Because there are no clear standards for such descriptions, not even mention tools available for supporting the generation of required information automatically. Furthermore, as there is no clear mapping between the service task and available service resources, it raises a challenge on how to aggregate these service resources to create the required aggregated service, which would carry out the required task(s) in the connected home (Kalasapr & Kummar, 2007).

Traditionally, the entities involved in a service process exist in the same organization, so they can rely on the central management system to manage the service binding. However, this approach does not work well in a cross-domain environment - the connected home, because lack of a central management platform today. Although the current Web service platform enables dynamic establishment of service composition and invocation between multiple organizations, it is too costly (though not impossible) to implement onto connected home devices, appliances and systems, which the majority of home environments are typically cost-sensitive (Wassermann & Ludwig, 2009). Managing cross-domain service processes in an efficient and affordable way is a challenge that needs to be addressed.

Challenges also exist in the service process execution. In most protocols, the service process represents a set of high-level behaviours in a standard format. It consists of information such as when and who to send message and receive message, etc. The execution of service process can be a very complicated transaction, including process comply, binding, and instantiation (Shang & Wang, 2008). Special features have to be introduced to the service delivery platform to handle the complexity of real time execution. What makes it even more complicated is that cross-platforms execution is unreliable, having a high probability of failure. An error handling methodology is required that can manage the service execution when abnormal situations arise in the service process execution.

3.3 Enabling Architecture for Service Delivery

3.3.1 Universal Plug and Play (UPnP)

The UPnP Forum (www.upnp.org) was formed by Microsoft in June 1999. It is an industry standard adopting peer-to-peer service-oriented architecture for consumer devices. UPnP is based on well-established standards like HTTP, XML and SOAP and defines an architecture for pervasive peer-to-peer network connectivity of intelligent appliances, wireless devices, and PCs of all form factors. The UPnP system architecture can be displayed in Figure 3.1 (UPnP Forum, 2008).

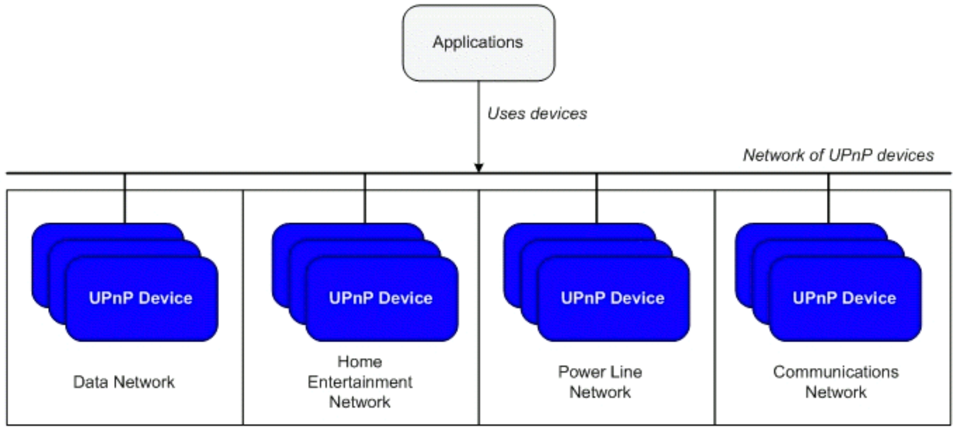

Figure 3.1 UPnP Architecture

The UPnP networking is based on Internet Protocol (IP) addressing scheme. Each device must have a Dynamic Host Configuration Protocol (DHCP) client and search for a DHCP server when the device is first connected to the network. If the DHCP server is available, the device must use the returned address assigned to the system, otherwise, the device must use Auto IP to get an address (UPnP, 2008).

When a device is added to the network, the UPnP discovery protocol allows the device to advertise its services to a control point on the network. Similarly, when a control point is added to the network, the UPnP discovery protocol allows the control point to search for devices of interest on the network. A device describes itself in discovery message and allows the control point to retrieve and learn about its capability.

The device description includes a list of actions the service responds to and a list of variables that model the state of the service at run time. A control point can send a control message to the control URL of the device as a function call; in response, the device service returns any action-specific values. When the state value changes, the service sends out an event message to the subscribed control point. Messages are expressed in XML using the Simple Object Access Protocol (SOAP).

UPnP does not specify any method for remotely accessing the services of home devices. Additional design decision that utilizes web service technology can expand the original scope of UPnP technology by supporting remote access to UPnP Networks. UPnP devices physically attached to the home network can connect to remote services or devices, as if they are connected to the same local network.

The advantages of UPnP can be summarized:

1. A protocol-based solution enables the implementation to become platform and language independent.
2. It is built on top of well-established standards and aligns with business operating environments.
3. The flat architecture is considered simple for construction and management. It is easy to extend and scale up, and avoid central point of failure.
4. Inherent plug-n-play capability simplifies the system installation and management.

The disadvantages of UPnP can be summarized:

1. The flat architecture could cause speed variation in network traffic and the communication could be interrupted due to congestion. There is no way to guarantee how long it will take a message to get from one node to another under all circumstances.
2. Due to its peer-to-peer nature, it is difficult to obtain consistent overall domain states (e.g. node population, absolute time).
3. IP dependence and heavy traffic would hamper its adoption for simple home devices because of resource requirements.

3.3.2 Jini

At the end of 1990s, new devices are increasing enhanced with embedded processors and capable to be interconnected via a network (Venners, 1999). Java started the Jini community to exploit the benefits of this trend, which as Bill Venners pointed out, intended to provide architecture for the emerging network-centric hardware environment. Jini is not an operating system, it based on the idea of federating groups of users and the resources required by those users. It aims to make resources can be used across a network as if they were available locally (Sun, 2005) .

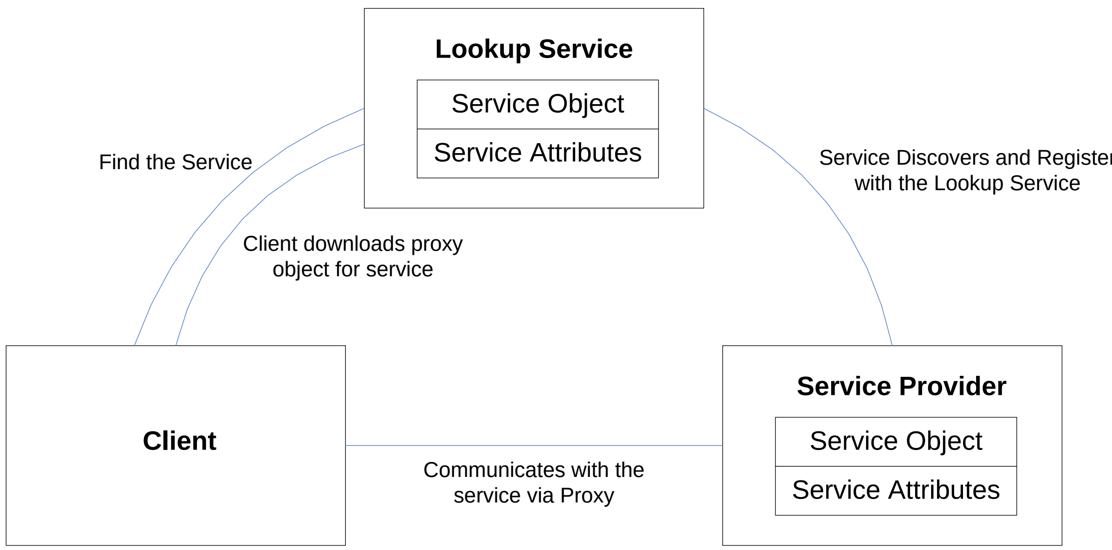

Figure 3.2 Jini Architecture

The Jini system takes advantage of the mobility of Java platform and extends Java applications from a single virtual machine into a network of machines. As shown in Figure 3.2 (Sun, 2005), in the heart of Jini system is the trio of protocols called discovery, join, and lookup. Each node within Jini network must discover a lookup server via preconfigured knowledge or multicast discovery. After discovery, the node publishes all its services to the lookup server for other integration and collaboration. Jini standardises Java interfaces to shield the implementation. Although the Jini specifications do not impose a specific protocol for the communication with the services itself, the implementation requires Remote Method Invocation (RMI), which is the Java implementation of remote function call for contacting the lookup server. In many existing deployments of Jini, RMI has been used for all communication. Thus, Jini is heavily reliant on Java implementation. Furthermore, RMI is currently only available on top of an IP network. For other network technologies, the proxy mechanism is used to convert them to a RMI solution.

Jini provides some significant features in handling dynamic integration and collaboration. Jini leasing mechanism allows leases to be defined on the use of service on different nodes. It automatically copes with dynamic nodes. It is possible to have more than one lookup server within a network. Through synchronizing multiple servers, it provides redundancy and avoids a central point of failure. For integration and collaboration, a lookup server is only used as a medium for discovery. After discovery, nodes can communicate with each other without a server present. Theoretically, the communication between nodes can be any protocol to accommodate network diversity.

The advantages of Jini can be summarised as:

1. Federal architecture provides reliable infrastructure for integration and collaboration.
2. Theoretically, Jini can work with any network protocol, although an IP-based network is the current popular implementation.
3. Extensive mechanism and protocols have been designed to handle the dynamic nature of network communication.

The disadvantages of Jini can be summarised as:

1. Jini is a Java-based solution. It does not support other software languages.
2. The initial communication is based on RMI, which built on top of an IP network, which puts a lot of constraints on the final implementation.
3. There is no clear definition of gateway within architecture.

3.3.3 Open Service Gateway Initiative (OSGi)

OSGi is an independent, non- profit corporation working to define and promote open specifications for the delivery of managed services to networked environments, with focus on homes and automobiles. The OSGi Service Platform is the Java based application server and its concept structure can be illustrated in Figure 3.3 (OSGi, 2009).

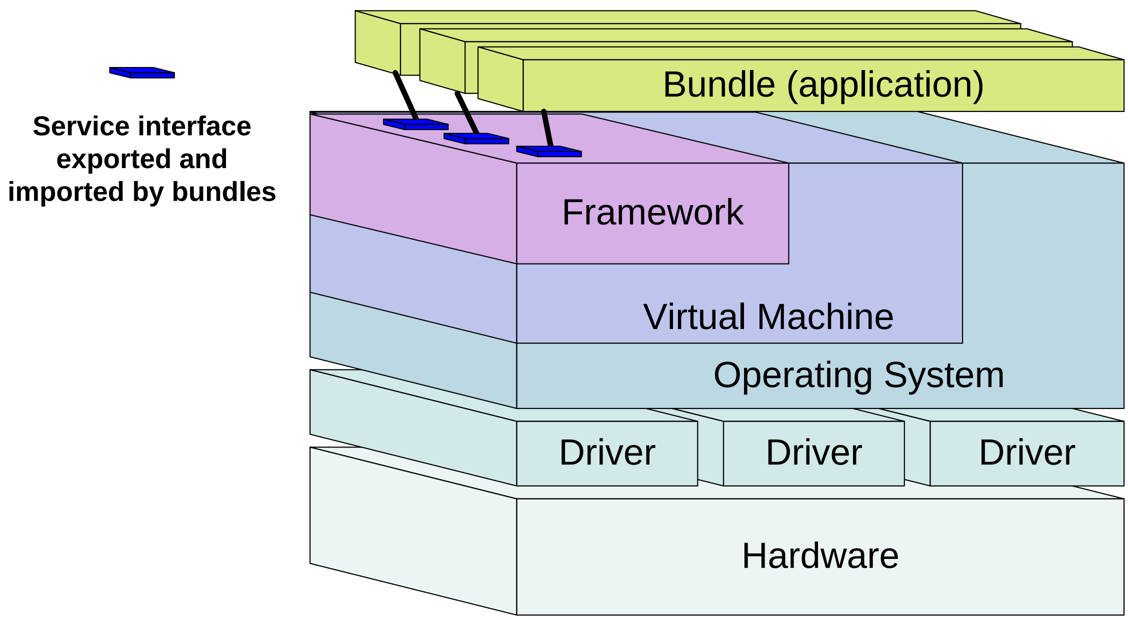

Figure 3.3 OSGi Concept

OSGi specifications provide the bundle format that packages applications in a standard Java Archive (JAR) file (OSGi, 2009). The bundle must be prepared and installed in the OSGi Platform, ready to be executed in a single Java Virtual Machine (JVM). The OSGi Platform is responsible for the download, install, start, update, stop, and uninstall management of bundles.

The main content of bundled JAR files is class files, which are the executable part of the bundle. Classes are grouped into a package, each having a unique name. Every bundle can export a set of classes to and import classes from other bundles. Every JAR contains a Mainfest, which stores information about its contents in headers. The OSGi Platform inspects the header and installs the package as an object in the OSGi Service Platform. The Platform also supports the object registered with the Service Registry, then can search for the matching objects when needed.

Nearly all bundles need to deal with events, either as an event handler receiving and handling event objects, or as an event publisher, sending out event objects. The event interaction in an OSGi Platform is based on publish-subscribe paradigm, which decouples sources from their handlers by placing an event channel between them through the Event Admin Service. The Event Topic defines the type of event and serves as a first-level filter for determining which handlers should receive the event, whereas the information of the actual event is provided as Event Properties.

The OSGi Service Platform supports the automated detection and attachment of devices by downloading and installing device drivers on demand. The platform customizes the selection of the best driver service from a set of suitable driver bundles. The support of web service is the ongoing area of OSGi. However, more and more devices have part of OSGi platform built in which can facilitate the integration and reduce the load of platform.

The advantages of the OSGi platform can be summarised as:

1. The tiered architecture and gateway-oriented platform mean it has a strong appeal for the service delivery system to build upon. Furthermore, it can accommodate any networks as long as there is a Java driver available.
2. The platform has a clearly defined mechanism and associated function to enable automatic setup and configuration.
3. Dynamic deployment capability provides flexibility to handle unforeseen change in requirements and/or operating environment.

The disadvantages of OSGi platform can be summarised as

1. Java-based solution constrains the development choice. Each network protocol and application must be converted into a Java object. It comparatively demands more resources than other platforms.
2. The integration and/or collaboration only occur at the gateway level. Peer-to-peer communication is not supported and management is centralized within OSGi platform. Thus, OSGi is prone to a single point of failure if redundancy design is not included.

3.4 Emerging Research Project Architectures

3.4.1 AIRE, MIT Artificial Intelligence Lab

AIRE (Agent-based Intelligent Reactive Environment) Group at MIT conducted an IRoom project, which is a conference room enhanced with cameras, microphones, and other sensors.

This system is called a reactive behavioural system inside which is a two-step process designed to assist users with their daily tasks: by understand the context of the current user; and to adapt the IRoom's response to that context (Look & Shrobe, 2007).

A middleware called Metaglue and its extension called Hyperglue are developed to assist service management and integration. Metaglue is a Java based platform that allows the creation of intelligent environments controlling software agents. The platform provides an inherited agent class to Java objects, a post compiler to generate new agents from Java compiled class, and a Virtual Machine infrastructure to run these agents.

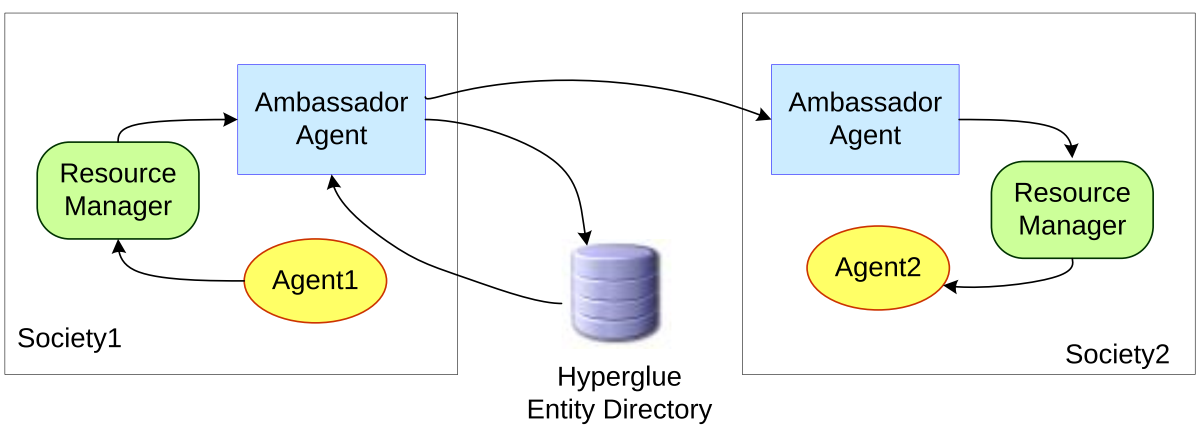

Figure 3.4 Hyperglue System in Operation

As shown in Figure 3.4 (Look, 2005), when an agent in one society need to communicate with another agent, it first will contact local resource manager. The resource manager will then find out if the requested agent is located at a remote society. The Ambassador agent acts as a proxy object for its society to other agent societies, and is registered to a global directory system called Hyperglue Entity Directory (HED) on start-up. When local Ambassador agent receives the communication request, it consults the HED for the location of remote society, then contact the remote society and pass on the request. The remote Ambassador agent can then pass on the request to its local resource manager, which can make the determination about how to fulfil the requirement and pass back the connection detail for any found local agents, and finally respond to the original requester.

It is pioneering research for intelligent environment using both agent technology and Service Oriented Architecture (SoA). However, as the evaluation report stated, the implementation and running of this system is very complicated and not reliable. That is one of the reasons that agent technologies are largely remaining as research topic in computing area.

3.4.2 Ambient Intelligence for the Network Home Environment (Amigo)

Between 2004 and 2008, 15 of leading companies and research institutions of Europe have joined forces in a project to research to develop open, standardized, interoperable middleware and intelligent user services for the connected home environment. (www.amigo-project.org) The Amigo consortium has offered their developed tools as open source software to promote wider take-up. (https://gforge.inria.fr/projects/amigo/).

Building on top of the OSGi framework, (http://forge.ow2.org/projects/oscar/) Amigo supports clients developed both on .NET and Java platforms. As displayed in Figure 3.5 (Amigo, 2005), the Amigo open source software components can be divided into three major parts: the middleware layer contains the basic functionalities that are needed to facilitate the connected home environment, including service discovery and interaction; the intelligent user services bridge between users, multiple sources of information and service providers, and make pattern-based predictions to schedule the service process; the application and services layer provides support for interoperability, security and service description to service developers.

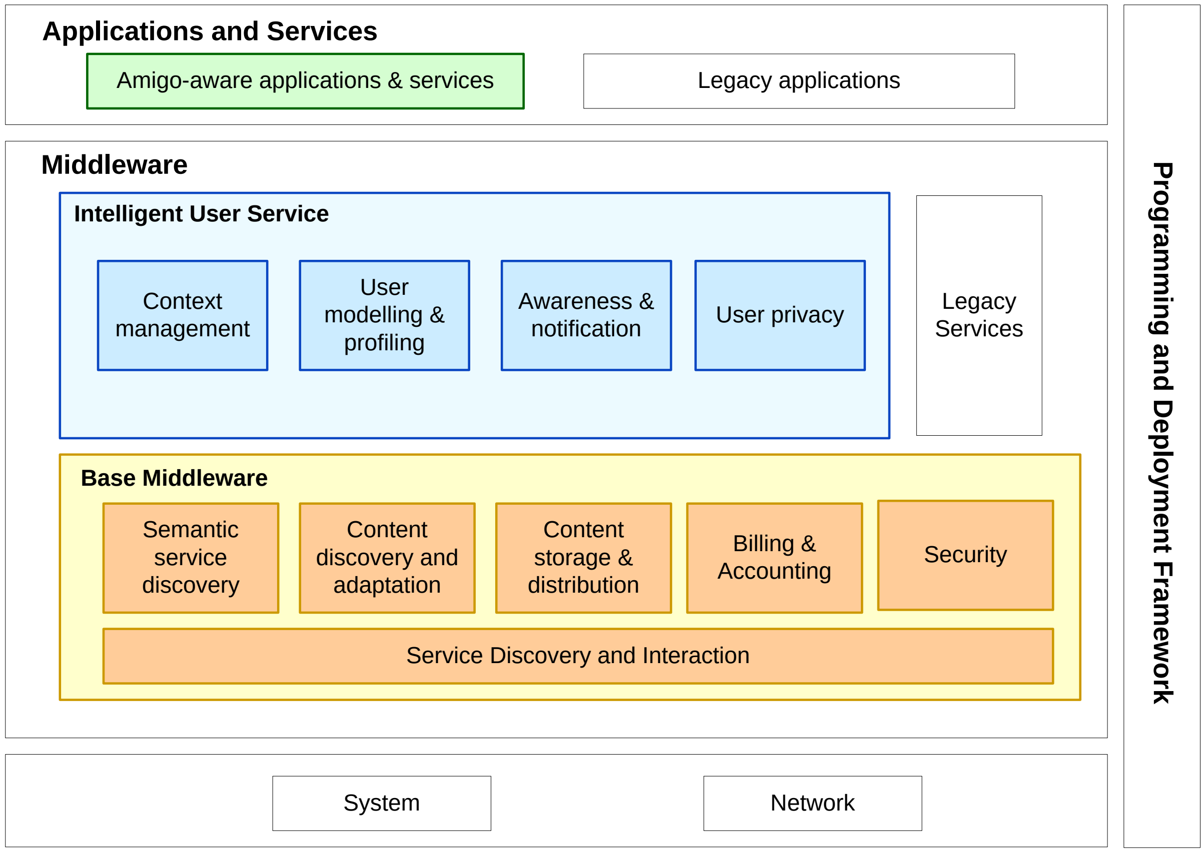

Figure 3.5 Amigo System Architecture

The implementation of Amigo is built upon the OSGi platform although its philosophy and design does not mandate this requirement. The Context Management Service (CMS) broker provides required functionality for discovering and using context sources (Leutnant, 2007). The CMS is responsible for the interface and the registration process inputting the context source automatically. The process of exchanging information can be split up into three steps:

1. The context client which is interested with getting information must make itself known by CMS by calling its *subscribe* method.
2. The context source posts information to CMS by two ways: “query driven” or “data driven”. The “query driven” approach is provided by the method *query*, which only replies to the specific query of the context client. The “data driven” approach is provided by the method *notify*, which pushes data to context client each time its sensor information changes.

3. The context client which is no longer interested in getting information needs to unsubscribe from the CMS by calling its *unsubscribe* method.

The user modelling and profiling service (UMPS) provides the methodology to enhance the usability of services and interfaces by packaging them together into a service application. The service application is responsible for tailored information presentation to user; reason about user's future behaviour; finding relevant information; and adapting interface and information features to the user and context client in which they are used.

3.4.3 The Application Home Initiative (TAHI) Open Architecture

Launched in 2003, the TAHI Open Architecture (TOA) is a conceptual specification that aims to provide an architecture of architectures to achieve interoperability and openness in service delivery and operation between the Service Providers and the residential premise. The essential concepts are made up by a Service Supply Chain Model, Remote Service Objects (RSOs) and Pervasive Service Agents (PSAs). The TOA employs object and agent technologies to realise the service supply chain and fulfil the operational requirements raised by the model. The overview architecture of TOA is represented in Figure 3.6.

The main building block of the TOA is the Remote Service Object (RSO). This represents the diverse service resources required to execute a service application or provide a service. As displayed in Figure 3.6, RSO is a one to one direct mapping of information of service objects at remote server, where the service or application may be modelled. RSO is intended to implement in remote server or home gateway but not on end appliances, as otherwise the possible change of the appliances may break the mapping.

The Pervasive Service Agent (PSA) is also another essential building block of the TOA. PSA interacts with RSOs to discover if they are available; able to supply the service and deliver the service(s) that meet the quality of agreed service(s).

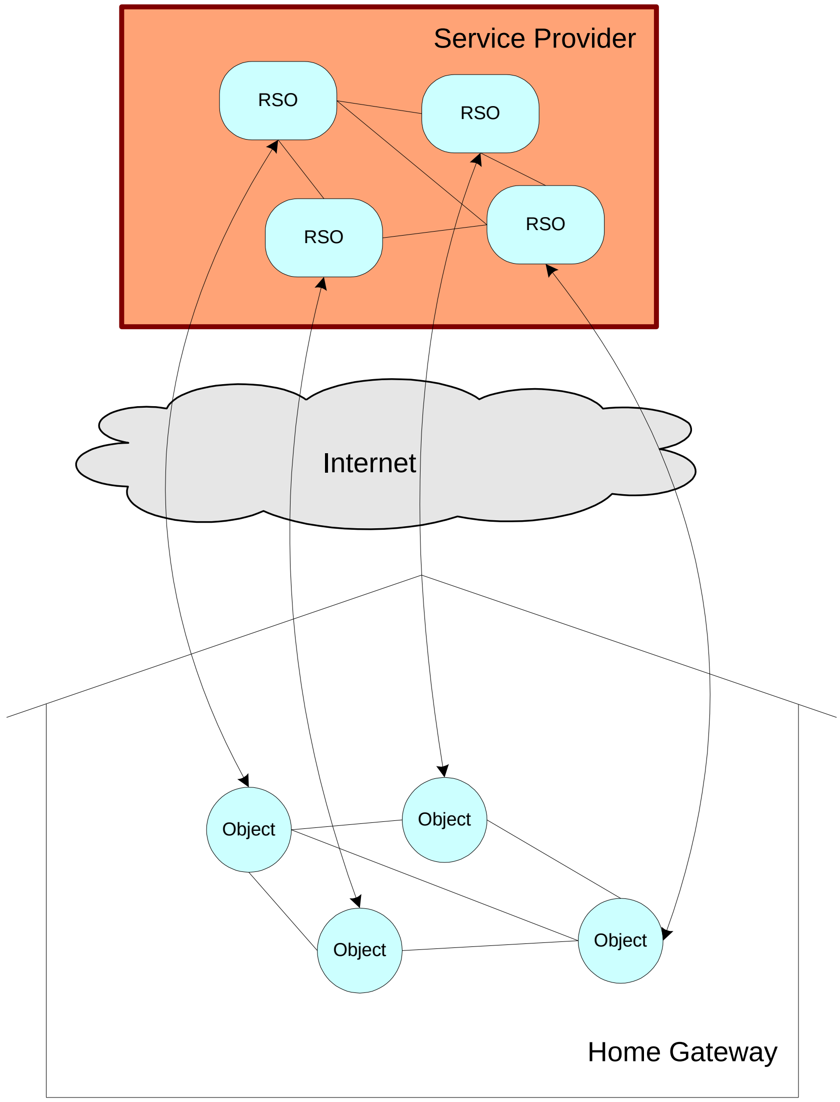

Figure 3.6 TAHi High Level Abstractionof TOA

Even though RSO and PSA provide a consistent and generic way to handle the issues of remote service delivery to the connected home environments, but it is only a conceptual specification that focuses on outline concepts, not its implementation. These concepts are still evolving, which the current specification is fairly ambiguous or subjected to individual interpretation. For example, one of the roles of RSO is to dispatch PSA to check service configuration. It did not explain when it sends PSA and how to check. In fact, the author believes this design is not easy to implement.

3.4.4 eService-On-Demand for connected homes Project

Between 2004 and 2005, Mechatronics Research Group (MRG) and OpenHub Ltd undertook a project entitled "eService-On-Demand for Connected Homes" which aimed to formulate an innovative way for remote service provision to connected homes (Moore, 2005). This project pioneered the On-Demand business model in which various services (e.g. domestic security, energy saving, healthcare and social care) could be made available to the householder as needed. It was the first attempt to implement remote service delivery platform based on the concept of TOA.

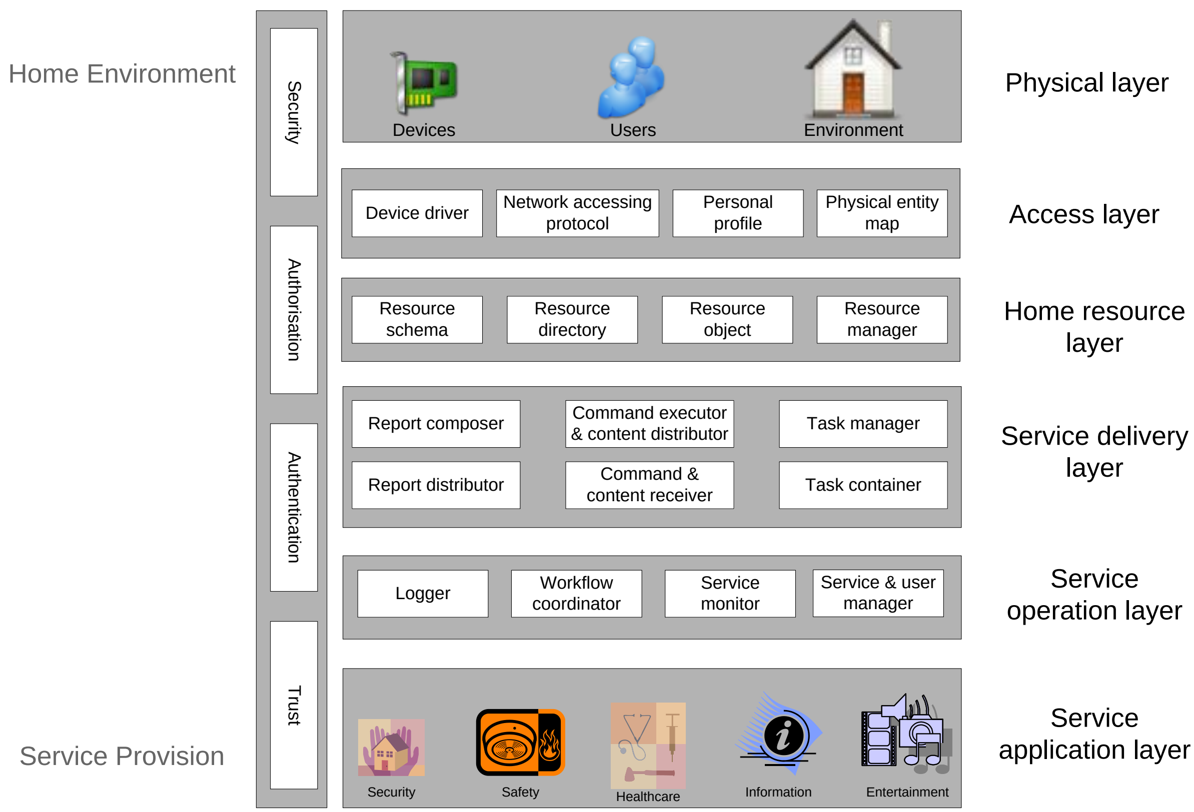

Figure 3.7 Reference Architecture for e-Service Delivery

The reference architecture for e-service delivery is illustrated in Figure 3.7 (Chen & Wong, 2005). In this system, the RSO represents diverse service resources that are available for supporting service provision and execution in connected home environments. Most of the entities involved in service delivery could be regarded as service resources, which include systems and networks in the service supply chain, such as devices, appliances and systems within the environment of the connected home, the service itself, and user interfaces. RSO embeds the service description and offers a set of standard service operations. The service description, which is written in XML provides the characteristic profile of embedded service resources. Basic RSO operations can be divided into lifecycle management and notification. The lifecycle management operation is responsible for the *register, unregister*, and *renew* process of every RSO. The notification operation provides functionality to notify the change of state of service resource and submit data to the interested entities.

Several basic objects (service resources) could be combined together into a composite object to fulfil a high level task. The composite object could be considered as an advanced RSO. Besides basic RSO operations, the composite object treat the underlying basic objects as child objects, and provides *Add*, *remove* and other operations to control the construction of object container. In this way, the composite object is able to aggregate relevant objects into a specific purpose and generates a new abstract and collective interface for new task instead of exposing individual interface.

The focus and major achievement of eService-On-Demand project is the design and implementation of RSOs. In this project, RSOs not only embed service resources (e.g. device, user, and environment information) into remote accessible objects, but also composes relevant objects into a high level RSO to support an advanced service task. RSO is platform independent approach when compared with Java based OSGi approach, which make it less prone to the change of technology.

Although this research project developed an advanced service aggregation approach, it is mainly depending on the device level aggregation that amalgamates the relevant objects into a composite object. The procedure in facilitating a third party to construct services has not been covered.

3.4.5 Existing Architecture Summary

Table 3.1 lists the architecture features are supporting by the current available platforms that might contribute part of the required solution for this research project. Although each platform did address different mix of challenges, none of them would offer all the necessary required support for remote service delivery to connected homes. However, the various design approaches did provide useful reference points for this research project.

	UPnP	Jini	OSGi	TAHi	AIRE	Amigo	EService-On-Demand
Specification	Device advertise its description to control point	Node publish its services to the lookup server	JAR bundle header Mainfest	Remote Service Object	Java object	JAR bundle	Remote Service Object
Configuration	Each device have a DHCP client	Service provider multicast to a lookup service	Service.pid property	Not specified	Hyperglue management component	Defined method like *active* and *stop* within the class	
Addressing	IP address via DHCP or Auto-IP	IP address	Java package and classes name	Not specified	Java class	Java package and classes name	
Message Interaction	SOAP based message, exchange through control point	Communicates services via the proxy	Publish and receive event objects from other bundles	Not specified	Java based Hyperglue communication model	Context management service broker functions	
Remote Access	No Support	Remote Method Invocation (RMI)	Ongoing support web service	Web Service	Web service	Web service	
Service Modelling	XML based	Java proxy objects	Java package based	Pervasive service agents	Megaglue agents	User modeling and profiling service functions	

Table 3.1. Features Supported in Existing and Proposed Architectures

3.5 Summary

Technical challenges for delivering services to connected homes could be summarised as: how to encapsulate device data and environment information into service objects; how to discover and manage devices automatically; how to identify the service object and deliver messages to the one needed; how to support messages interaction between devices and service applications located at different sites; and how to aggregate relevant service resources into a specific service task and manage its execution.

States of the art service delivery protocols and service delivery architectures have been reviewed. Among them, Java-based Jini and XML-based UPnP are often adopted by majority of design and implementation. However, none of them are meet the requirements of remote service delivery to connected home environments, which able to offer “Choice” and support “Mass Customisation” for both service providers and recipients.

OSGi is an open specification focus on the remote delivery of managed services to the connected home and automobile. Many existing connected home research projects have adopted this specification and built their platform upon it. However, OSGi is designed as a gateway standard for both connected homes and vehicles and its Java-based solution constrained the development choice.

The TOA is the only architecture that covers every aspect of remote service delivery in its supply chain. Its service supply chain model encourages a wider and deeper understanding of service delivery and provision. Although its concepts of RSO and PSA are not fully developed when this research project is begun, its vision has opened a new horizon in research and development of future remote service delivery platforms. This research project is also partly inspired of their vision on interoperability at all levels.

The eService-On-Demand was the first implementation of the TOA. It offers a glimpse of potential benefits of the concept in RSO. Compared with the Java based OSGi approach, RSO is platform independent which is less prone to the change of technology. However, this project only demonstrates the RSO based service aggregation approach and did not demonstrate any other aspects in the remote service supply chain such as service contract generation by service providers.

Chapter 4 Analysis and Design for the Remote Service Delivery System

The previous chapters have introduced challenges for developing a remote service delivery system to connected homes which would support “Mass Customisation” and provide “Choice” to the service providers and recipients. This chapter discusses the proposed analysis and design of such a system.

4.1 Requirements Specification of Remote Service Delivery System

With reference to Chapter 2.1.5, remote service delivery environments for connected homes comprise three distinct layers: home environment, service delivery platform, and service provision.

Current research on home environment layer focuses on how to encapsulate various connected devices, appliances, systems, users and ambient information into service objects; register service objects to the home service gateway; and make them accessible for remote service providers. Since the unstructured nature of home environments, i.e. “No home is alike”, it is almost impossible to fully automate these processes. The most accepted school of thought is to adopt a middleman approach in which an Integrated Service Provider (ISP) would be responsible to manage the home service gateway in connected home environments on behalf of householders. According to this approach, the ISP would be responsible to remotely and/or locally manage connected home resources. Although this approach can be beneficial from a technical perspective, the author disagrees with it from user perspective. For the majority would see their home environments as private to them and loathe passing their authority to third parties. This research employs an alternative strategy that would enable householders to retain full control of their connected home environments.

However, it should not expect householders to have the necessary skill and knowledge to manage the complexity of remote service delivery. The home service gateway should be able to take the full responsibility for service objects' registration and life cycle management in an automatic manner. Approaches and methods are also required to monitor and manage the state of change of underlying service objects and forward their messages to appropriate service recipient(s). Furthermore, messages from service objects are also required to have "Service" image built-in (i.e. delivered messages will custom to their service recipients), for example, PIR sensor sends "intrusion message" to security service application but sends "occupancy message" to care service application instead.

Current architectures for remote service delivery do not have the concept in supporting service "Mass Customization" for connected home environments and providing "Freedom of Choice" to both remote service providers and recipients. This research aims to devise a feasible methodology, together with toolset to realize the "On-Demand" remote service delivery to connected homes. As domestic systems and environments are cost sensitive, the proposed solution should not only focus on technical aspects, but also take into account of the life cycle cost.

A method and associated mechanism(s) which are able to present and access connected home resources locally and remotely in a unify way is a "MUST", partly because "no home is alike", and partly because of enabling life cycle management for connected resources such as service discovery and registration. The devised method must able to encapsulate all the details of connected devices, appliances, systems, users and ambient information and only exposes their service(s) interfaces. Exposed interfaces should be described with open standards and able to support the machine-to-machine interaction in local heterogeneous networked environment, as well as over the Internet. Furthermore, remote service providers do not possess the knowledge of the home layout and its relationship with available connected home resources, a method and associated mechanism(s) are also required for the description of home layout and the physical location of connected home resources, if applicable. This leads to the requirement of a universal accepted dictionary to facilitate such information flow. Currently, there is no such required semantic dictionary available and further research to devise such dictionary demands huge research effort and not a small task. Most importantly, such research potentially related to ontology is not the focus of this research and the author will not attempt to conduct this work, but to establish limited vocabularies for supporting the implantation of the demonstrator for this research.

In principle, there are two different execution models to achieve remote service provision: the service applications install and execute in the local service gateway, or the service applications host and execute remotely on systems that managed by remote service providers. The second execution model relies on the quality of Internet connection to deliver their services, which Internet connection may not robust enough to guarantee the quality of services for certain type of services such as remote care and security services. Due to the quality of service concern, the proposed architecture adopts the first execution model although it requires greater power from the home service gateway.

The success of remote service provision is largely dependent on available connected home resources. The delivered service application must be able to operate in a home service gateway. It must also dynamically bind (or aggregate) the available resources and orchestrate their activities to realize the required service(s), which are customized to home settings and meet the needs of householders. The service application represents a contract between the service provider and the household. A method and associated mechanism(s) are required to carry out this task.

4.2 Remote Service Delivery System Building Blocks

The remote service delivery system at the home environment consists of three major components: Remote Service Objects (RSOs), Service Delivery Platform (home service gateway), and Service aggregator. In principle, RSOs represent various home service resources in a uniform way that are required for supporting service provision and also enable service interoperability in a heterogeneous networked home environment; the home service gateway is responsible for the life cycle management of RSOs and the interoperation between RSOs and remote service providers; the service aggregator is responsible for generating service application (s) that would fulfil the agreed service contract and also responsible for the quality of services. With reference to the remote service delivery system discussed in Section 3.2, the remote service delivery system should consist of the following building blocks:

4.2.1 Remote Service Objects

In principle, RSOs represent various home service resources in a uniform way that required for supporting service provision and also enable service interoperability in a heterogeneous networked home environment. With reference to Section 3.2.1, the RSO is required to add/include the “service image” (i.e. customised message according to the type of service recipient) on top of the initial “image” (i.e. raw message regardless the type of service recipient) in manufacturing stage and encapsulate its interface into a set of standard interfaces that based on open standards.

The service provider would retrieve data and information from RSOs which reside in the connected home environment; analyse the data and information; and then generate the service application according to a predefined service contract. The binding and interaction between remote service providers and RSOs is realised through the home service gateway.

The abstract view of the RSO could be represented in Figure 4.1 in which the RSO has to consider two layers of integration, the local area layer and the wide area layer. At the local area layer, the RSO wraps devices or appliances, together with user and ambient data into a service object, and provides a set of standard interfaces through which other devices, appliances and systems interact with. This set of interfaces includes registration interfaces which are responsible for the RSO auto-registration to the home service gateway; a self-description interface that provides the semantic information to the service registry; and a standard interface that enables the communication between service objects and the home service gateway. The capabilities of RSOs would advertise via proxy which is deployed at the home service gateway. The proxy is responsible to translate RSOs into web accessible service objects which are based on open standards to mediate the communication between service objects and remote service providers for supporting remote service provision.

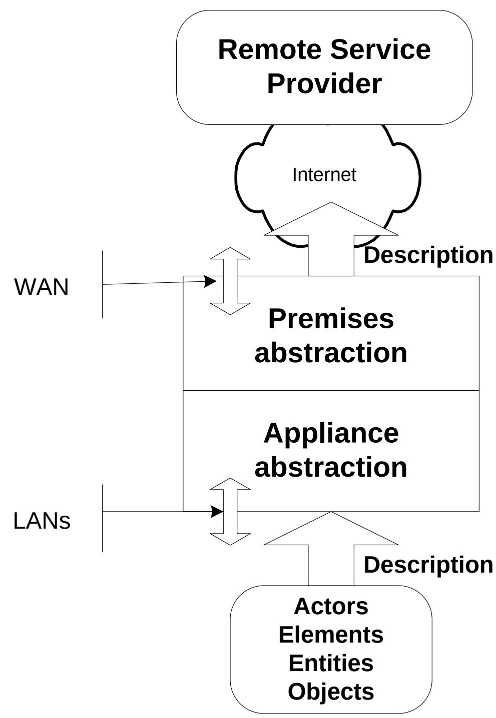

Figure 4.1 RSO: Abstract Interfaces

4.2.2 Service Discovery and Service Management

The design of RSOs embedded part of service delivery mechanisms within service objects, which can significantly reduce the administration overhead and increases the usability of connected service resources. However, there are still aspects of service discovery and management which need to be considered because the diversity of service resources available from connected home environments. As illustrated in Figure 4.2, RSOs encapsulate related service resources into service objects and register themselves to the registry that residing in the home service gateway. RSOs are designed to responsible for the service discovery and life cycle management of underlying service resources. A standard format (or template) will be devised for facilitating processes of service registration, and the home service gateway will publish service descriptions that based on well-defined service dictionaries to remote service providers through a set of web interfaces. Service descriptions will be categorized by service topics and not system (or platform) dependent on enable service interoperability. The service contract between the household and the service provider will be based on a specific service topic. This contract defines service processes into a standard document (or template) which later can be complied and executed on the home service gateway.

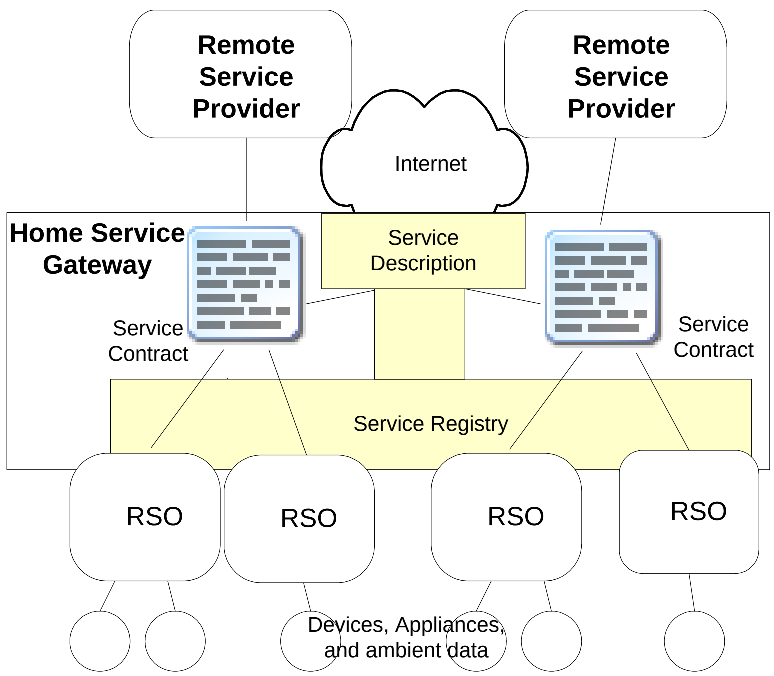

Figure 4.2 RSOs Registry

4.2.3 Addressing Method and Service Description

Remote service delivery platform for remote service provision has to deal with diversity in networking technologies and no single network will meet all requirements in the foreseeable future. Due to this reason, the addressing method and associated mechanism and service description must be carefully constructed. With reference to section 3.2.3, the RSO encapsulates service resources into a standard service object which has been given its own URI(Uniform Resource Identifier) for identification. The home service gateway translates and treats RSOs connected through different local networks as the same and publish proxy objects at home service gateway for their Internet consumers. Therefore, the remote service provider would able to interact with the home service resources via their URI.

RSOs and home gateway is required to present the low level service functionalities in the form of service description which not only be able to describe underlying resources and actions, but also relationships between them. For instance, to structure an advanced security service, information of sensor location and home layout is required, through which the service application can compare detected activities with relevant information to decide if it is really a genuine intrusion, hence reducing the possibility of false alarm.

4.2.4 Message Integration and Event Broker

With reference to requirements stated in Section 3.2.4, RSOs would function as adaptors that translate heterogeneous information sources and formats into standard message formats based on well-defined and open standards based protocols. As shown in Figure 4.4, there are a set of event listeners residing in the home service gateway to handle upcoming events regardless type of networks within the home environment. When an event has been detected, the event listener forwards it to the service container and processes according to the pre-defined service contract. A high level message would send forward to the desired destination. For example, a high level alarm message is based on the comparison result of the detected activity and relevant information.

The service execution container is responsible for the service binding between RSOs represented available home service resources and remote service providers. Services delivered remotely should not expect any human intervention for their daily operations. Therefore, the service container is required to be able to manifest itself against to service contracts, monitor service processes, and also enable to handle and report any exception events automatically.

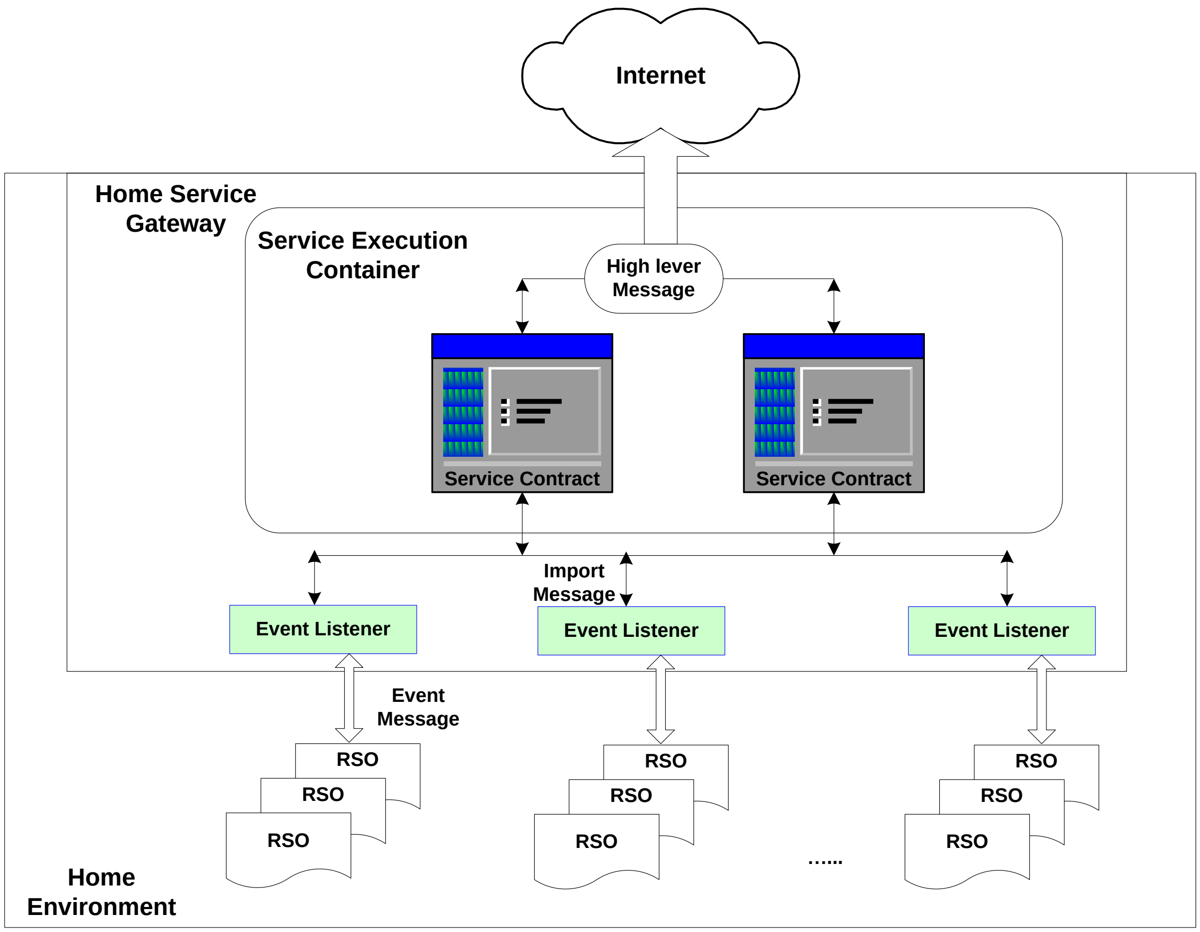

Figure 4.3 Message Integration

4.2.5 Service Aggregation

With reference to Section 3.2.5, to develop a non-proprietary remote service delivery platform with support of service "Mass Customization" and provide "Freedom of Choice" to both service providers and recipients, methods and associated mechanisms for supporting service aggregation are essential.

The proposed service aggregation approach focuses on providing functionalities to handle challenges of dynamic and heterogeneous nature of home service resources. As illustrated in Figure 4.5, the key motive of established concept is to hide the diversity of devices and associated environment information through an integration layer. Thus enabling a smooth system integration and reduce the administration and performance overhead. The integration layer also acts as a proxy in the home service gateway for underlying service resources at the connected home environment.

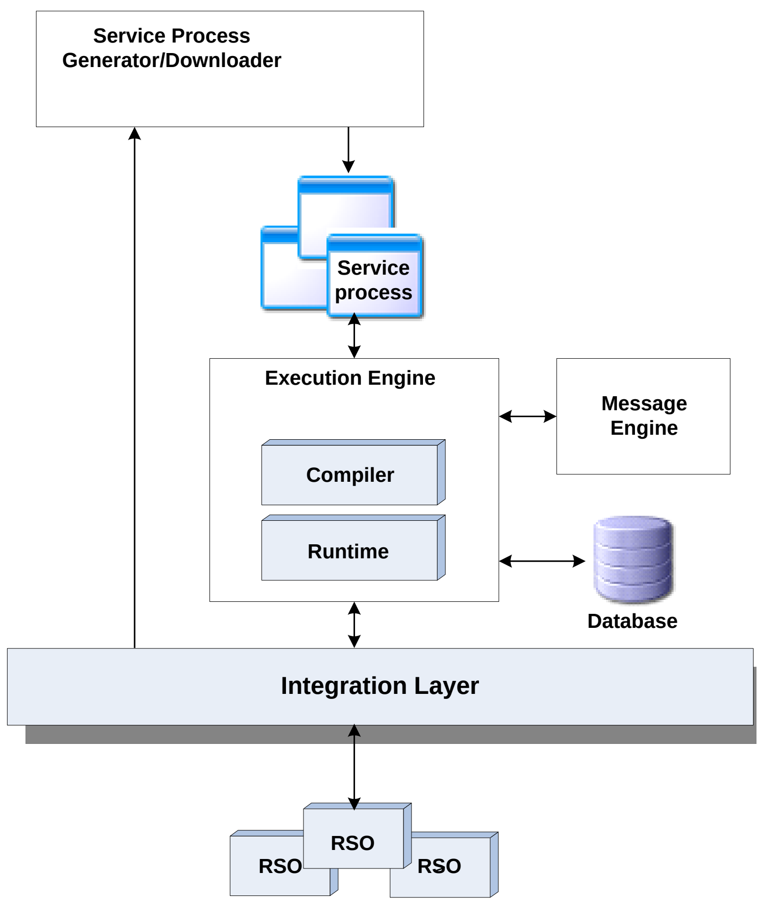

Figure 4.4 High Level view on Service Aggregation Approach

A service process is modelled as activities and goals. Activities are relevant service object events selected from the home service gateway proxy according to requirements of home users. A goal indicates the target to be achieved by the current activity. The service process is based on open standards (i.e. BPEL) and generated by the service provider remotely. The service process generator /downloader are responsible for the building and downloading of service processes to the home service gateway.

The main component for the execution of service processes is the execution engine. The execution engine resides in the home service gateway, and is responsible to dynamically transform source of service processes into execution form. The executable form is an application that can interact with underlying service objects with the support of other service components within the home service gateway. The execution engine also handles the dirty work of process execution by implementing the logic necessary to determine the message invocation and delivery, and providing various error exceptions handling.

4.3 Reference Architecture of Service Delivery Platform

Figure 4.5 presents the overview of the proposed reference architecture. The reference architecture can be viewed from three different perspectives: RSOs and management, service process and management, and execution container.

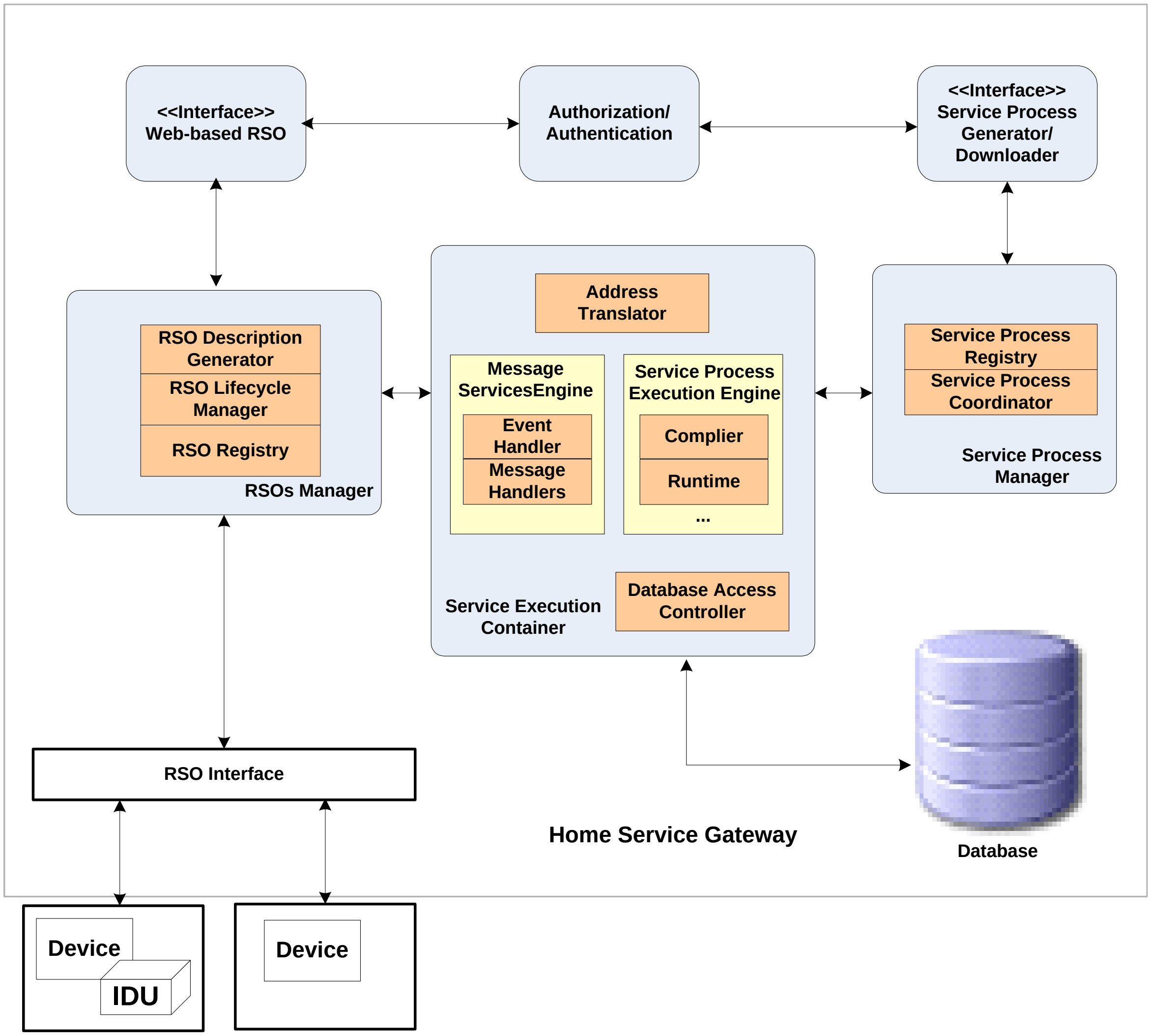

Figure 4.5 Reference Architecture

- RSO and RSO Interface

The RSO acts as a proxy for underlying devices, appliances and/or ambient information. It encapsulates data and information of underlying devices, appliance and/or ambient information into a functional object with a set of standard interfaces through which can interact with the home service gateway and remote service providers. The interactions include the followings: the RSO registers itself to the home service gateway; the RSO provides a self-description on available service(s) to other parties; the RSO submits data and information to interested parties; the RSO responds to the service call from other parties.

- RSOs Manager and Web-based RSO Interface

The RSOs manager is responsible for the registration and life cycle management of RSOs. Information related to RSOs including their location, available services, and information required for other parties to request their services, are stored in the RSO registry. The life cycle management of RSOs is responsible for the installation, updating, and un-installation of RSOs.

The home service gateway provides web-based interface to external service providers. The web-based RSO could be a single RSO wrapped with a set of web-based interfaces, or a combination of several RSOs. The home service gateway also provides the home layout that represents the physical location of RSOs and their potential relationship.

- Service Process and Service Process Generator

The proposed platform supports remote service providers to generate services according to agreed service contracts. The service process in the proposed system is a combination of a set of relevant web-based RSO activities, which can work together and deliver a specific service task. One service task may include more than one service processes.

The service and underlying service processes could be created remotely at remote service provider platform and migrate to the home service gateway, or the home service gateway provides a remote accessible integrated development environment that allows remote service providers to generate services and service processes directly on the home service gateway.

- Service Process Manager and Service Process Execution Engine

The service process manager is responsible for the configuration and management of services and their underlying service processes. It provides a mechanism to establish a communication channel between RSOs and remote service providers.

The service process execution engine is responsible for transforming XML based service processes into executable form. The execution engine runtime handle the details of process execution by providing an application-level interface and translation mechanism.

- Message Service Engine and Address Translator

The message service engine is responsible for transforming heterogeneous information into well-defined standard message format. It includes a set of event listeners that react with upcoming events through different networks in the connected home environment. When an event has been detected, the event listener forwards it to the service process execution engine and processes it according to the pre-defined service contract.

The address translator is responsible for the identification of each RSO and the translation of network address as well as protocol between different local network media.

- Database and Database Access Controller

The database provides a persistent storage for RSO profiles and service processes. The database access controller mediates the interaction between the service execution runtime and the underlying data store. It facilitates the implementation of Intranet and Internet access of data/information residing in the database and provides functionality for storing, retrieving, modifying and deleting the data/information.

4.4 Summary

Requirements specifications for developing a remote service delivery system can be viewed from three distinct layers: home environment, service delivery platform, and service provisions. Since majority householders would like to keep the privacy of their home environments, this research employs an innovative strategy that would enable householders retaining the full control of their connected home environments without involving technical details.

The remote service delivery system at connected home environments consists of three major components: Remote Service Objects (RSOs), Service Delivery Platform (home service gateway), and Service Aggregators. In principle, RSOs are responsible for representing various home service resources in a standard format, supporting service provision and enabling service interoperability in heterogeneous networked home environments; the Service Delivery Platform is responsible for the registration and life cycle control of RSOs and the interoperation between RSOs and remote service providers; service aggregators are responsible for fulfilling the service contract and controlling the quality of services.

This research also takes into account the concept in supporting service “Mass Customization” for connected home environments and providing “Freedom of Choice” to both householders and remote service providers. Six building blocks have been identified to devise a feasible solution for remote service delivery to connected homes. RSOs are required to add/include “Service image” on top of the initial “image” in manufacturing stage and convert their interface into a set of standard interfaces through which remote service providers can retrieve data and information from connected home environments; service registration and management is required to support automatic registration and life cycle control of underlying service resources; address mapping and service description is required to convert RSOs connected through different local networks as the same and present low level service functionalities at the home service gateway; message integration and event broker is required to translate the heterogeneous information source into a standard message format based on open standards then detect upcoming events and forward them to the pre-defined service contract; service aggregation is required to handle the dynamic transformation of service processes into executable forms and manage the execution of service processes.

Reference architecture is presented to introduce the overview of remote service delivery system and the relationship between different building blocks.

Chapter 5 Realisation of Remote Service Provision for Connected Homes

5.1 Introduction

The proposed reference architecture lays a foundation for realising remote service provision to connected home environments. In this chapter, methodologies for realising remote service provision are proposed.

5.2 RSO Development

RSOs encapsulate various types of initial service resources, such as device information at manufacturer stage, environment information, and user identity into standard service objects to facilitate service interoperation and communication. As the interoperation mechanism of remote service delivery platform is based on XML confined by XML schema, hereby, it indeed is a service-oriented platform at where service resources and applications can be treated as registered service objects.

Under this principle, RSOs can briefly be catalogued into three types: *DomesticObject, CompositeObject, ProxyObject*. *DomesticObject* encapsulates service resources in the home environment; *CompositeObject* specifics advanced service functionality; *ProxyObject* acts as a mediator to bridge *CompositeObject* and remote service providers.

The RSO design and implementation can be largely divided into three topics: deployment structure, object model, and RSO interfaces.

5.2.1 RSO Deployment Routes

There are two possible routes to deploy RSOs: distributed and integrated.

Table 5.1 Comparison between two possible RSO deployment Routes

	Distributed Route	**Integrated Route**
Gateway resource consumption	As RSO run at gateway, it could require significant amount of computing resource from gateway.	RSO mainly consume device computing resource. It less relies on the gateway resources.
Scalability	The system scalability is both restricted by the gateway computing resource and communication capability.	The system scalability is only deterred by network capability.
Installation	RSO has to be deployed at the gateway before installation process could be carried out. It could be tedious with many manual involvements.	The embedded RSOs are ready to interact with gateway. The installation process could be as simple as “Plug-n-Play”.
Off-line	When device is off-line, RSO can still response to some request. It can also temporary store some messages and pass on to the device later.	When device is off-line, RSO and device cannot be contacted. Gateway acts as coordinator to store and pass on message when required.
Un-installation	After certain time of off-line, RSO can un-register itself from gateway and clear garbage autonomously	After certain time of off-line, gateway responds to un-registration and clear up garbage
Configuration persistence	If device reconnect to gateway after clearance, the gateway couldn’t maintain the previous configuration	If device reconnect to gateway after clearance, RSO can maintain the previous configuration
Interoperability	Device can only achieve interoperability through gateway	Devices are readily interoperable to any compliant service objects
Collaboration	RSOs associated with devices can form logic clusters with the restrict of gateway	RSOs embedded devices can freely form logic clusters without restriction
Resilience	The system is prone to central failure. When the gateway has gone, the entire system is dead	The system has certain level of resilience. When the gateway is dead, the system can still carry out some tasks

For the distributed route, RSOs are deployed at the home service gateway and communicate with devices through a proprietary protocol. Hereby, RSOs play a similar role as device drivers. For the integrated route, RSOs are hosted by devices. RSOs can interact with home service gateway as well as other service objects as far as they could understand its XML message. The technical tendency is to embed part of service delivery platform into devices. The main differences between these two structures are summaried in table 5.1.

The distributed route is suitable for services only involving handful devices (e.g. weight monitoring) and occasionally used devices (e.g. TV). The integrated route is suitable for services involving complex devices and requiring high resilience (e.g. security). But devices are required to constantly keep the connection.

5.2.2 RSO Object Model

RSO object model contains three sets of interfaces: basic operation interface, event interface and domain specific extension. The basic operation interface provides interface to facilitate the management operation; the event interface provides a standard format for message exchange; the domain specific extension defines interface to fulfil individual domain needs.

As shown in Figure 5.1, any compliant objects stem from the abstract object *RemoteServiceObject*, which provides the description and offers several standard operations. The description provides the object profile written in XML and confined by XML schema. The *register, unregister, and renew* operations are shared by all objects through which define a standard way to inform others who it is, what it can do and how long it will available. The RSO model adopts a lease scheme to manage the RSO life cycle. When a RSO registers to the system, it will specify its life time or expire time. When the life time has been reached and no renewal has taken place, the system assumes the RSO is no longer available and will clear it automatically.

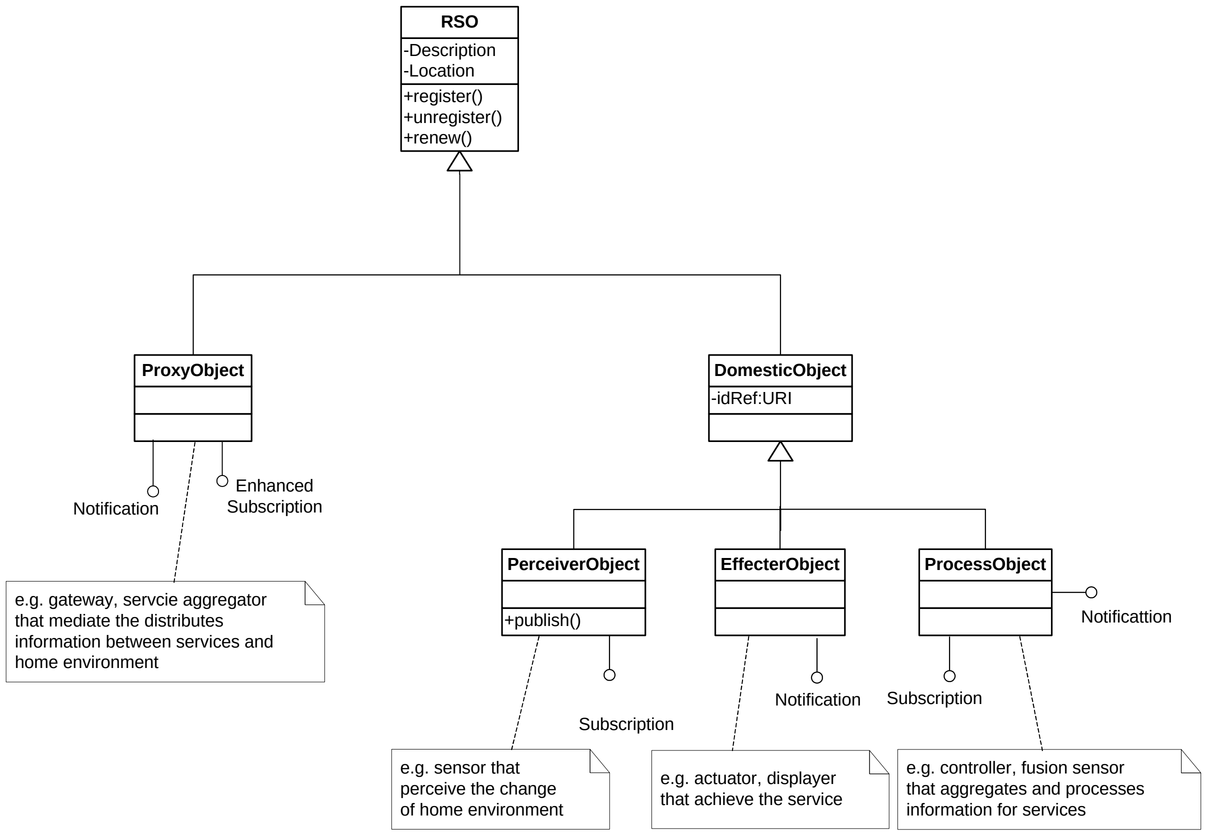

Figure 5.1 RSO Model Object

The subclass *DomesticObject* and *ProxyObject* directly inheriting from *RemoteServiceObject* represent objects in home and service environment respectively. Based on the characteristics of object, *DomesticObject* can be further evolved into *PerceiverObject, EffectorObject* and *ProcessorObject. PerceiverObject* represents sensing resources that sense the change of home environment. It could be sensor at home or personal database tracking personal information, which provides information through *Subscription* interface. *EffectorObject* represents end effectors that manipulates home environment accordingly. It could be actuator and displayer, which received information through *Notification* interface. *ProcessorObject* likes an information hub, which aggregate and processes information for service, and perform some predefined service tasks at home. It could be controller and fusion sensor, which needs to implement both *Subscription* and *Notification* interface. *ProxyObject* gathers, organise, and mediates

DomesticObject within home environment and publish abstract *DomesticObject* containing every relevant objects to remote service provider. Due to its roles, it implements two interfaces for *Notification* and *EnhancedSubscription.* In terms of physical implementation, it could sit on residential gateway or home portal, which is intended to reduce the complexity of home service resources and facilitate the access of service applications.

Although objects use URI schema to represent accessible address, but the URI of objects within home environment should be able to represent and hand non-IP address. Comparatively, the *ProxyObject* outside home should be an IP compatible address, as service applications are only deliveried via Internet.

The essential interface implemented by RSOs can be summarized in Figure 5.2, notably *Subscription, EnhancedSubscription* and *Notification.* These interface encapsulate and facilitate the communication operations (i.e. publish & subscribe communication model). *Subscription* and *Notification* is a pair of interface to enable event-driven communication mechanism. *EnhancedSubscription* is an enhanced version of *Subscription*, which offers programmatic capability to send notification message. However, it still more complicated consume more computing power than *Subscript*. Hence, it is likely to implement in server-like object.

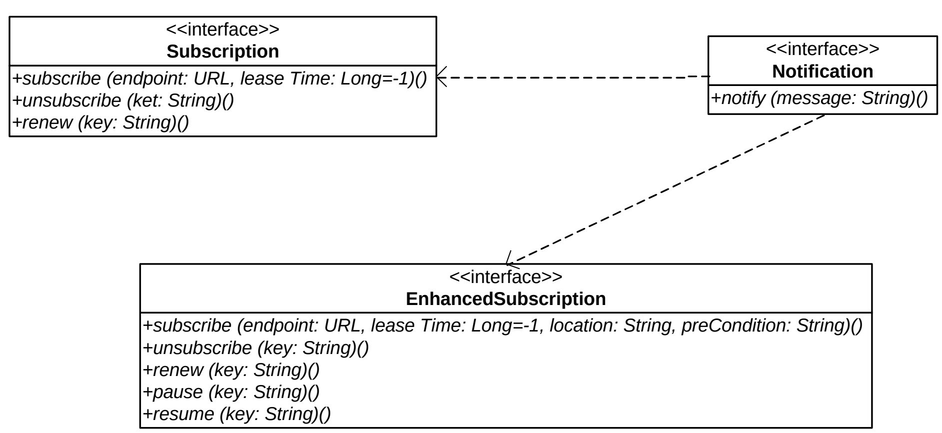

Figure 5.2 RSO Essential Interface

5.3 The Address Mechanism

Currently, there are various home networks (e.g. Bluetooth, Ethernet, LON) coexists at home environments. RSOs are designed as a mediator and adapter that convert data and information from different networks into general standard and enable them to interoperate with each other's. Inspired by the success of Web Service for the business integration, RSOs adopted Simple Object Access Protocol (SOAP) as the basic communication protocol. Therefore, RSOs not only can take the advantage of the openness and robustness SOAP offers, but also can benefit from its future advance in terms of technical solutions and support tools. In addition, both SOAP and Web Service technology are easy to align with the RSO development and service delivery solution. Although currently the majority of implementations of SOAP are for IP-based network, as far as the network is able to transfer the message, it is not forbidden to use a non IP-based network. Despite the HTTP is the only protocol described in the SOAP specification, any transportation protocol can also be used. The communication model for RSOs consists of four elements: address, transportation relay, registration, and notification.

5.3.1 Addressing

An addressing mechanism is required to identify each RSO and simplify RSO message delivery (e.g. message routing, address parsing) to the specific object. In order to take advantage of existing standard and web development, the RSO implementation adopts and extends URI standard as address mechanism. The address is made up of several fragments, which is separated by the standard URI separator "/". Each URI fragment is a qualified URI representing unique address within specific domain. The URI is consisted of a sequence of three main components: <scheme>://<authority><path>. The meaning of this specification depends on individual developments but will not specify the usage of the query part defined in the URI standard.

As the Internet is the dominant force and backbone of wide area communication, the root domain of a nested domain must be a URL (Uniform Resource Locator) that can be interpreted by DNS (Domain Name System). The other domains are appended behind it following the validate route to access the specific RSO. As illustrated in Figure 5.3, RSO_2 address would be *http://mobile.service.com:80/sms://07930123456*, which consists of two domains: mobile.service.com and 07930123456. This address schema not only indicates the address within each domain, but also specifies the transportation protocol within a certain domain. Thus, entities belonging to different domains can easily figure out the shortest and most efficient route to reach each other. For example, only way that RSO_1 can reach RSO_3 is via gateway point (http://myGateway.home.com:80/bluetooth). For the communication between RSO_3 and RSO_4, it can be achieved without WAN connection. The route is even shorter for RSO_5 to connect to RSO_4. Principally, it is allowed that one physical node can host multiple RSOs and the number of RSOs is only determined by it capacities such as memory, computing power.

The focal point linking two network domains namely *pivot* plays a vital role for communication services and their addresses implementation. Every pivot presents at two network domain and can be accessible from both domains. It responds to translate network address as well as protocol between two domains. The detail about its functionality will be described in the following section. For the home environment, the residential gateway or gateways is the hub(s) to physically connect various network media and handling diverse protocols. It is unsurprised that the most pivots are deployed in the gateway(s). But it is not compulsory.

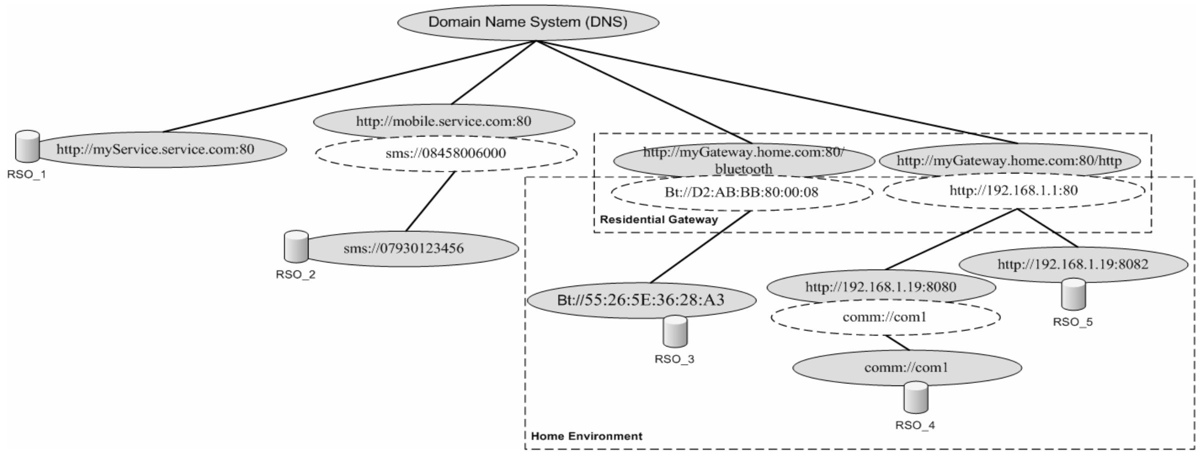

Figure 5.3 Addressing for RSO

5.3.2 RSO Relationship and Home Layout

Since this research study will not cover the semantic web and ontology description, a graphic home layout is provided to help third party to understand the home setting and analyze the possible relationship between domestic service resources. Although there are many existing software tools that can support the drawing of home design and layout, this research study uses Windows Office Visio, which could be installed together with Office or freely downloaded from Internet.

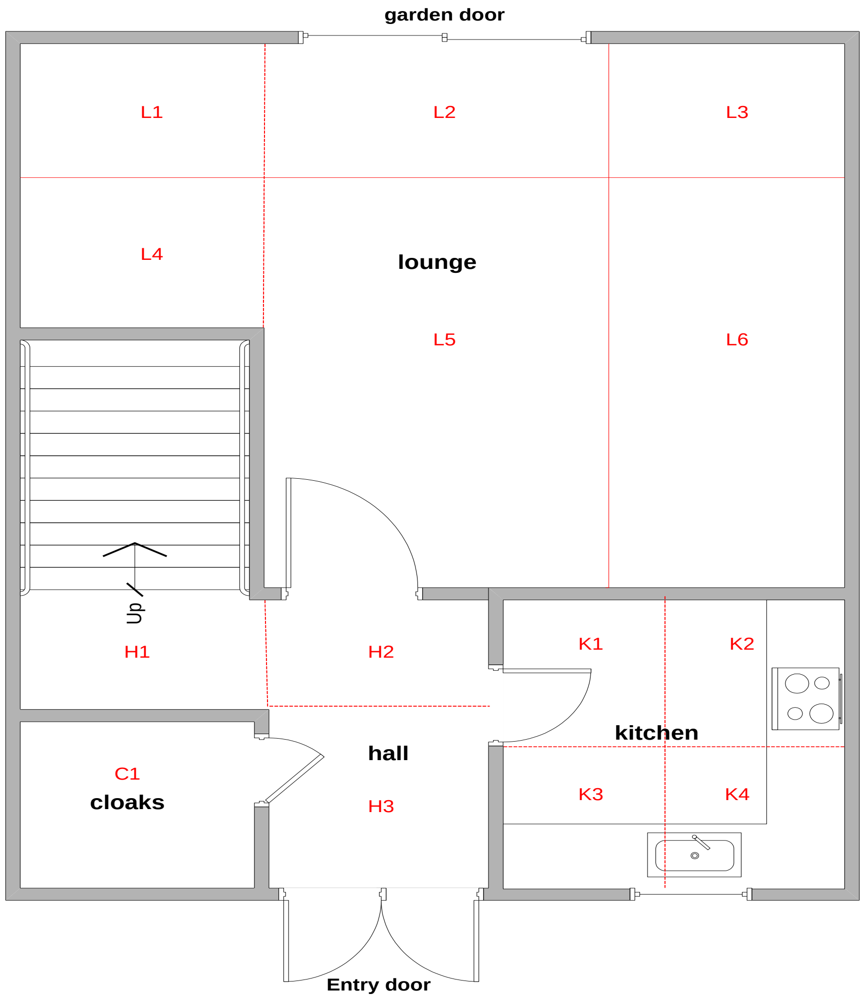

Figure 5.4 Home Layout Example

As illuminated in Figure 5.4, the drawing indicates the basic shape of the home by the walls, rooms, stair and structure stencils. The position of door and window is clearly displayed and each independent space is given its own name. The detail of home plan is absent to protect home user's privacy. Every room has been divided into various sections and each is given a unique code. For instance, the hall has been divided into three sections. Thus, if a motion detector has been installed at H2, together with its functional description; remote service provider should be able to understand its functionality inside the connected home. Although this is not an idea solution in terms of automation and intelligence, however, as current connected home services do not involve too much service resources, this solution should be able to facilitate the remote service provide to understand the home setting and device locations.

5.4 The Communication Mechanism and Information Manager

5.4.1 Transportation Relay

A SOAP message for RSO communication could go through a chain of domains and protocols before reaching the destiny. The whole transportation process can be described as transportation relay as illustrated in Figure 5.5. The entire process can be described as that the message is loaded onto a specific protocol and transported to a middle point named *transport pivot*, and then the transport pivot responds to unload message and load it onto another protocol for another pivot or destination. Finally it reaches the destine RSO. The basic functionalities of pivot must include two essential categories: one pair of addresses mapping between two network domains and a utility to shift messages between protocols.

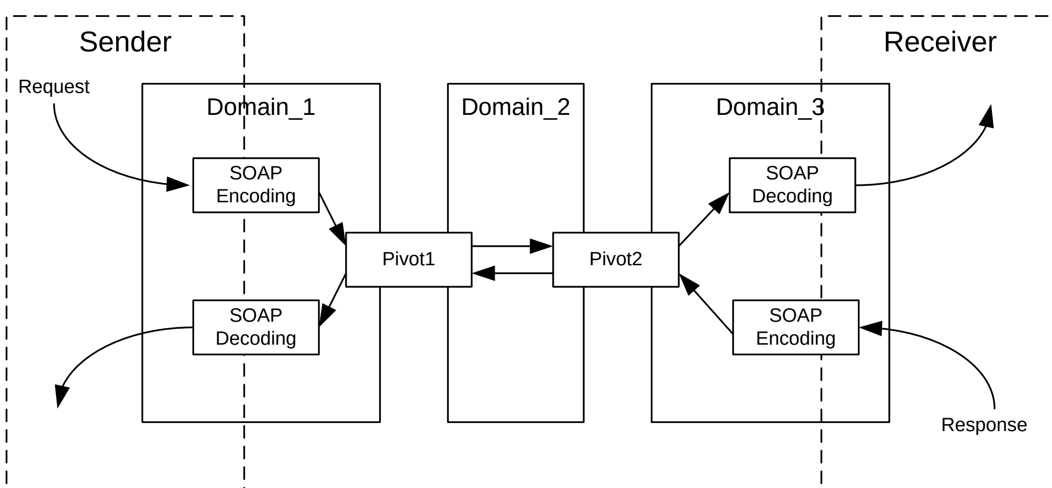

Figure 5.5 Transportation Relay Flow

In terms of a message object, it is contained within soap envelope has two parts: the message data itself, named payload or message body, and the message headers, as shown 0. The payload of the message (SOAP body) must keep untouched during the transportation process. The message headers, however, is employed to facilitate routing the message through various domains. The address adopting the above described schema will be used to formulate the address for the header. The pivots direct messages through the path by means of subtracting the certain address fragment.

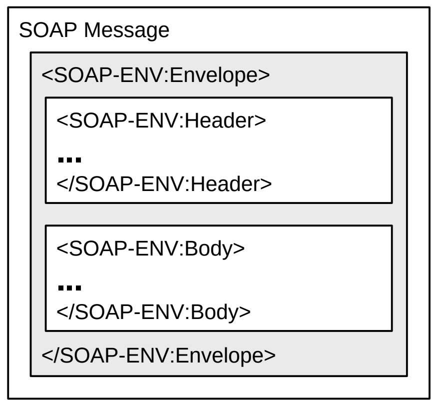

Figure 5.6 SOAP Message Structure

5.4.2 RSO Registration Processes

The main goals of registration process are:

- Registering RSO into a local registration server;
- Uploading RSO description to the local registration server;
- Figuring out a feasible route to access RSO;
- Assigning a unique and accessible address for every RSO.

It is assumed that every RSO is capable of obtaining a validate address within it existing domain. For example, if RSO within IP domain, it will use an arbitrary IP address or employ DHCP to obtain an IP address.

The registration is built on the top of transportation relay scheme and SOAP message. Since the address for registration server is unknown and could be various depending on the system installation, asterisk or "*" instead of a definite address is used for the destination of registration message. The message will be propagated until it reaches the registration server or the available limit.

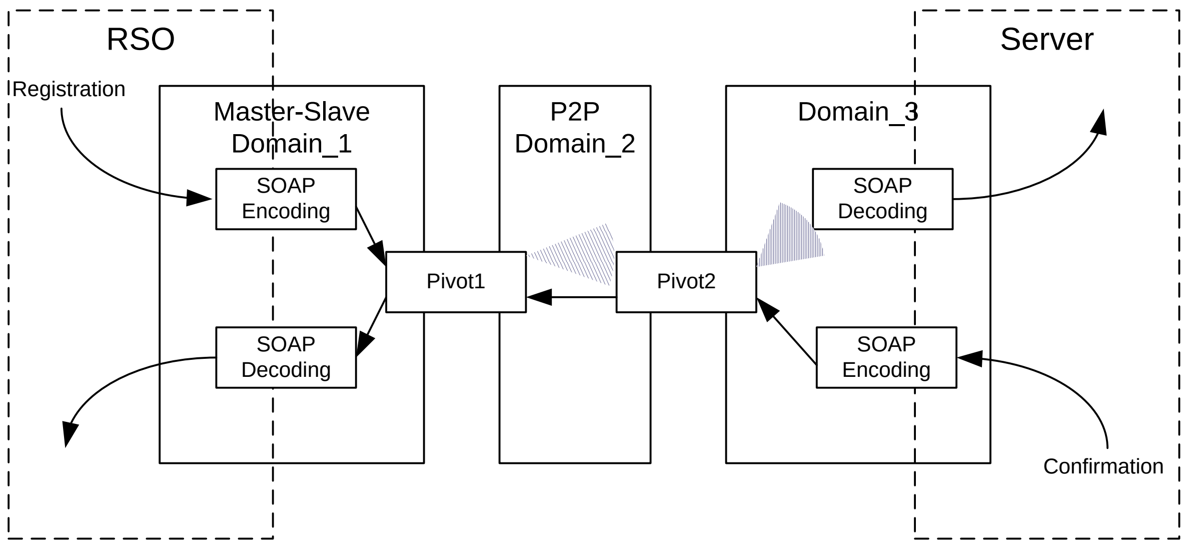

Figure 5.7 Registration Process

The registration process can be classified into two patterns: master-slave scheme and P2P (Peer to Peer) scheme. The master-slave scheme only allows one master node within the network (i.e. USB). Thus, it is the only place where the registration server could be. RSO will send the registration message directly to such arbitrary address. Within P2P scheme, there is no pre-defined node for registration server and every node is equal (i.e. IP network). RSO broadcasts the registration message and server picks up such message whenever it is available to achieve registration process. If there is no server within a certain domain, the pivot will assist to propagate the message. During the process, each pivot will establish the route by means of concatenating domain address the message passing by. The entire process can be illustrated as 0.

5.4.3 Notification Schema and Event Generation

The RSO communication is based on publish-subscribe model, where publishers (RSOs) selectively and periodically push events to subscribers (remote service providers). A set of event brokers are deployed on the home service gateway that serve as intermediaries in delivering events from publishers to subscribers. Instead of assigning subscriptions with similar interests to the same broker, this research employs a mechanism that includes service topic at the head of event message through which the home service gateway is able to forward the event message to the predefined service contract.

The notification schema adopts the SOAP protocol, the communication mechanism is beyond the request-reply paradigm. Here, the description is the extension of the SOAP protocol, which is considered as the added-on elements for the standard without contaminating the basis.

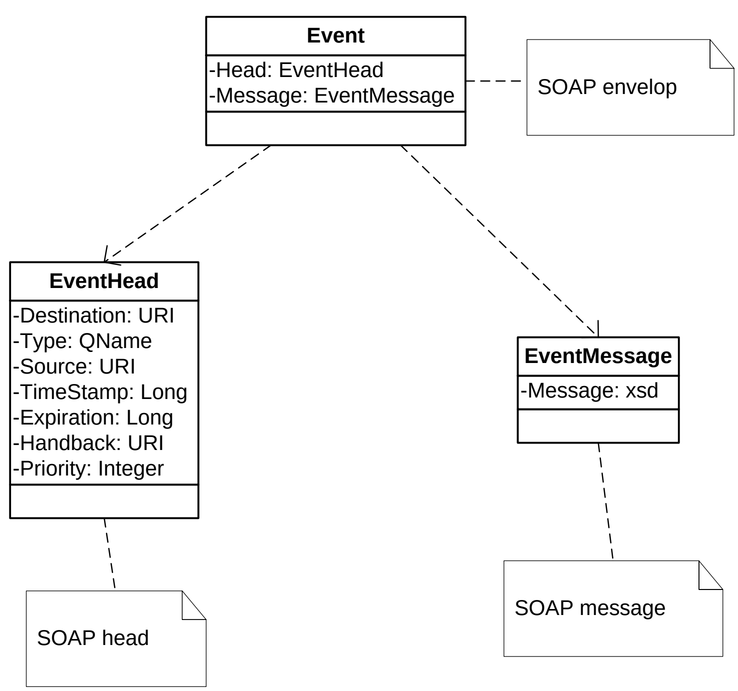

Figure 5.8 Message Model

The message that is passed on between objects is specified in Figure 5.8 based on the SOAP standard. The message platform consists of head and message two parts. The head includes application independent information to facilitate message delivery and process; the body carries application specific information that is confined by application XML schema. The main extension focuses on extending and specifying the *header* of message, which is normally ignored. The *message header* provides metadata about the message, which facilitates to transfer and process the *message body*. The details of the head are:

- *Destination* - The *Destination* header identifies the destination of message delivery, which is specified by URI. It is valuable to service aggregator and service proxy that re-directs the messages.
- *Type* - The *Type* header uniquely identifies the type of *message* payload. The main purpose is to identify the message structure and type of payload as a simple way for message consumer to label the message body. The type includes two parts: namespace and type hierarchy. The namespace distinguishes the type defined by different party that is identified by URI; the type hierarchy is differentiated with dots. For example, the type can be given as http://ehome.mrg.dmu.ac.uk:healthcare.monitor.scale.identificableScale.
- Source - The *source* identifies the originating point of message. The main purpose is to provide the message consumer to verify whether accept or reject further process the message.
- SeqNo - The *SeqNo* identifies the sequence of message. It provides the message consumer a mechanism to eliminate the duplicate message and acknowledge the message loss.*TimeStamp* - The *TimeStamp* represents the time that the message is generated. The time is the elapsed time in milliseconds since midnight, 1 January 1970. The main purpose is to facilitate the message archive and validation about the message.
- *Expiration* - The *Expiration* gives the validation time of the message. When the expire time is passed, the message should be omitted rather than further process.
- *Handback* - The *Handback* is the reference point to pass the response message back to message provider. It is possible that the message provider cannot directly handle the response message. The message could be handled by intermediate (e.g. gateway).

- *Priority* - The *Priority* specifies the order to process the message. It is optional that message consumer can demand the urgency about the message. The priorities can be graded from level 0-4.

5.5 Service Aggregation Development

Service aggregation play crucial role in remote service provision. Advanced services such as health care services to elderly or disadvantage people are only become possible if different types of connected home service resources could be collected and performed in a certain sequence. To support the "On-Demand" business model with "Freedom of Choice", the proposed remote service delivery platform is required to be an open platform which could facilitates third party remote service providers to participate the creation and provision of connected home services.

5.5.1 Service Aggregation Development Structures

There are two levels of deployment structures for service aggregation: device level aggregation and service level aggregation. The device level aggregation is responsible for integrating primitive resource entity (i.e. sensor, environment information) into high level service objects; exposing service objects in home service gateway with web-accessible interfaces; and enabling service objects to be interactive with remote service providers. The service level aggregation is responsible for orchestrating relevant service objects into a specific service application; converting the source service application into a complied representation suitable for execution; and managing the interaction with underlying service objects.

- As shown in Figure 5.9, the device level aggregation can be a simple wrapper that gives connected home service resources a set of standard web-based interfaces, or can be a service compositor that combines several primitive devices into one service object. Web-based interfaces can be developed on the RSO if it embedded enough programming capability or on the home service gateway as a device driver.

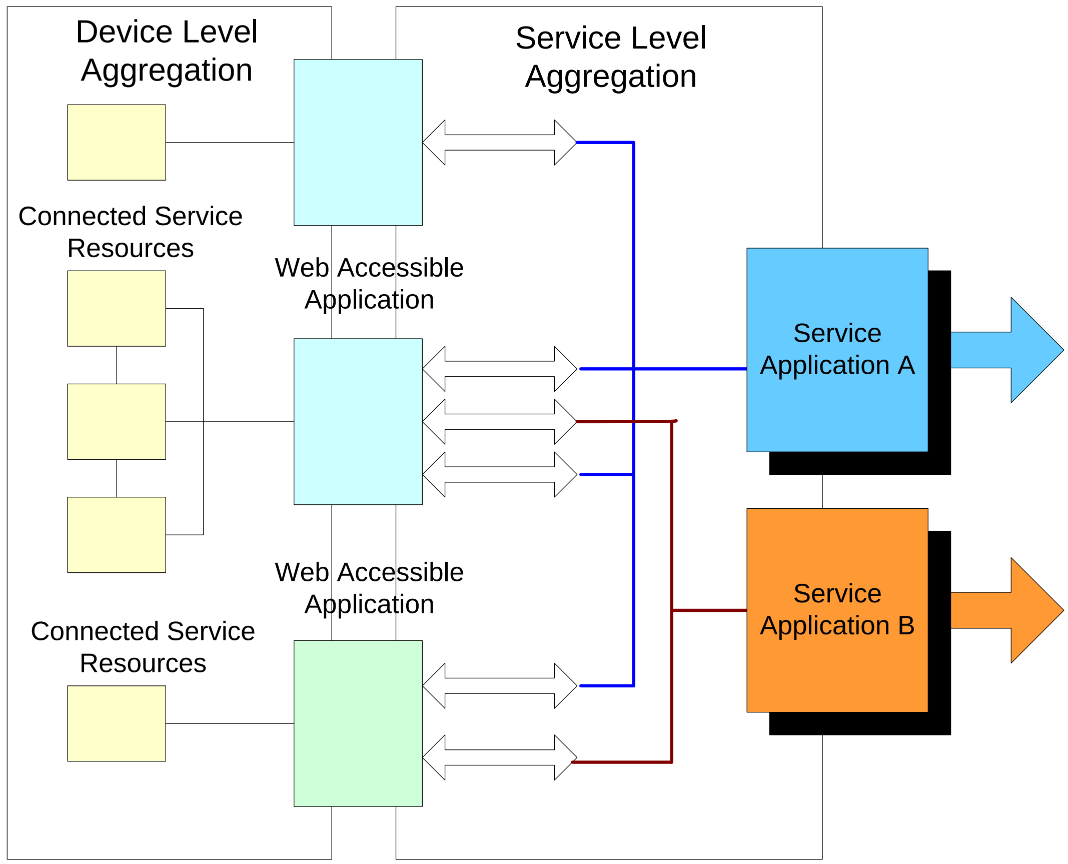

Figure 5.9 Service Aggregation Deployment Structure

- The service level aggregation is a service application that orchestrates relevant connected home service resources into a specific service task. The service application is self contained and executes automatically. Connected home service resources should be able to sharing between different types of service applications with different meanings regardless of their application domains and service platforms

5.5.2 Service Aggregation Model

The class diagram of a service aggregation model is shown in Figure 5.10. The diagram presents attributes and model associations to support a specific service task.

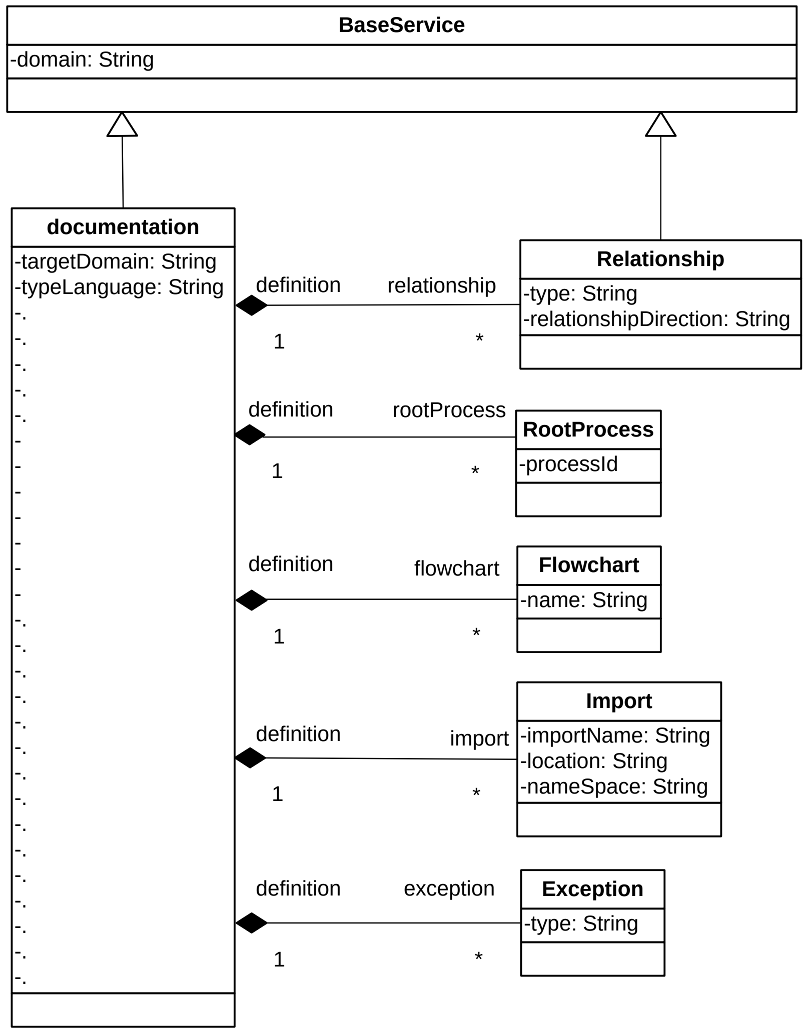

Figure 5.10 Service Aggregation Model

The *BaseService* is an abstract super class for each service application (i.e. security, healthcare, or energy saving) It provides the attribute id and documentation, which underlying service process will inherit. The *domain* attribute is the name of the service (i.e. health care and security) is used to uniquely identify this service application. The *documentation* includes the description of the service application and the definition of associated model and attribute. The *relationship* class is optional, which is used to define the relationship between different underlying service processes. One service application may contain several service processes. Each service process has it own unique *processId* and defined in the documentation. The *Flowchart* class is optional as well. Some system would like to provide high level solution, which can generate service process directly from flowchart. The *import* class is used when referencing underlying service object port binding. The event send out from service objects will active a service process through the event broker connected to the service application execution engine. The exception class is used to handle the error raised during testing and execution.

5.6 Service Application Creation and Execution

Since the domestic users are comparatively more cost sensitive, the home service gateway should be relatively “thin”. Therefore, the proposed approaches should avoid place any service application on the home service gateway which requires the using of large historical data.

5.6.1 Service Application Development Strategy

The cost sensitive nature of connected homes makes the service applications development strategy different than commercial and industrial systems. Since the home service gateway is expected to be “thin”, the proposed service delivery platform would not support complete historical data storage inside the connected home. Alternatively, the proposed methods support data/information to be submitted to the interesting remote service providers, who are responsible for the data storage and utilization. Therefore, the service application in this research study is only responsible for the specification of service processes. As shown in Figure 5.11, each service application (i.e. security, healthcare, and maintenance) could contain more than one service process. These executable service processes integrate closely to connected service resources and execute according to the pre-defined process order to provide an advanced service to the householders.

There are two types of deployment structure for service applications: centralised and decentralised. For centralised structure, the service application is deployed and hosted on the remote service provider's platform. This centralised management solution is same as the majority of current commercial systems and supported by all existing service aggregation engines. In the decentralised structure, the service application is deployed by the remote service provider but would be executed on the home service gateway. The decentralised management solution requires a service application execution engine to be embedded in the home service gateway, which is responsible for the conversion of the source service application into a complied representation suitable for execution. The centralised structure is suitable for connected homes which have stable and continue Internet connection.

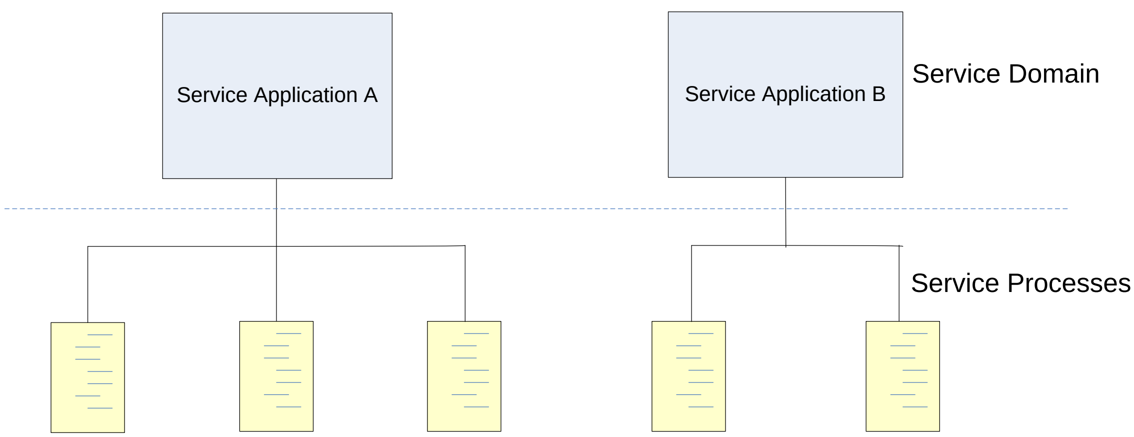

Figure 5.11 Service Application Development Structure

Therefore, the service process placed at the connected home is much simple than those at enterprise environment. In this research study, as shown in Figure 5.11, each service task (i.e. security, safety and maintenance) could contain more than one service process. These executable service processes integrate closely to the underlying RSO and execute according to the pre-defined process order to provider an advanced service task.

5.6.2 Service Application Design-time and Run-time

The development of the service application should be considered both from design-time and run-time. As illustrated in Figure 5.12, at design-time, the remote service provider create the service application and underlying service processes using the available home service resources exposed at home service gateway. The service application is XML based which could not directly interact with underlying service objects.

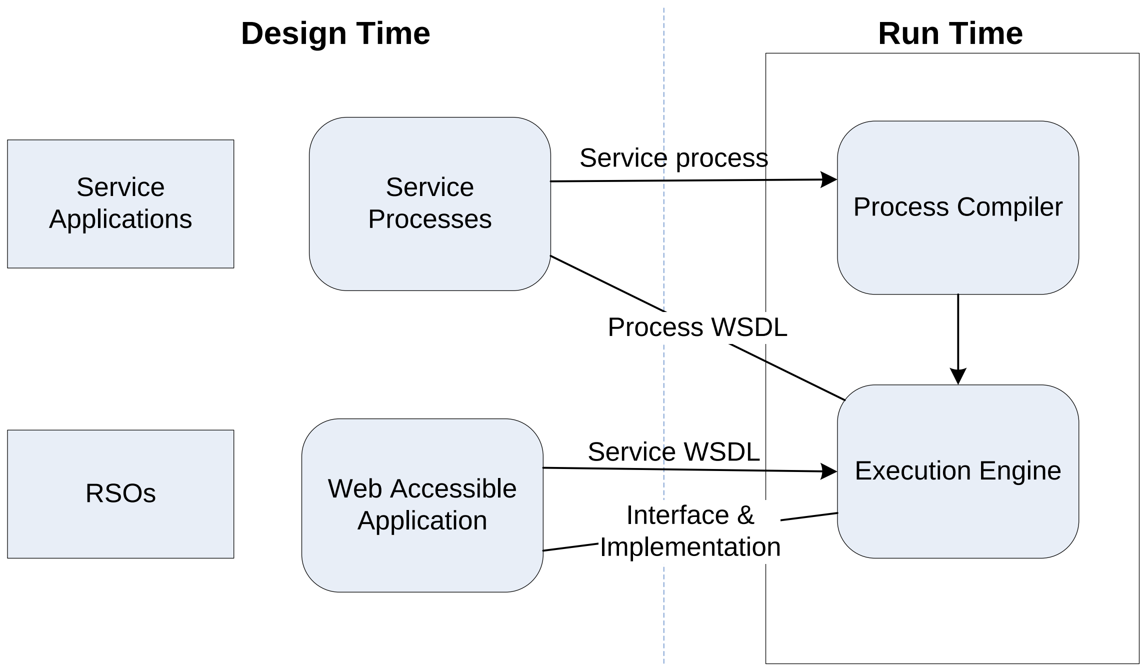

Figure 5.12 Service Process Design-time and Run-time

However, the service process has to be converted into a run-time executable format. An execution engine with compilation capability needs to be introduced to handle this type of complexity. The *process complier* is responsible for the conversion of the XML based service process into a complied representation suitable for execution. The output of the complier is either a "good" representation, or a list of error expectations indicating problems with the source service process. The generated representation by the complier is an object model similar in structure to the underlying RSOs.

The *execution engine* provides for the execution of the complied service processes. The run-time implementation of service process constructs at the instance level and provides an application-level interface and translation mechanism for interrupting execution and persisting execution state. In detail, the execution engine provides a persistent virtual machine for execution service processes.

5.7 Authentication Mechanisms

The authentication method utilises in this research study is based on Web Service Security (WS-Security) specification. WS-Security is a specification for securing Web Services by associating security tokens with messages. (Satoh &Yamaguchi 2007) As shown in Figure 5.13, the householder sends the service requests and home service gateway IP address to the service provider to order a service. The remote service provider then send a Security Binding Assertion, which specifies security tokens used for signatures and encryptions, back to the home service gateway. The home service gateway would immediately pop up a notification window to inform the householder and ask for the permission to access home service resources. A security authentication model developed in the home service gateway is responsible to transform the security policy into configuration files and generate a WSDL based platform configuration file which could be gained by the remote service provider.

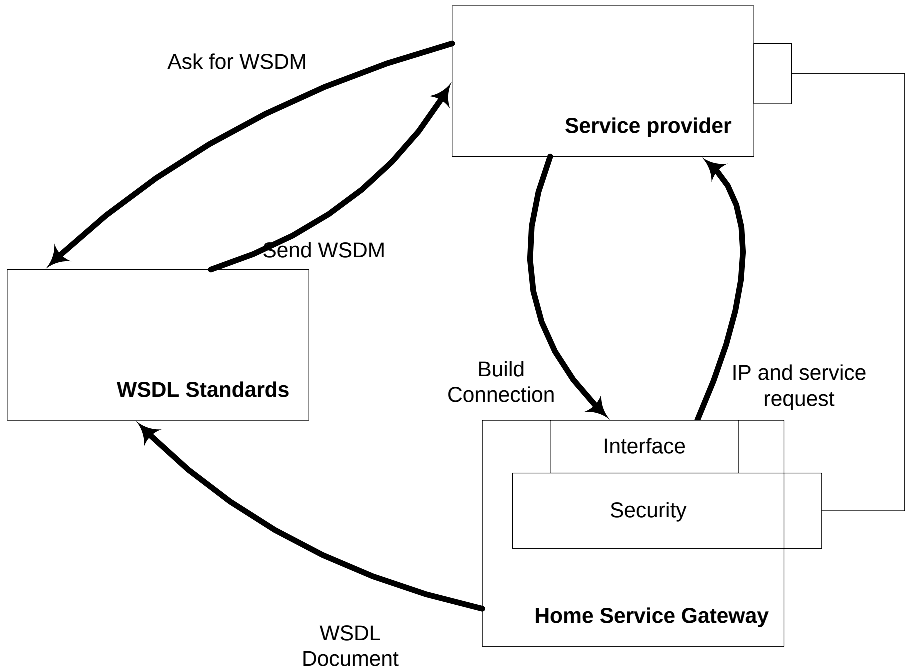

Figure 5.13 Authentication Mechanisms

The communication between the connected home and the remote service provider is point-to-point, with no third party service providers or intermediaries involved.

5.8 Summary

RSOs can be deployed in distributed and integrated routes. In the distributed route, RSOs are deployed at the home service gateway and communicated with connected home devices through proprietary protocol. This approach is suitable for services only involving handful devices. In the integrated route, RSOs are hosted by the devices. Therefore, RSOs can interact with both home service gateway and other service objects. This approach is suitable for services involving complex devices and requiring high security, but is required to keep constant connection between devices and home service gateway.

There are three sets of interfaces need to be considered for RSOs implementation: basic operation interface, event interface and domain specific interface. The basic operation interface is responsible for facilitating the operation from remote service providers to RSOs; the event interface is responsible for publishing a standard format message to interesting parties; the domain interface is responsible for fulfilling specific tasks for individual domain.

Since there are different local networks coexist at home environment, RSOs are designed as a mediator and adapter to convert data and information into a standard format. The RSO implementation adopts and extends URI standard as address mechanism which can help to identify each RSO and simplify RSO message delivery process. The root domain of a connected home must be a URL and can be interpreted by DNS. The other domains are appended behind it following the validate route to access the specific RSO.

RSOs adopt SOAP prototype as message standard and publish/subscribe schema as the communication mechanism. A RSO message object contains two parts: the message data and the message headers. The message header is specified based on SOAP standard to facilitate messages delivery process. The message body carried specific information that is confirmed by XML schema, but must keep untouched during the transportation process.

Service aggregation plays crucial role in remote service provision. There are two levels of service aggregations: device level aggregation and service level aggregation. The device level aggregation is responsible for integrating primitive service resources into high level service objects and enables the interaction with remote service providers. The service level aggregation is responsible for orchestrating relevant service objects into a specific service application to deliver advanced service and convert the service application into a complied representation suitable for execution.

As the cost sensitive nature of connected homes, the proposed home service gateway supports data/information to be submitted to remote service providers, who are responsible for the data storage and utilization. There are two types of deployment structure for service applications: centralized and decentralized. For centralized structure, the service application is deployed and hosted on the remote service providers' platform. This approach is suitable for connected homes which have stable and continue Internet connection. For decentralized structure, the service application is deployed by remote service providers but would be executed on the home service gateway. This approach requires a service execution engine to be developed in the home service gateway.

Chapter 6 Implementation of the Remote Service Delivery Platform and Associated Tools

6.1 Introduction

This chapter describe the implementation of the remote service delivery platform. The implementation is intended to use open source components wherever possible.

Associated supporting tools are developed to facilitate the creation and execution of remote service applications.

6.2 Development Tools and Implementation Strategy

This remote service delivery platform is developed on an IBM open source framework – Eclipse.

6.2.1 Development Tools

The main development tools for the remote service provision are *BPEL designer* installed inside Eclipse development environment at remote service provider's platform.

- *Eclipse* (http://www.eclipse.org) is a multi-language software development platform that comprises an IDE and a plug-in system to extend it. (Vogel, 2009) The platform is written primarily in Java, but by means supporting plug-in written in other languages, such as C, C++, COBOL, Perl, PHP, and others. The runtime system of Eclipse is based on Equinox, an OSGi standard compliant implementation. The difference between Eclipse and some other software platform is that instead of being hard coded, Eclipse employs plug-ins that support all of its functionality on top of the runtime system. The plug-ins architecture of Eclipse provides a consistent feature set on multiple platforms. (Police, 2005)

- *Eclipse BPEL Project* is a related open source project that provides an Eclipse plug-in for the visual development of BPEL processes (Platonov and Papkov, 2008). The BPEL project aims to add comprehensive support to Eclipse for the definition, authoring, editing, deploying, testing and debugging of BPEL processes (http://www.eclipse.org/bpel/). The BPEL project includes five key functional components:

  - **Designer** - a graphical-based designer to edit BPEL processes
  - **Model** – an EMF model to represent the BPEL specification
  - **Validation** - a validator which operates on the model and produces errors and warnings based on the BPEL specification
  - **Runtime framework** - an extensible framework which allows developed BPEL processes to execute at a BPEL engine
  - **Debug** - a debugger which allows the user to step through the execution of a process.

6.2.2 Implementation Strategy

As a research project, the main use of the research demonstrator is not for developing a complete middleware and associated supporting tools that can be prototyped into production without further improvement. Although a service-oriented service delivery platform could engage many emerging technologies, such as semantic web or mobile agents. However, this research is aimed at providing a low cost solution for service generation and delivery, a set of assumptions will be adopted to reduce the complexity of the system, which allow researchers to concentrate on the overall architecture and the characteristic of key components and their interactions.

The implementation will use open source and reuse off-the shelf software wherever possible. Beside the economic concern, open source tend to conform closely to general standards, since standards change but slowly and interchange formats are often particular stable. Also, software developed by open source can allow other parties to access the source code, which provides the possibility to build future development on top (Gbdirect Ltd, 2009). By reusing third party's powerful components, researchers can concentrate on this research study instead of developing software. Not to mention many off-the-shelf software components are more powerful and professional than ones developed by researchers, some of them even available for free downloading. Component reuse is the trend of software development.

6.2.3 Runtime Infrastructure

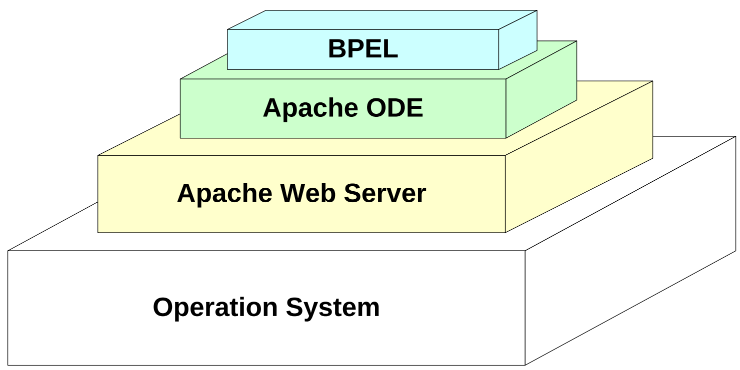

Figure 6.1 Runtime Infrastructure in the Home Service Gateway

- *Apache Tomcat* (http://tomcat.apache.org) is a servlet container developed by Apache Software Foundation (ASF). Tomcat is the official reference implementation of the Java servlet and JavaServer Page (JSP) specification, and provides a "pure Java" HTTP web server environment for Java code to run. It is also the world's most widely used open-source Java web application server (Zeichick 2008).
- *Apache ODE* ("Orchestration Director Engine") (http://ode.apache.org) provides an execution container for service processes written following BPEL standard. (Brown 2007) ODE is responsible for interacting with web services, sending and receiving messages, handling data manipulation and error recovery of service process. It supports

living process execution and orchestrate interoperation between all the involved service objects.

6.3 RSO-compliant Device Implementation and Management

6.3.1 RSO Development Model

The implementation model of RSO can be illustrated in Figure 6.2. It consists of five key functional blocks:

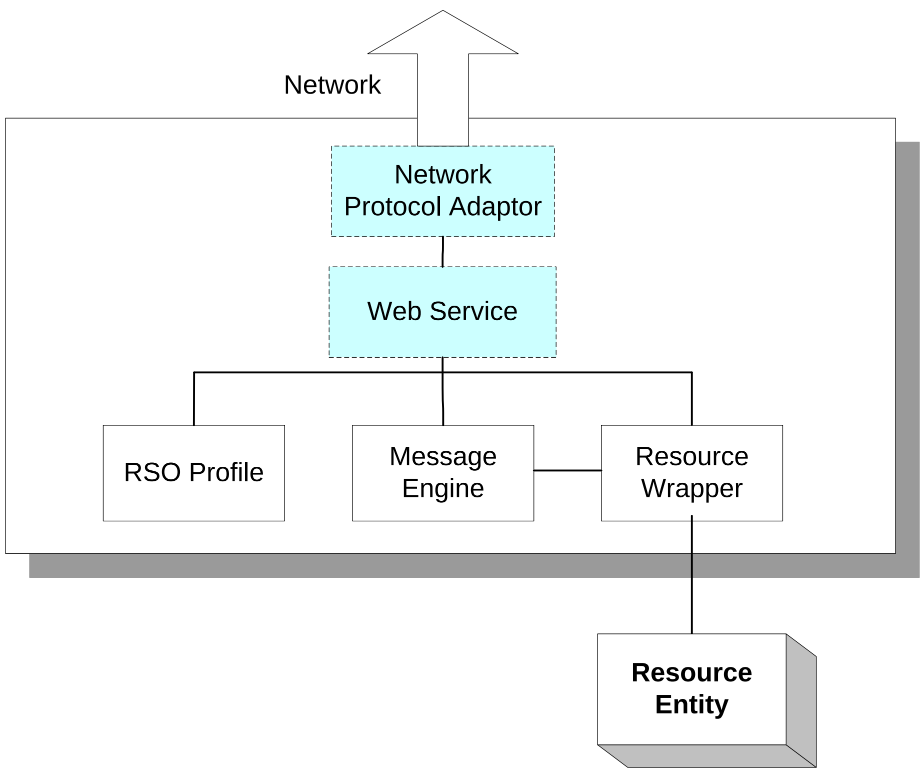

Figure 6.2 RSO Implementation Model

- Resource Wrapper - the resource wrapper control the detail operation of individual resource entity (e.g. device and user profile). The communication mechanism with resource entity could be communication protocol (i.e. RS232) or database operations, which depends on resource type and interface.

- Message Engine - the message engine monitors the statuses of resource entities via resource wrapper and notifies the interested parties about these changes. The “subscription-event” model is the basic operation pattern. It also responds to handle the incoming notification message from other relevant parties, if it is necessary.
- RSO profile - the RSO profile is the RSO description of RSO identity, features, capabilities and functional interfaces. The profile utilizes XML to organize the content, which allows both human and computer can efficiently interpret the meanings of the description. Besides the static description, it could also contain some dynamic information reflecting its usage status (e.g. how many time has it been triggered).
- Web server - the web server is an embedded version server to limited foot print as well as capability for device implementation. It comprises the standard web and web service utilities to process web-based protocols (i.e. HTTP, SOAP) and achieve various RSO operations.
- Network protocol adaptor - the network protocol adaptor responds to adapt various home network protocols and encapsulates the diversity of low-level network operations. Through such abstraction, it provides a single stable and high level interface for subsequent internal operations.

It should be noted that web server and network protocol adaptor are optional, which depends on the RSO deployment strategy. For the integrated deployment, RSO is embedded within resource entities. In this case, it should include all five modules. However, the host of resource entity could lack sufficient computing resource to enable RSO capability or be a legacy system which bids such extensions. The only way to enable the RSO capabilities is via a proxy that is deployed at residential gateway or other capable devices separated from entity host. In this scenario, RSO implementation may leave out web server and network protocol adaptor by utilizing the infrastructure components.

6.3.2 Device Prototypes

The device prototypes developed by De Montfort Mechatronics Research Group can be illustrated in Figure 6.3. Each prototype consists of three active elements: instrumental element, intelligent element and communication element. Instrumental element refers to various existing device (e.g. PIR sensor, blood pressure monitor). A new RSO-compliant device may include more than one off-the-shelf device. For example, combining weight monitor and RFID reader together forms a user-aware weight measurement device to fulfill healthcare service requirement. A generic intelligent module has been developed to host RSO implementation based on the proposed model and convert legacy device into RSO-compliant device. On the one hand, it connects to off-the-shelf device via legacy I/O interface that could be I/O signal or simple protocol such as RS232 or even network protocols such as LON; on the other hand, it uses the communication element to join into home networks. The communication element offers essential communication capability. Currently, two network standards have been supported that are Bluetooth and Ethernet.

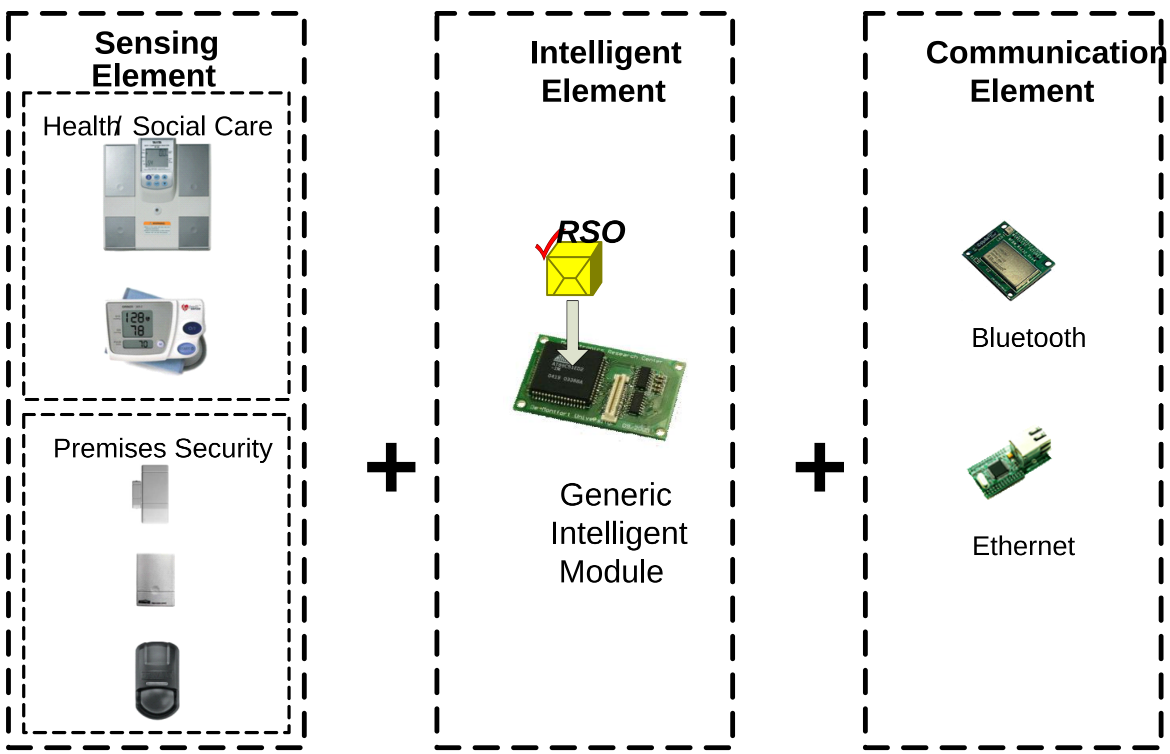

Figure 6.3 RSO-compliant Device Prototype

6.3.3 RSO Registration Process

As demonstrated in Figure 35, when RSO joins the network, it broadcasts a User Datagram Protocol (UDP) packet with the destination address of 255.255.255.255 to the home gateway, in order to discover available DHCP servers. The home gateway administrator forwards the UDP packet to a DHCP server.

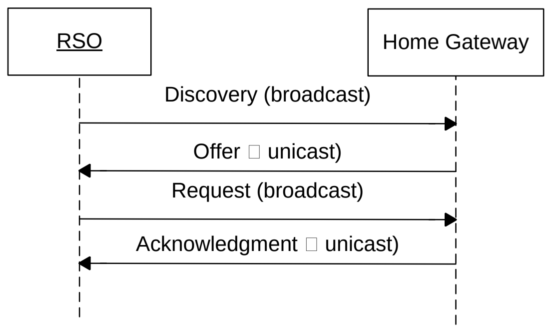

Figure 6.4 RSO Registration Processes

When a DHCP server received an IP lease request from a RSO, it reserves an IP address in the home gateway for the RSO and sends back a DHCPOFFER message to the RSO. This message includes the RSO's MAC address, which is the IP address offered by the home gateway.

After the RSO received the DHCP offer, it will broadcast a DHCP request message. When the DHCP server receives the DHCPREQUEST message from the RSO, the registration processes enters its final stage. The DHCP server will send a DHCPACK packet to the RSO. The packet contains the duration and other configuration information that the RSO might have requested.

6.4 Service Aggregation Development

6.4.1 Device Level Aggregation and Web-based Interface

Web services are the primary means of exposing services to clients, be it in electronic commerce and are being used and integrated with customer-managed applications as well as in complex mashups (Wassermann & Ludwig 2009).

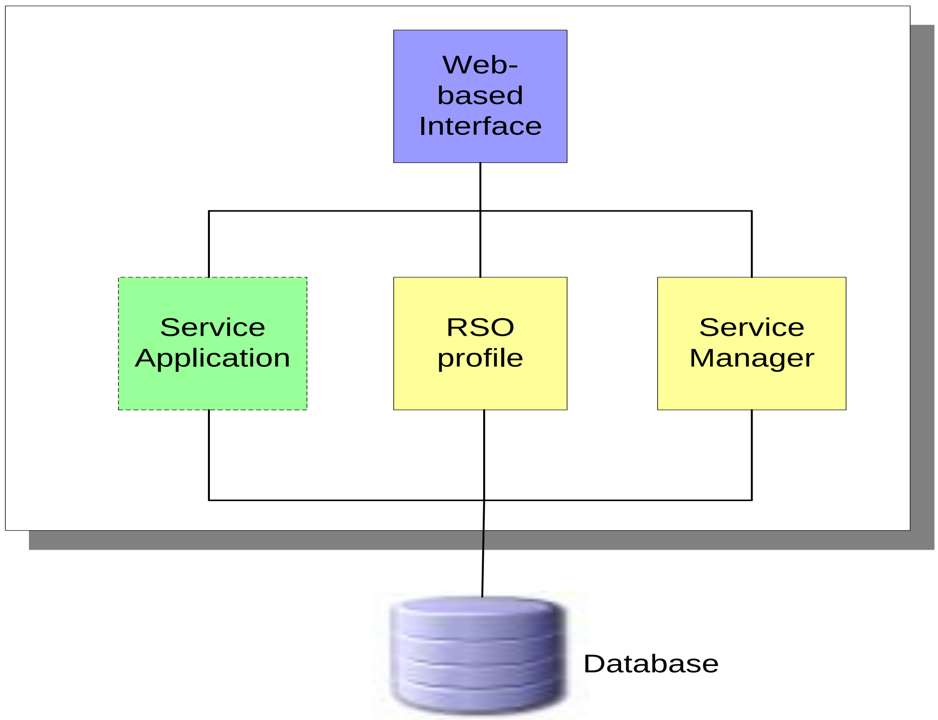

Figure 6.5 Implementation Model of Web Service

Following the trend of distributed service aggregation, a standard web service combining with relation database is built on the home service gateway. As illustrated in Figure 6.6, the web service provides a web browser-based operation interface to ensure the interoperation for operator via Internet; the transaction and management date are stored inside the database. The implementation model is similar to the model of domestic RSO model. There are two types of web service implementation: one type is on only the wrapper of RSO; another type integrated some low level devices (i.g. sensor) into a service application to simplify the service description and delivery. The later model required a *service manager* to active service application and user accounts, verifies identities and grant correct access right. In general, the web service application is to re-wrap the RSO in order to provide a better management and standard interface.

6.4.2 Service Process Structure

The service process template can be displayed in Figure 6.7. The *import* element declares a dependency on external namespace and XML Schema. The *partnerlinks* represent the interaction between the current service process and each of the processes with which it interacts. The *variables* section defines the data variables used by the process. *Variables* allow processes to maintain state between message exchanges.

The processing taking place inside the *Sequence* element consists of several types of activity. The *receive* activity is used to receive message passed on by the event broker. The *invoke* activity is used to invoke an operation on a home service object. The *reply* activity is used to send a result to a predefined service application. The structured activity prescribes the order in which a collection of service activities is executed.

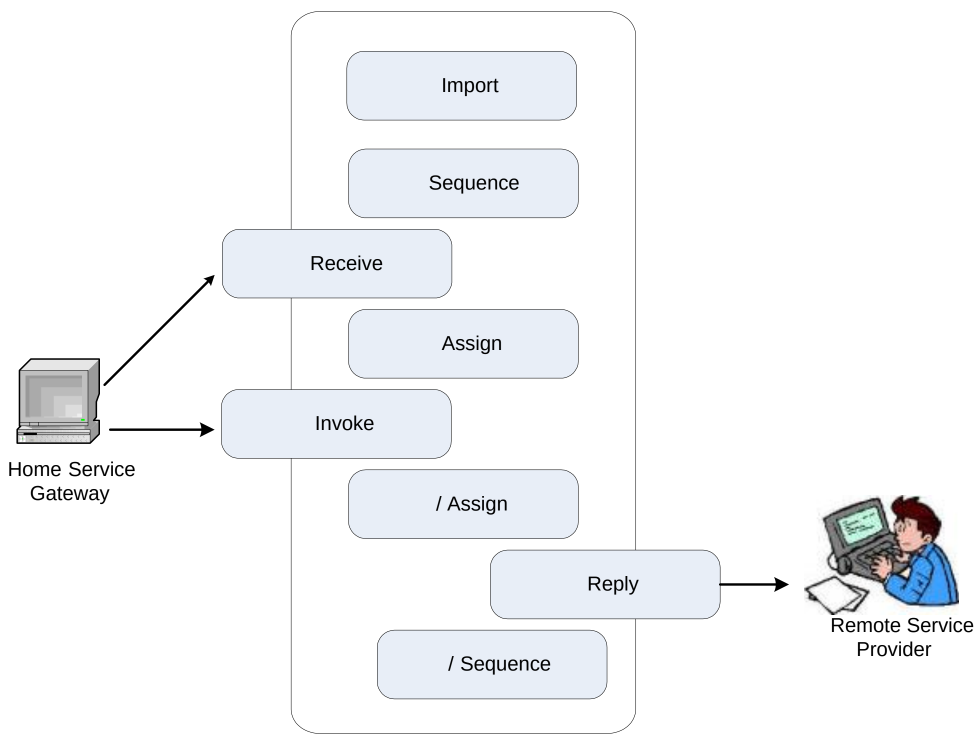

Figure 6.6 Service Process Structure

6.5 Plug Points and Binding Methodologies

The home service gateway provides plug points for RSO-compliant devices and a set of Internet accessible interfaces for service providers to use the available home service resources. A set of binding mechanisms are deployed to make these interaction become possible.

6.5.1 Binding Home Service Resources to Service Application

In this research study, the service application is responsible for constructing the relevant home service resources' functionalities into a specific task. The service application is described as a composition of an abstract service. Each service application could include more than one service processes. As shown in Figure 6.7, Instead of refer to a concrete service, only a logic name, such as "*Window Shock Detector*" or "*Hall Motion Detector*" and the service interface are used.

The RSO's WSDL interfaces could be viewed from an "abstract" section and a "concrete" section. The "abstract" section of the WSDL defines the interface and portType; while as the "concrete" section of the WSDL defines the binding and endpoint. The remote service provider would specify that RSO endpoints are bound to the binding points of service processes. The portTypes inside service processes point to the WSDL portTypes with the abstract interface of a RSO. Hereby, when a service application is deployed, the WSDL portTypes that make up the RSO interfaces is consumed by the service application. Alternatively, the RSO is given the address where the operations or the process itself are provided.

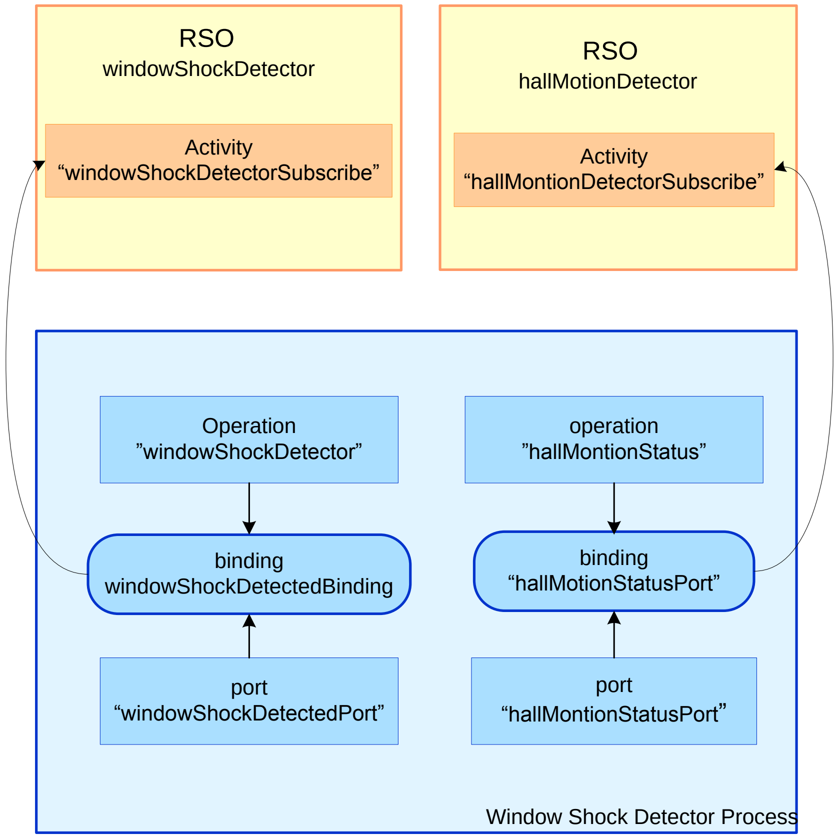

Figure 6.7 Binding a service process to RSOs

6.5.2 Invoking Service Applications from Home Service Resources

The service process starts with a specific activity, the invoke activity, that allows the service process to become active by a RSO message. For example, as shown in Figure 6.8, the subscription of a RSO message results in the triggering of a service process instance to handle this activity. When the message starts a service process, a service application space is allocated, within which the created service instance as well as subsequent dependent instances will execute. The proposed architecture adopts the *publish-subscribe* paradigm, in which a RSO that generate event publishes the type of events to the assigned service application. Multiple RSOs can notify events take place in one service application.

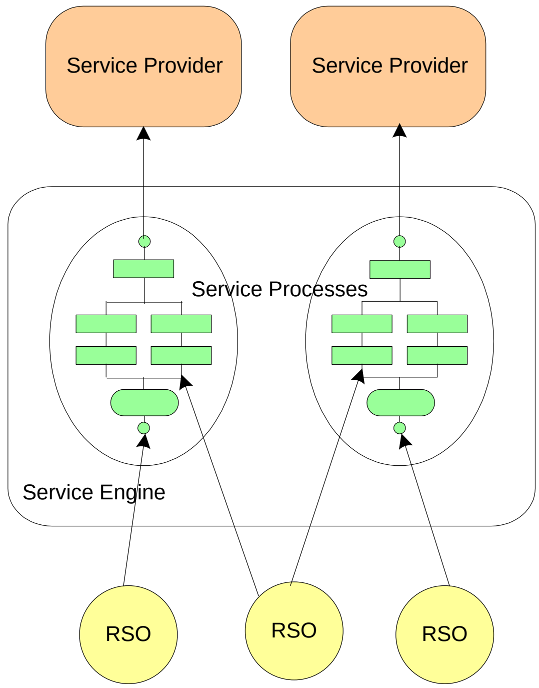

Figure 6.8 Invoking Service Process by RSOs

A service process instance has completed normally if the service logic of the process could be executed successfully. However, if the process instance is terminated explicitly, or if a fault reaches the overall fault handler of the process, the completion is considered abnormal.

6.6 Summary

The development of remote service delivery platform and associated tools is based on Eclipse, an open source software development platform followed OSGi standard. The runtime system comprises an IDE and a plug-in system that support flexible extension of its functionalities on top of the platform. Eclipse BPEL is plug-in to provide an environment for a visual developing, editing, testing and debugging of BPEL service processes.

Apache Tomcat is used as a HTTP web server environment for the home service gateway. Apache ODE is used as an execution container for service processes written by BPEL standard. Apache ODE is responsible for interacting with web services, sending and receiving event message, handling data manipulation error recovery of service processes.

The implementation model of RSOs consists of five key functional blocks: resource wrapper, which control the detail operation of individual service resource entity; message engine, which monitors the status of change of service resource entity and notifies the interested parties; RSO profile, which describe RSOs identity, features, capabilities and functional interfaces; web server, which is an embedded server with limited web service capability; network protocol adaptor, which responses to convert different local network protocols into standard interfaces for interoperation.

A standard web service combined with relation database is deployed on the home service gateway. There are two types of web service implementation: one is only wrapped an individual RSO, and another integrates some low level devices into a service application. Remote service providers can apply subscription of data through a web browser based interface. The transaction and management data are stored inside the database which will be used to direct notification messages to interested parties.

Chapter 7 Test Cases

7.1 Introduction

Test cases have been devised to evaluate the methodologies, designs and toolset, and to demonstrate the potential benefits and viability of this research undertaken. The test case approach can be summarised as follows:

- Use of the methodology and associated tools to generate domestic services which support the “On-Demand” business model with concepts of “Choice” and “Mass Customization”.
- Use of home service gateway to verify the service execution process and to demonstrate its capabilities.
- Evaluate test results.

Two classes of service, namely remote security monitoring and an advanced information service have developed to facilitate the evaluation and demonstration processes. A demonstrator has also developed supporting these processes.

7.2 Overview of the Demonstrator

The demonstrator (or test bed) shown in Figure 7.1 is developed to demonstrate the remote service delivery to the connected home, with the ability to support service “Mass Customization” and provide “Freedom of Choice” to both service providers and recipients. The test bed consists of five RSO-compliant sensors representing various types of intrusion detection devices (e.g. PIR motion detector and shock detector), which use different communication technology such as Bluetooth and wired Ethernet to simulate a heterogeneous networked environment.

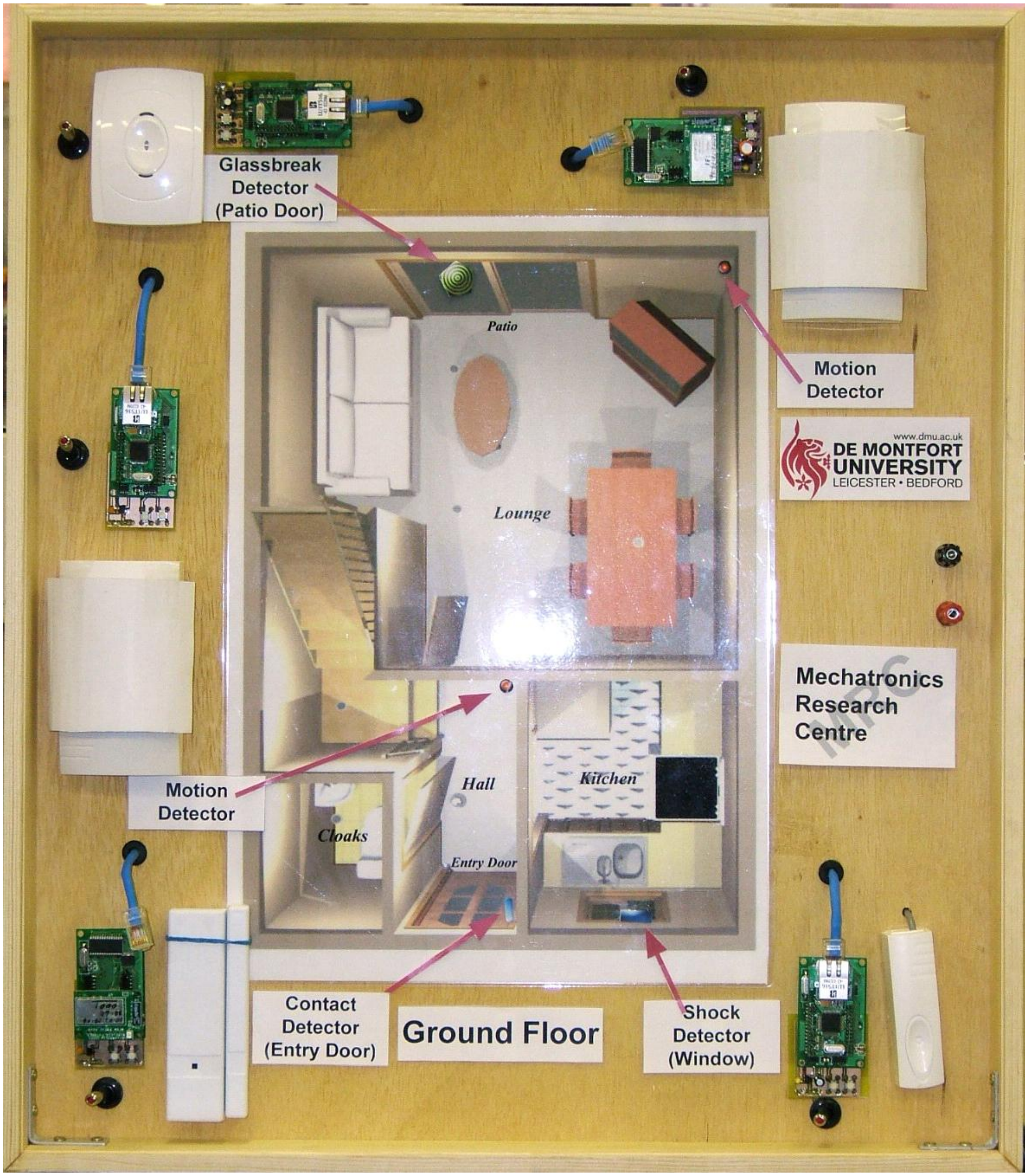

Figure 7.1 Project Demonstrator

As shown in Figure 7.2, two computers are used for demonstration. One computer is used as a home service gateway on which the proposed service delivery platform is developed. The home service gateway is responsible to manage the registration and life cycle of RSO-composed sensors, and delivery services from remote service providers to RSO-composed sensors. Another computer is used as remote service provider, which is responsible to retrieve data and information from home service gateway and delivers advanced services according to the predefined service contracts.

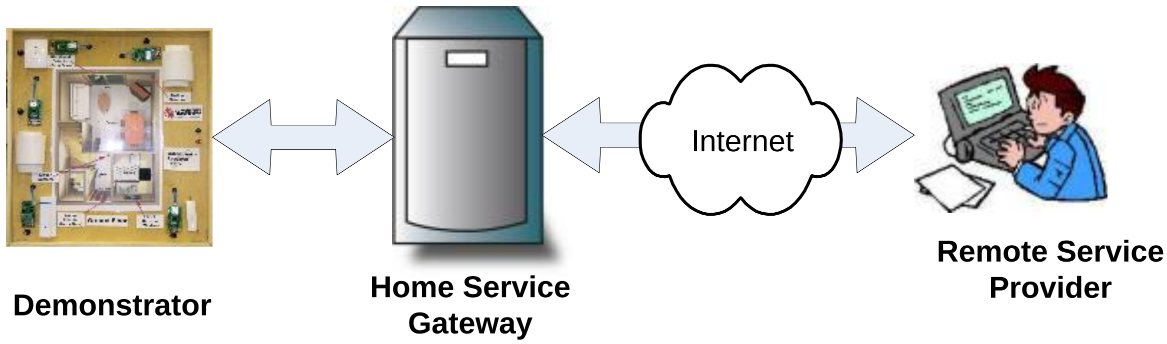

Figure 7.2 Overview of Demonstrator Components

Two different types of service scenarios are designed, namely remote security monitoring service and an information service, are devised for supporting the research study and also act as a test bed to evaluate the proposed methodologies and associated tools for the realization of remote service provision that would support service “Mass Customization” and provide “Freedom of Choice” to both service providers and recipients.

7.2.1 Overview of Remote Service Provision Model

“eService-On-Demand” is an operating model pioneered by the Mechatronics Research Centre at De Montfort University for supporting remote services provision. “eService” is aimed to deliver various types of services such as home security, healthcare and social-care to connected homes remotely through the advanced use of ICT. Remote delivery is seen as the key in delivering quality services in an economical and cost effective manner. The key ingredients are remote delivery and “Pay-as-You-Need”. The essence of this operating model is captured in Figure 7.3.

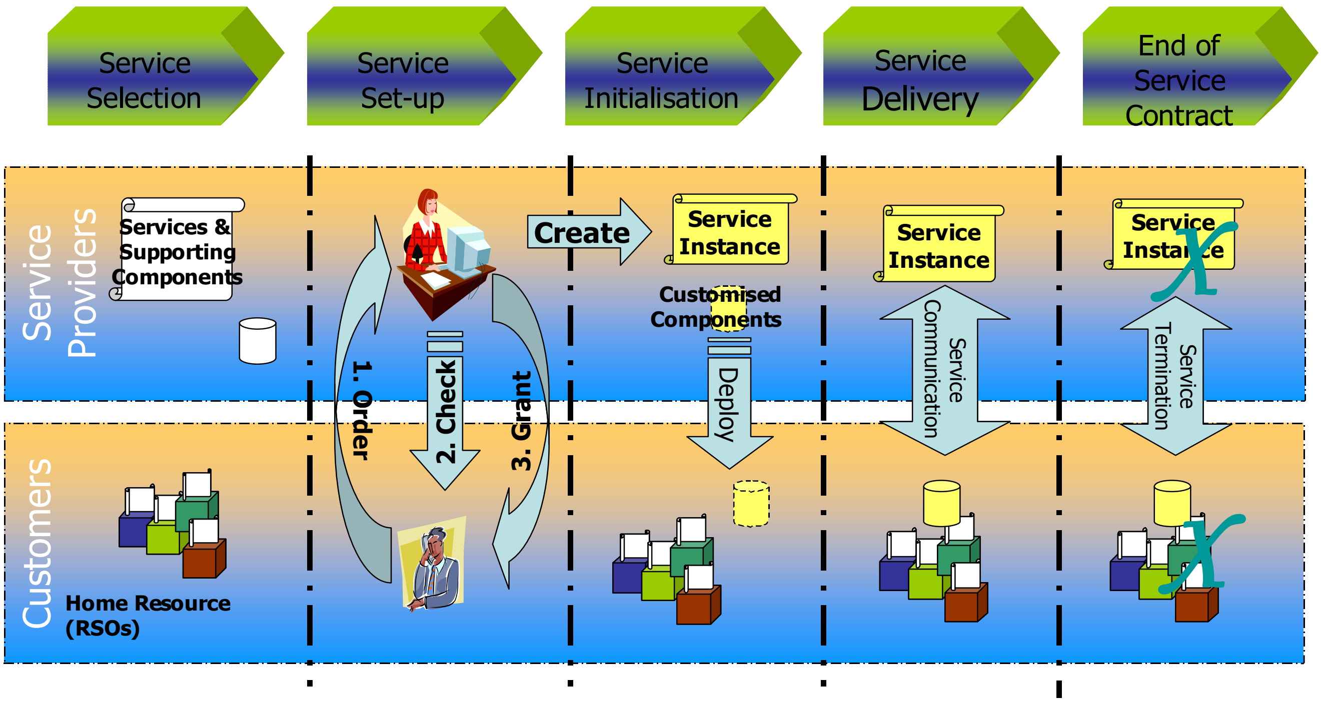

Figure 7.3 eService-On-Demand Process Diagram

7.2.2 Test Scope

Domestic property security and information services are used as the scenarios to test, evaluate the unique features and capabilities of the open remote service delivery architecture that has been devised. The test require can summarised as follows:

- Building RSO-compliant devices using the proposed protocol and methodology; test the service resources discovery and life-cycle management of RSOs

  Mechatronics Research Centre has developed a generic micro-controller based hardware module that allows a developer to customise its functionality via custom embedded software. This project employs this generic intelligent module to transform the selected "off-the-shelf" devices (i.e. different types of sensor) to fulfil the requirements of RSO. Therefore, the transformed devices not only become interoperable-networked devices that are independent from their communication transportation protocols (e.g. TCP/IP and Bluetooth), but also presented as a standard service resources regardless of their hardware specific details via encapsulation. Tests will be designed and conducted to evaluate and demonstrate service resources discovery and life-cycle management of the service resources.

- Creating advanced service using the proposed methodology and developed tools

  The aim and key objectives of this research study is to develop an open service delivery platform which would support “On-Demand” remote service provision with the “Freedom of Choice” to both service providers and recipients. Tests will be designed and conducted to evaluate and illustrate the required capabilities of the service delivery platform such as enable the remote service provider to acquire the knowledge and information of the available connected home service resources automatically, and customise the delivered service application to utilise the available connected home service resources to meet the specific service contract regardless the difference of the home environment settings (i.e. Mass Customisation). The performance of the generated service application will not carry out in the developed test cases because it is difficult to setup objective criteria to measure the performance such as quality of service.

- Employ the remote service delivery platform and prototype tools to create different service applications to demonstrate the concept of “Resources Sharing” within a heterogeneous networked home environment.

  One of the objectives of this research study is to develop an open and general service delivery platform which supports remote service provision of different service applications and allowing them to share the available connected home service resources. Tests will be designed and conducted to prove the viability of “Resources Sharing” between service applications regardless of their application domains using the methodologies and tools.

7.2.3 Test Approach

The test approach is illustrated in Figure 7.4:

- Using the proposed service provision methodology and associated prototype tools to build all the essential components of the project demonstrator such as RSOs for sensor devices, service delivery platform that residing in the home service gateway and service applications.
- Using the test case scenarios to conduct all the tests that are designed for the assessments such as “Mass Customization”, “Resources Sharing” and “Freedom of Choice”.
- Comparing the test results against the expected results and evaluates the viability of the proposed methodologies and developed tools.

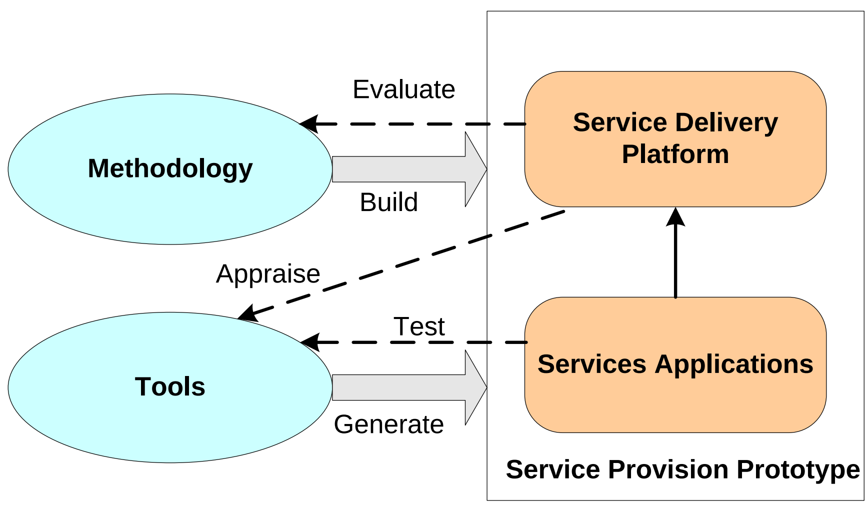

Figure 7.4 Test Approach

7.2.4 Expected Results and Test Criteria

The expected results are discussed as follows:

- The information related to the service(s) offered by the developed RSO-compliant sensors should be available for their service consumer such as service applications. Such information would also available to user through the home service gateway.
- The remote service provider should be able to utilize the service(s) offered by the registered RSOs to create the required service application(s) through the developed method and mechanisms.
- The RSO published events should be able to trigger its subscribers (e.g. service process) residing either locally (i.e. home service gateway) or remotely (i.e. remote service provider systems).
- The service process should be able to execute according to the predefined sequence (including exception handling) and send out the expected result(s) to the service provider.
- More than one type of services should be able to provide through the developed remote service delivery platform.
- Service resources should be able to share between different service applications.

7.2.5 Assumptions and Constraints

Some assumptions and constraints for the design of test cases are described as follows:

- Although two test case scenarios have been constructed on the developed SDP, only one scenario (i.e. remote security monitoring service) was implemented as standalone service application due to the limitation of the project demonstrator. Therefore, another scenario (i.e. information service) does not include standalone service application logic and is only developed as an advanced service object for supporting the service provision of other service applications.
- Although security of ICT is an essential component for any type of remote service delivery platform, it is not the focus of this research study. The required development of the security component within the proposed home service gateway architecture is assumed to be considered in other research studies. The developed home service gateway prototype for this research study has adopted one of the current advanced secure

communication methods to maintain the safety of communication and security in accessing the connected home environments.

- The home service gateway prototype is currently built on a Personal Computer (PC) for ease of implementation; another PC will be used to simulate the remote service provider. Although BPEL is chosen as an enabling standard for the development of the SDP prototype, the majority of existing BPEL engines only support centralised architecture for business systems which are too complicated and expensive for connected home environments. Therefore, a lightweight approach, which an embeddable BPEL engine – Apache ODE is adopted to facilitate this research study and supporting the development of the demonstrator, although interfaces of associated tools are not very user friendly.

7.3 Test Case One: Remote Security Monitoring Service

This test case is using domestic home security application to evaluate and demonstrate the concepts behind this research study.

- A household can order a short-term security service contract (e.g. two weeks) during family holidays. The security service provider will link up the home security system with their monitoring centre via the Internet to provide a "virtual" patrolling service and other related security services.

During the service provision lifecycle, most differentiating and challenge part is "On-Demand" subscription process illustrated in Figure 7.5. It highlights the essential requirements to achieve "On-Demand" eService delivery, which are remote inquiry of home characteristics and immediate service delivery. It also implies the necessary of security scheme to establish mutual trust between end-user and service provider.

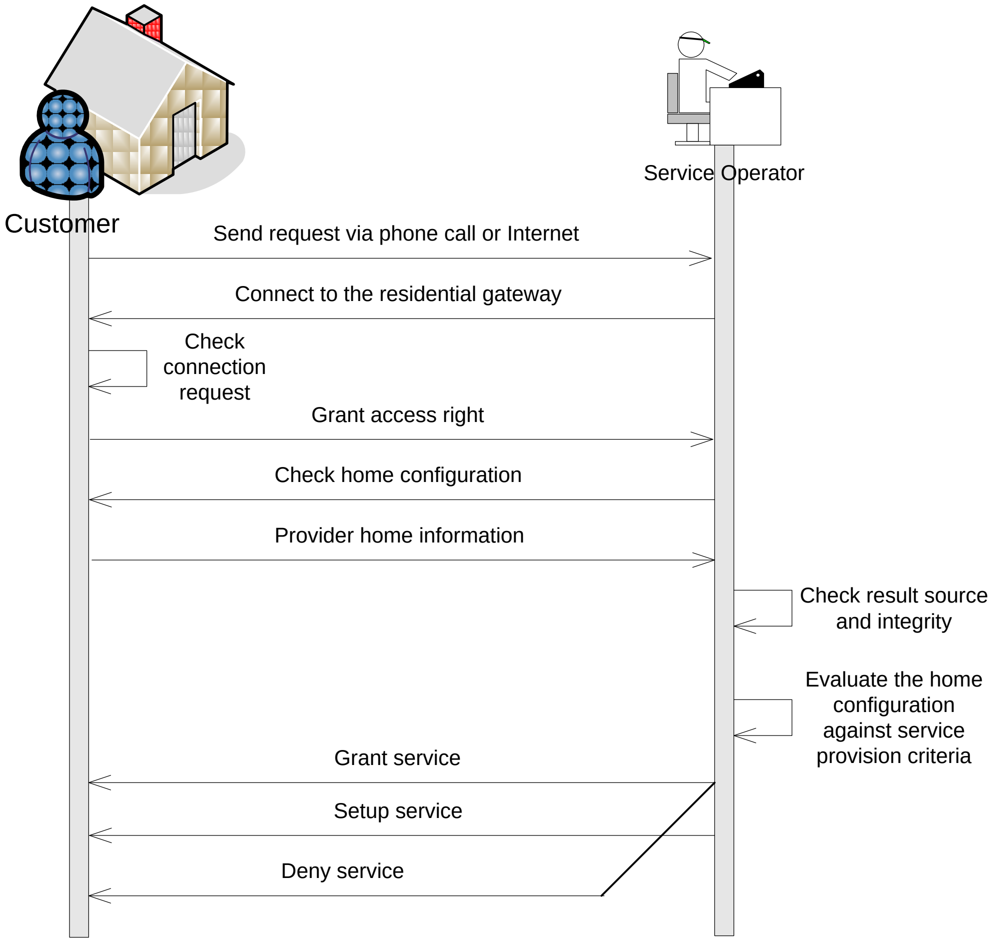

Figure 7.5 On-Demand Subscription Diagram

This test case included the follows:

1) Analyze the RSOs information housing in the home service gateway prototype via web browser to check the viability of methodologies in service resources management, including service discovery and life-cycle management.
2) Design a remote security monitoring service application prototype that would utilize the exposed RSO resources.
3) Use the developed home service gateway to create a service.
4) Execute the service sequence with the aid of the project demonstrator.
5) Evaluate the proposed methodologies associated tools.

7.3.1 Access of RSO Resources in the Home Service Gateway

The criteria of the evaluation are focused on the requirements of RSO in this research study are listed as follows:

1) User interface
2) Security management
3) Data description
4) Data acquisition
5) Home layout and resources relationship

7.3.1.1 RSO Specification

Figure 7.6 is one of the interfaces which designed for supporting the installation process of RSOS. It shows the essential information of a RSO-compliant sensor exposed in the developed home service gateway. The adopted *Domain Address* scheme is a URI basis address for the sensor. This address scheme gives a unique address for each connected device, appliances and systems that can be used for binding with the service process. The *Category*, *type* and *Model* elements describe the important information related to the sensor; *target* element could be used to describe the physical location of the sensor within the home environment. The *Home layout* is also shown in the RSO management application. *Start* element indicates the installation date. *Period* element is used for device life-cycle management.

Item Name		Content
Object Name		Shockdetector
Domain Address		http://192.168.1.19:2165
RSO Identification		401c9e35-87ef-4320-b82c-ad8ac949ac97
RSO Type	Category	Security
	Type	SecurityEquipment.Shockdetector.IdentifiableShockdetector
Provider	Name	Oregon_Scientific
	Model	Digital Shockdetector
	Serial Number	273adf9463
Features	identifiability	true
	owner	
	target	
Warranty	Start	2009-10-23T14:34:10
	Period	3
	Unit	Year

Figure 7.6 Web Browser Information for RSO-compliant Sensor

7.3.1.2 Security Management

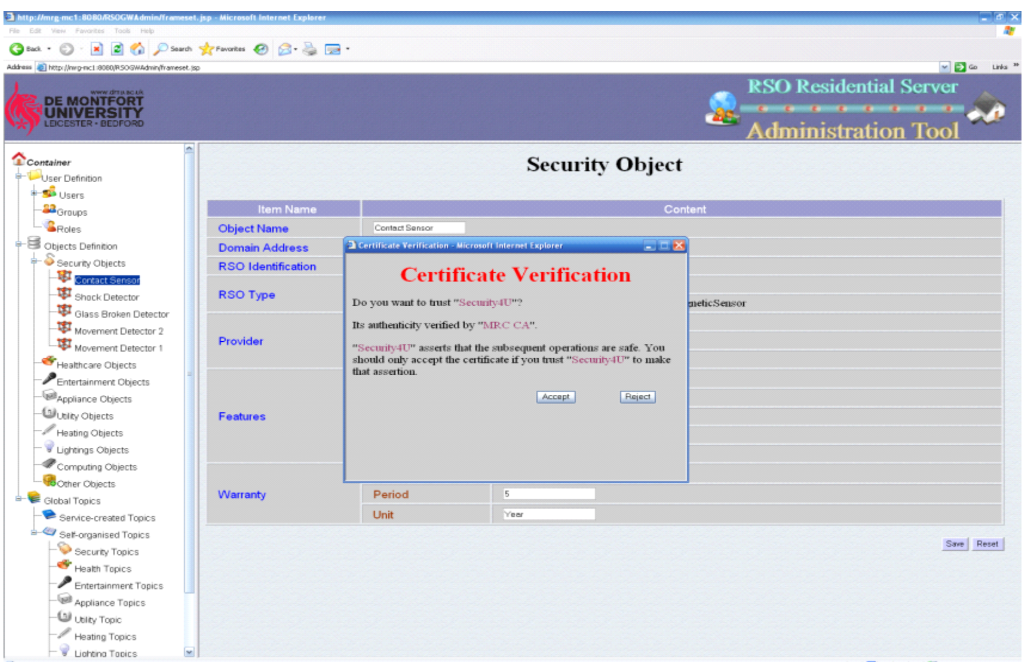

Figure 7.7 Service Provider Certification Verification

The service application prototype utilises WS-Security scheme associated with digital signature exchange to ensure the sender identity and message integrity. Example digital signature is illustrated in Figure 7.7. When the remote service provider sends the inquiry request with its digital signature to the home service gateway, the home service gateway will immediately pop up a notification window to inform householder and ask for permission of access. After the access request has been granted and returns to the remote service provider, the householder identify (i.e. Digital signature) is also attached with the granted message to proof the information source and its integrity.

7.3.1.3 RSO Messaging

The RSO only supports publish-subscribe mechanism for message exchange at this stage. As shown in Figure 7.8, the RSO provides an Interface for its subscribers. The service application logic specifies the specific RSO message(s) that would trigger the relevant service process and also responsible to pass the URI address of the service process to the RSO. When an event arises, the RSO will send a message to the home service gateway, which will then route it to the assigned service process. Each RSO is able to manage multiple subscriptions.

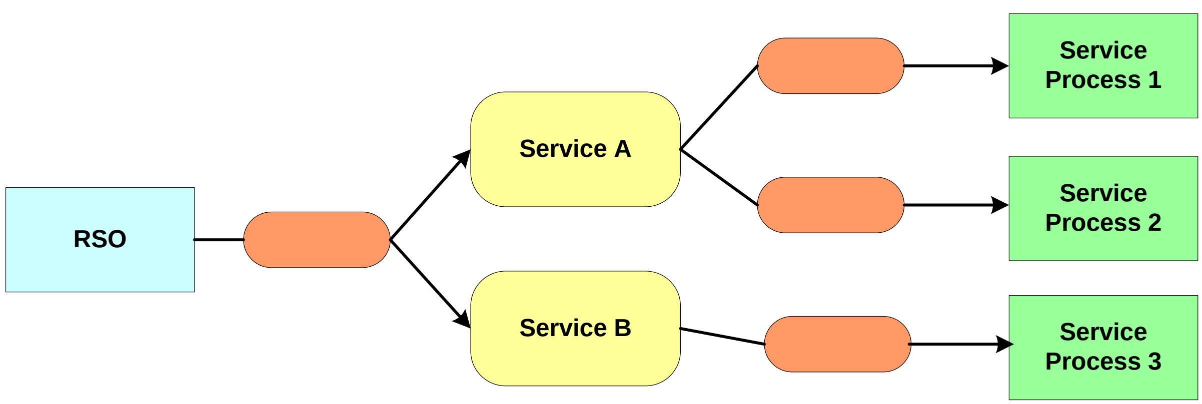

Figure 7.8 RSO Messaging Mechanism

7.3.1.4 Home Layout and Device Location

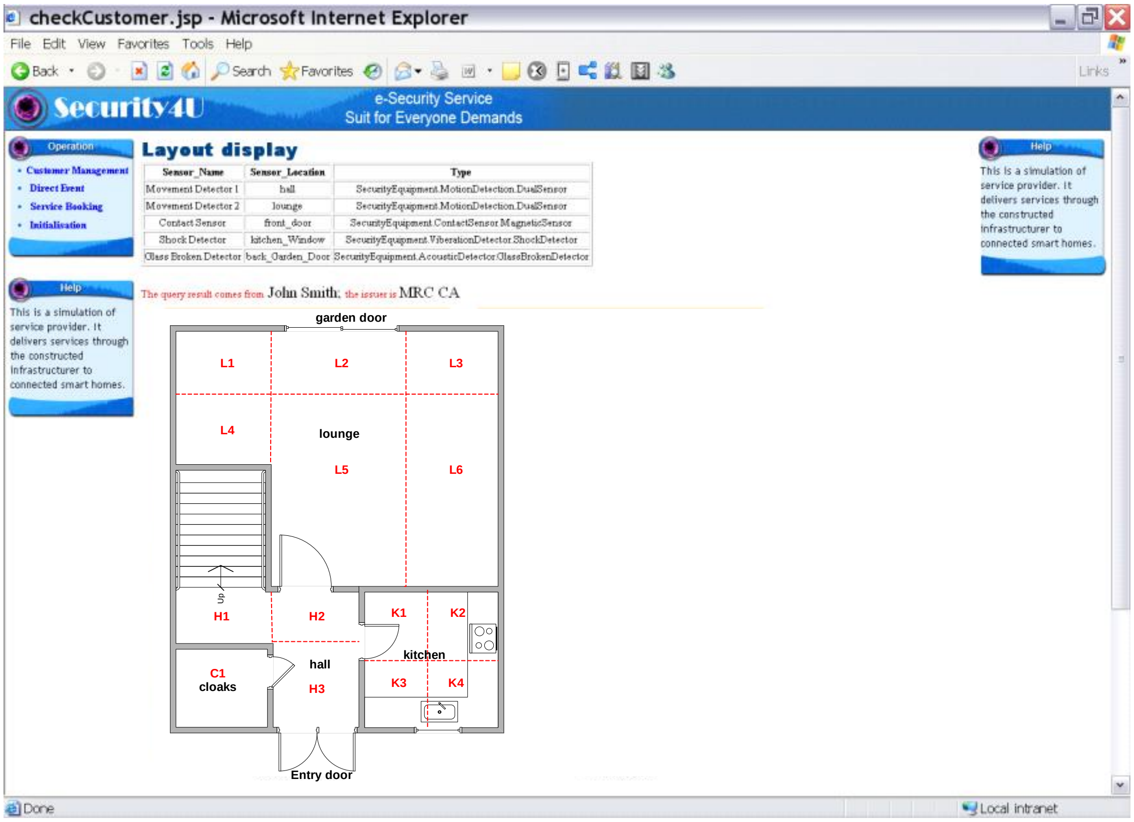

Figure 7.9 A screenshot of on-line home layout

As shown in Figure 7.9, the developed home service gateway prototype provides a home layout drawing through the RSO management application. The home layout only provides the necessary information for supporting service provision and hides all the unnecessary details to protect the privacy of householders. However, there is no well accepted ontology model (or dictionary) which can clearly describe devices and their physical location within the home environment. To help remote service operator acquire better understanding of the home environment and available connected home service resources, the adopted approach divides a single room into several sections. For example, the location of *Window Shock Detector* is K4, and the *Hall Motion Detector* is H2. Together with the RSO information shown in Figure 8.3, it would be enough for the remote service provider to acquire the necessary understanding of the home settings and construct a suitable service application.

7.3.1.5 Evaluation

The evaluation on the requirements of RSO is summarised as follows:

1) User interface for RSO has proved to be reasonably well structured via the opinions of other researchers and visitors who viewed the demonstration. The remote service provider can acquire the knowledge about available connected service resources and their brief location from the RSO administration application that residing in the home service gateway.
2) The home service gateway uses WS-Security scheme associated with digital signature to ensure the secure access and message integrity.
3) The data/information exchange mechanism has proved to satisfy the requirement of data/information transformation between RSO and the remote service provider via the demonstrator and test case scenario although it only supports the publish-subscribe mechanism.
4) Although the proposed location description approach is not a sophisticated solution in terms of automation and intelligence. This additional information has proved to be vital and essential to facilitate the understandings of the home settings and the device locations.

7.3.2 Using available RSOs to Create a Remote Security Monitoring Service

The remote security monitoring service is established with the developed home service gateway prototype. As illustrated in Figure 7.10, RSO-compliant sensors have been registered their service(s) to the home service gateway and exposed them to the remote service provider via web-based standards as a web service application. The remote service provider would use the information supplied by the home service gateway to analyse the available connected home service resources and their locations within the connected home environment. The information is not only used to assess whether the required service resources are available, but also for supporting the decision on level of service(s) that could be offered to the connected home.

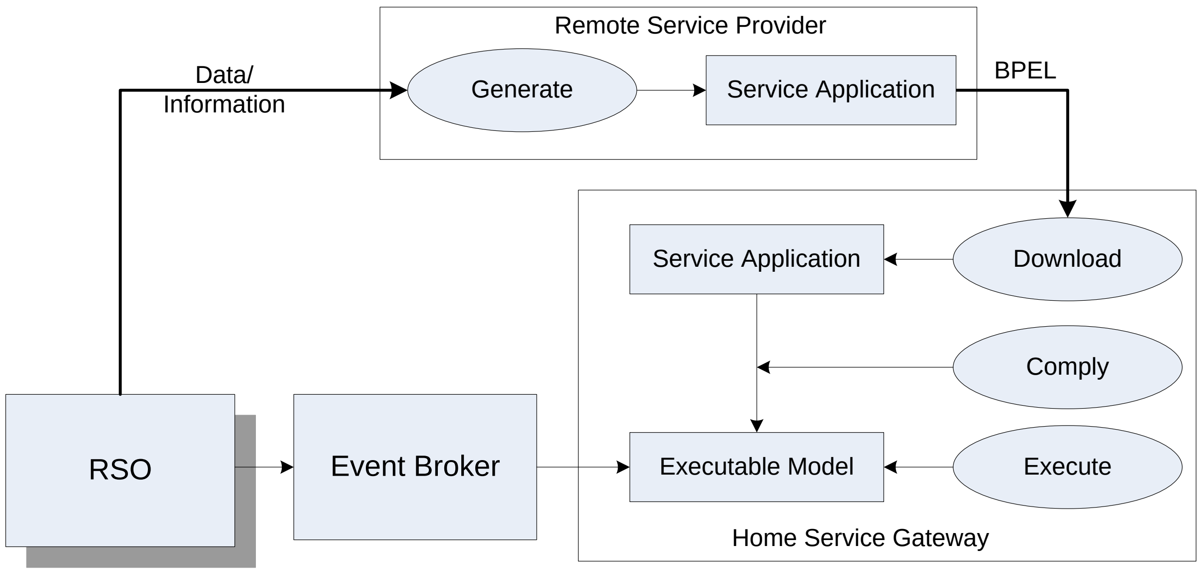

Figure 7.10 Service generation and delivery model

The service application adopts the BPEL standard, which is a widely accepted format for integrating and assembling Web Services. As most of the commercial BPEL engines are too complicated and expensive for the connected home environment, this test case adopts a light weight open source BPEL engine – Apache ODE, which could be embedded into the home service gateway. The remote service provider will use the BPEL designer (part of the BPEL bundle) to create a service application and underlying service processes; employ the simulation environment to test and debug the service process; and ultimately deliver service to the connected home environment.

The service process will be installed inside the execution engine (i.e. Apache ODE) at the remote service provider platform when the service generation process is completed. The Apache ODE is responsible to transform these XML based service processes into executable process which will trigger by the published RSO events through Event Broker.

7.3.2.1 Overview of Remote Service Monitoring Service

With the support of the home service gateway prototype, householders can order a short-term security service during their family holiday (e.g. two weeks). The remote security service provider will acquire the information of the available connected service resources in the connected home environment via electronic means; using the acquired information to determine the level of security service that could be offered; customise the required service application(s) and provide service through the home service gateway if both parties agree the terms of the service contract (e.g. remote security monitored service).

Instead of sending out alarm event to the remote monitoring centre immediately when any of intrusion detection sensors have been triggered, the security monitoring service application will forward the event to a service event verification process for further verification. For example, when the kitchen *Window Shock Detector* sends out a “Shocked” message, it may only be caused by a cat from neighbour jumping on the windowsill. To reduce the possibility of false alarm, an event verification process will check the status of the *Hall Monitor Detector*. If the statue of the *Hall Monitor Detector* is also changed within defined period of time, it is likely the intrusion event is genuine. An alarm message with relevant information will be forward to the remote security monitoring centre and might even alert the householder via mobile phone. Simultaneously, the burglar alarm will be triggered to raise the attention of the nearby and might even warn the intruder with pre-recorded message. The benefit of such information fusion is that the service application can be easily realised and customised to meet the individual needs regardless the difference of home settings with the support of the developed home service gateway prototype.

7.3.2.2 Work Flows of the Remote Security Monitoring Service Application

There are three service processes within this remote security monitoring service application: *Window Shock Detector* process, *Door Contact Detector* process, and *Patio Door Galssbreak* process. Their work flows are illustrated in Figure 7.11. Five RSO-compliant sensors have involved in these services. Among them, the *Hall Motion Detector sensor* has been used in two of these service processes.

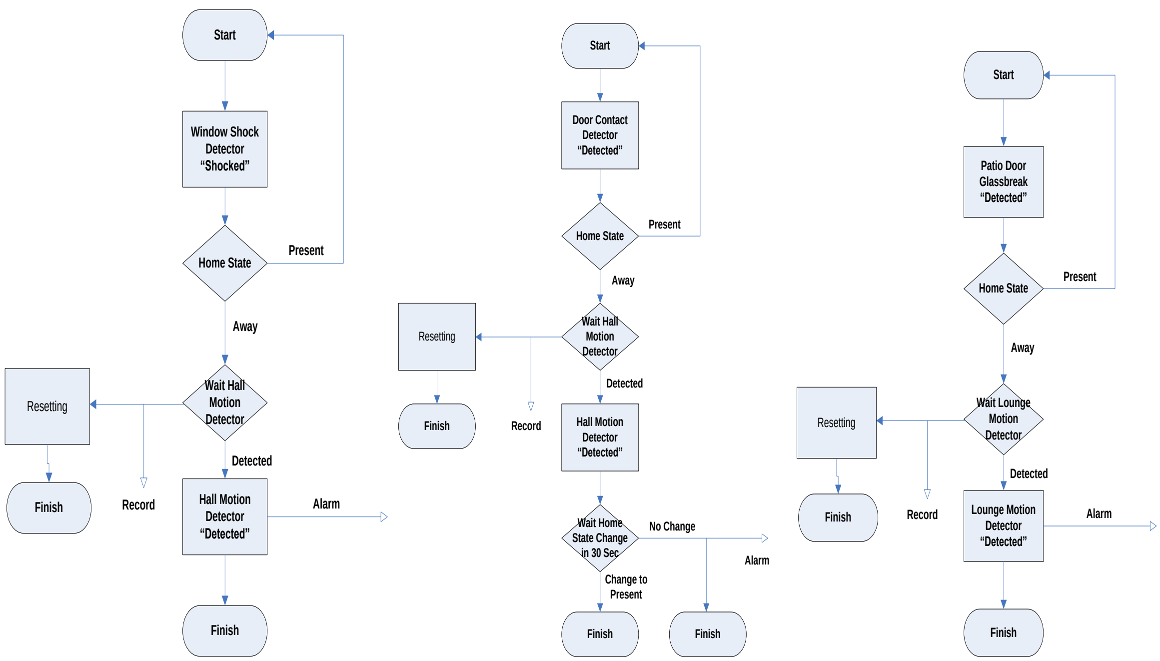

Figure 7.11 Work Flows of Service Processes

7.3.2.3 Remote Security Monitoring Service Creation

Inside the service application, .bpel documents are used to describe the order in which the operation of the RSOs' activities sequence; and .wsdl documents are used to describe the interface of the service process itself – the message that it accepts and returns, operations that are supported, and so on.

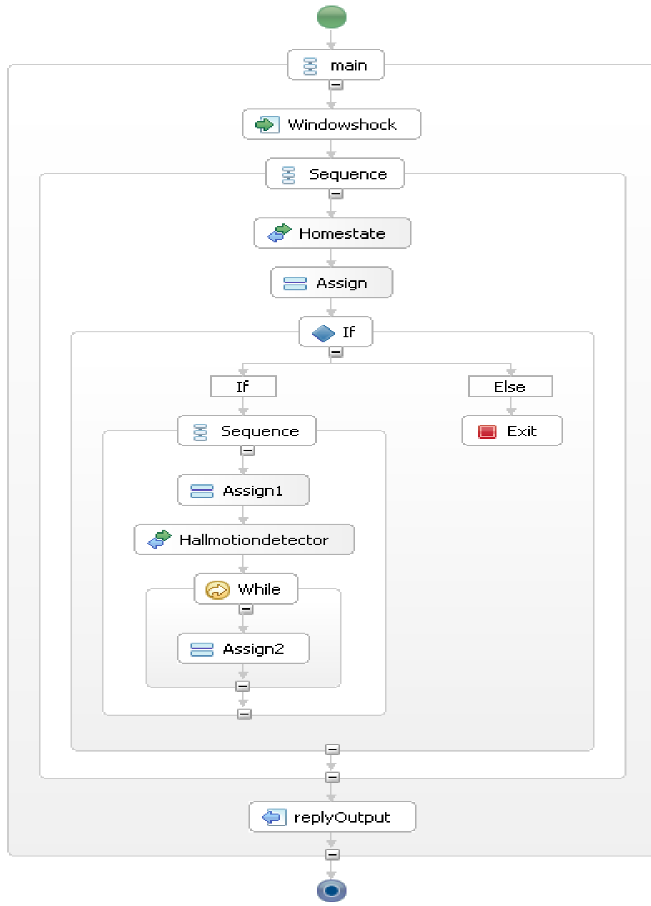

Figure 7.12 Window Shock Detector Service Process

Figure 7.12 is a representation of the *Window Shock Detector Service Process* in a *BPEL designer* developed to simulate the remote service provider. As discussed in Section 8.3.3.1, this service process is used to verify if the "Shocked" activity detected by the kitchen *Window Shock Detector* is a genuine intrusion. By checking with the status of change of *Hall Motion Detector*, the service process is able to provide a more reliable alarm message to the remote security monitoring service provider. The service process has three ports, namely *Window Shock Detector*, *Hall Motion Detector*, and *Remote Security Monitored Service*, each requires to be bind with those pre-known Web services applications.

The .wsdl document provides a mechanism to bind Web services applications to these ports. As shown in Figure 7.13, a binding associated tool inside *BPEL Designer* could bind a port to a URI address, which could be either a RSO inside the connected home or a service on the remote service provider's platform.

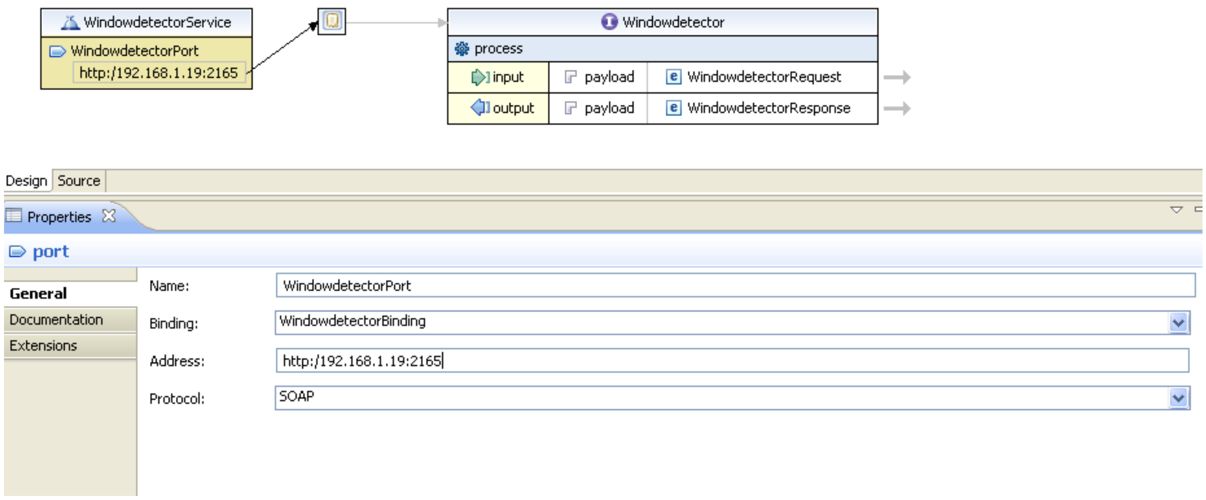

Figure 7.13 Web Service Binding Interface

7.3.2.4 Service Process Simulation and RSO Subscription

The remote service provider has to test every flow path and condition to ensure the quality and performance of the service application before its deployment. The developed *BPEL Designer* provides a simulation environment allowing every service process to be thoroughly tested. As illustrated in Figure 7.14, service providers can provide a variety of return message for each service process, permitting the simulation of an infinite number of fault handling scenarios.

Remote service provider could apply RSO subscription through its exposed interface. The service provider sends the subscription requirement with the URI address of each port in the home service gateway to the corresponding RSO. The complied service application will then be installed into the ODE that embedded in the home service gateway. Therefore, when an event arises, the developed home service gateway is able to forward the message to the assigned service process and execute according to the predefined sequence and send out the expected result(s) to the remote security monitoring service provider.

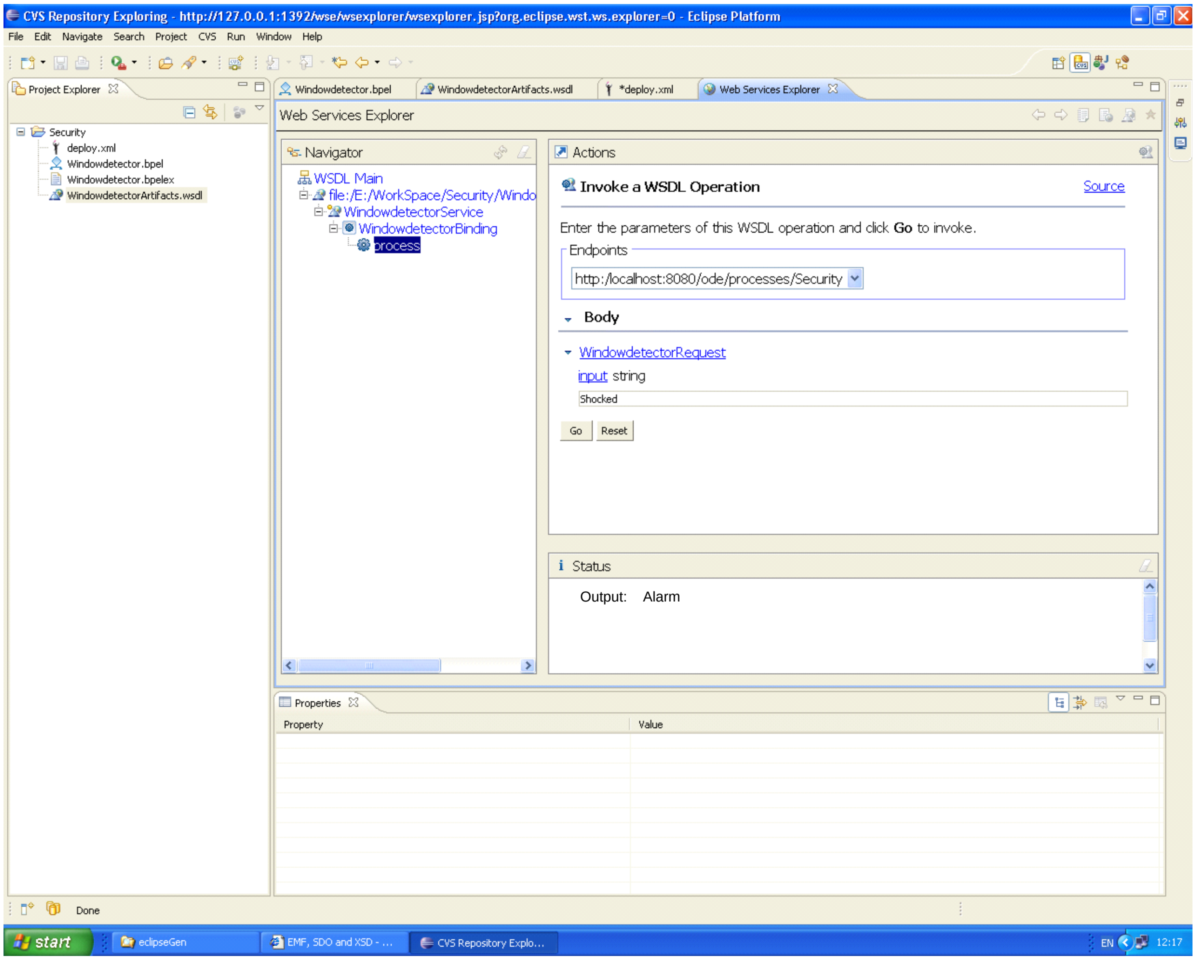

Figure 7.14 Simulation result

7.3.2.5 Demonstration of the Remote Security Monitoring Service

The proposed remote service monitoring service has been tested on the developed demonstrator. As shown in Figure 7.15, the status of RSOs could be viewed on the home service gateway. When press a button of the *Window Shock Detector* installed on the demonstrator, the status of the RSO will change to “Shocked”. As shown in Figure 8.15, the SOAP packets sent out by the RSO can be captured by the associated tools (Commview) installed in the home service gateway. This approach can help to monitor and analyse the message exchange between RSOs and the home service gateway. If the *Hall Motion Detector* has detected a movement within the predefined period of time (say 2 minutes), it will send out a SOAP message to the home service gateway, which will then forward to the assigned service process. The service process will send out an alarm message to both the remote security monitoring service application and an assigned mobile phone. Reset the demonstrator, active the *Window Shock Detector* again. If the *Hall Motion Detector* can not detect a movement within the predefined time period, instead of generating an alarm message, the service process will only send out a reporting message to the remote security monitoring service application.

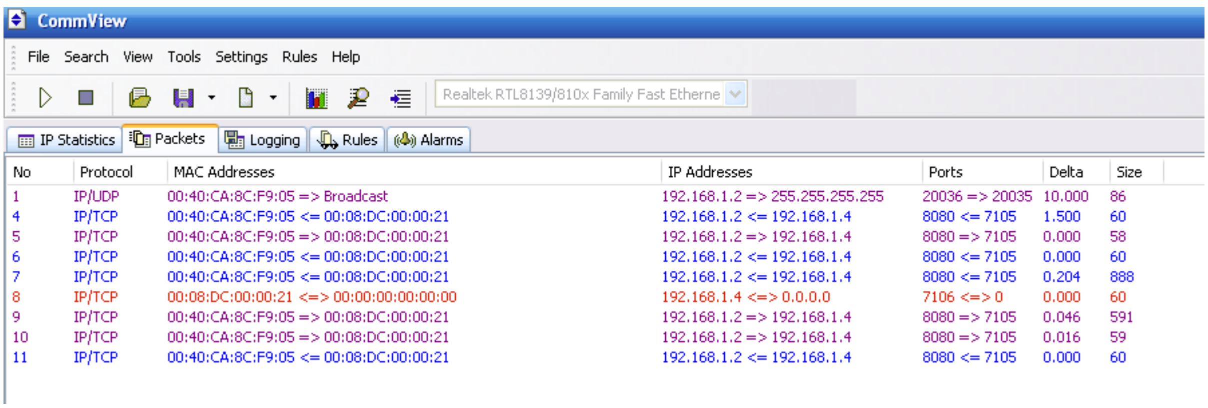

Figure 7.15 Screenshot of SOAP message exchange between RSOs and home service gateway

7.3.2.6 Evaluation

The evaluation for remote service creation methodology is summarised as follows:

1) The adopted BPEL open standard has proved to be well structured and able to describe the execution sequence of service application. By using the open source BPEL development toolsets, the BPEL based service application is reasonably easy to build, test, and debug.

2) The demonstration has proved that the home service gateway is able to manage the binding between remote service application and domestic service resources.

3) The service application has proved to be able to deliver advanced services to the connected home environment. High-level messages have proved to be able to be sent to remote service provider and householder when it is necessary.

4) The demonstration has proved that the embedded ODE engine can support the compliance and execution of BPEL based service applications.

5) The demonstration has proved that the developed home service gateway can support the creation of services when needed. The authenticated remote service provider has the right to start or stop the service according to the pre-defined service contract.

6) The demonstration has proved that with the information exposed on the home service gateway, the remote service provider could customise the delivered service application to meet the specific service requirements of householders regardless of the difference of their home settings.

7) The demonstration has proved the methodology, remote service delivery platform and associated tools have created a “level-playing field” for SMEs, mainly because of its “Openness” and standards based approach. Such openness and approach would improve the “Freedom of Choice” to both service providers and recipients immensely.

8) The demonstration has proved that the proposed remote service delivery platform would able to improve the quality of service (e.g. reducing false alarm in security monitoring service) via the concept of services aggregation with the proposed methodology, platform and associated tools.

7.3.3 Summary of Test Case One

1) A remote security monitoring service has been developed by utilizing the proposed realization methodologies and associated tools.

2) The test of the remote security monitoring service is carried out with the developed demonstrator. The evaluation shows that the developed home service gateway and available RSOs not only able to support the remote service provision that operating with "On-Demand" business model, but also support the service "Mass Customization".
3) Although the proposed methodologies does not provide ideal solution in terms of automation and intelligence. However, they have proved that with the information exposed on the home service gateway, the remote service provider could customise the delivered service application to meet the specific service requirements of householders regardless of the difference of their home settings.

7.4 Test Case Two: Advanced Information Service

This test case introduces an advanced information service to evaluate and illustrate the openness of the home service gateway, which is able to support third parties to create different types of services without discrimination. It also demonstrates that the platform is able to support the concept of "Resources Sharing" within the connected home environment.

- The occupancy information of the *Lounge Motion Detector* could be utilised to create an advanced information service application on the home service gateway, which would translate the status of *Lounge Motion Detector* into service specific meaning via adaptable messaging mechanism. It can facilitate the occupancy information of the *Lounge Motion Detector* to be shared between different types of service applications.

This test case focuses on the follows:

1) Design an advanced information service application prototype that would translate the original device message into service specific meaning via adaptable messaging mechanism.
2) Design a remote information service application that could utilize the exposed advanced information service application functionalities.

3) Execute the advanced information service application and remote information service with the aid of the project demonstrator.
4) Analyze and demonstrate the concept of “Resource Sharing” within a single connected home.
5) Evaluate the proposed methodologies and associated tools.

7.4.1 Creating an advanced Information Service Application

The advanced information service application is also established within the home service gateway prototype. It devised and implemented some advanced functionalities to convert the status of the *Lounge Motion Detector* into service specific message.

As illustrated in Figure 7. 16, the advanced information service application is a wrapper of RSO model in order to provide different types of event notifications to different service providers. For example, the “Occupancy monitoring” notification could be used in a health care service application to monitor that an occupancy is immobile for a given period of time (say 1 hour); and the “Movement detected” notification could be used in a security monitoring service application to observe an intrusion activity. When service applications subscribe to a RSO event, the type of event is specified with a criterion as to the event values of the attributes. Whenever an event of that type occurs that matches the attributes, the assigned service applications will be notified.

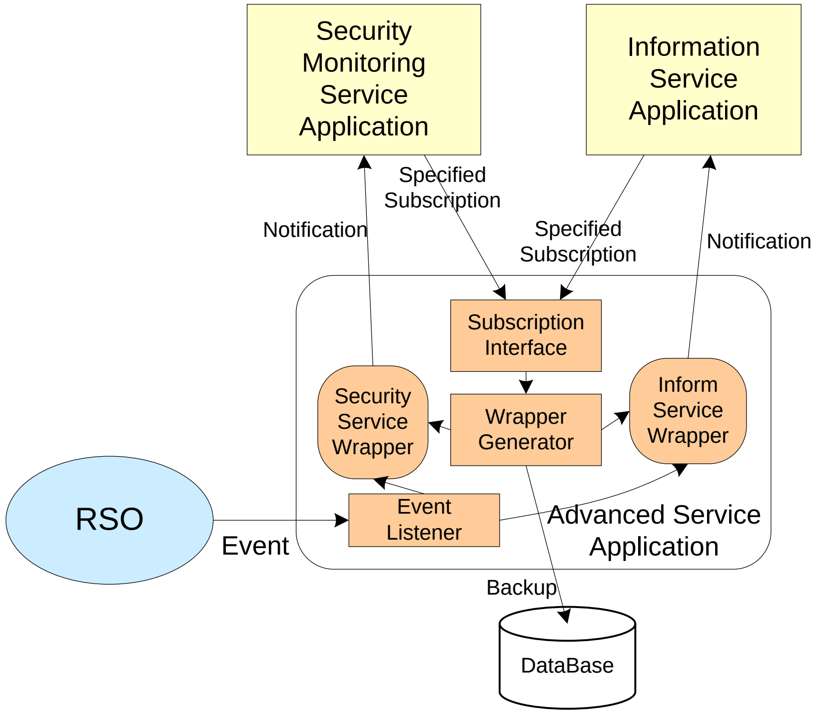

Figure 7.16 Advanced Information Service Application Architecture

Criteria of the advanced information service application are exposed in its WSDL interface. During subscription processes, the remote service provider would generate a specific subscription document based on these criteria and sent it back to the advanced information service application. The advanced information service application is responsible to retrieve these specific attributes and save them into the home service gateway database as backup. It would also generate a script document as a wrapper to this service provider with specific attributes and service logic.

7.4.1.1 Overview of an Advanced Information Service

As shown in Figure 7.17, the original *Lounge Motion Detector* object has been specific into two functional processes: Occupancy monitoring process and Movement detected process.

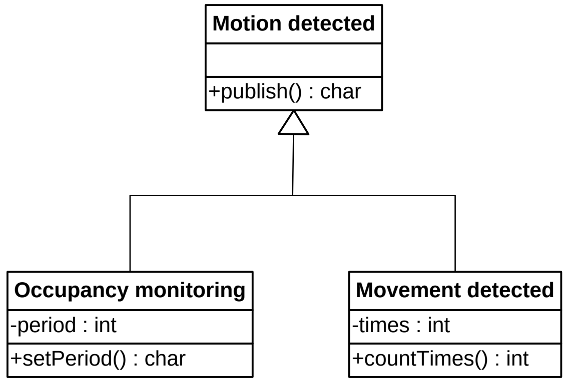

Figure 7.17 Class Diagram of Advanced Information Service Application

The “Occupancy monitoring” functional process is designed to be used in counting the immobile period of time of occupancies. Instead of sending notification to assigned service application every time when the status of the *Lounge Motion Detector* has changed, the specific object (wrapper) would calculate the immobile period of time and only send out the event notification when the predefined time has lapsed. This information could be used to signify a potential threat in well-being of a person who is under remote care service (e.g. health monitoring). For example, if the sensor detected an elderly person had not moved in the day time for over an hour, it may signify the elderly is suffering a condition or an accident. If the healthcare staff or relatives could receive an alert on time, those raised concerned can be deal with swiftly to avoid any unwanted consequence.

The “Movement detected” functional process is designed to be used in counting the times of moving activities. This information could be used in the remote security monitoring service to verify if the “Broken” message sends out from the *Patio Door Glassbreak Detector* is a genuine intrusion event.

7.4.1.2 Demonstration of Using Advanced Information Service Application

A remote information service application has been created to test and evaluate the proposed advanced information service application. As shown in Figure 7.17, the available functionalities of the *Lounge Motion Detector* could be viewed on the exposed WSDL interface on the home service gateway. The remote information service provider can easily retrieve the interface and use it to predefine the static period of time (say 3 minutes for demonstration).

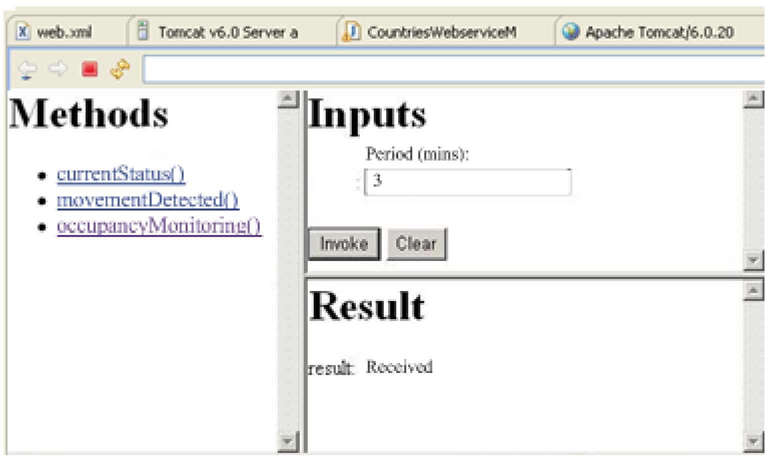

Figure 7.18 Screenshot of Advanced Information Service Application Subscription Interface

The subscription processes of the remote information service application are similar as the remote security monitoring service application. However, as the static period of time would be monitored by the specific wrapper generated in the home service gateway, the remote information service application can directly use the event notification send out by the advanced information service application. Therefore, the service process of the remote service application could become very simple.

7.4.1.3 Evaluation

The evaluation result of the "Resources Sharing" methodology is summarised as follows:

1) The flexible concept of RSO would able to provide a uniform interface for a well-defined service type regardless the difference of the devices, appliances and systems, it represents. It also proved the concept of “Adaptive Message” not only ease the use of the service(s) it provided but also promote “Resources Sharing” to lower the “Total Cost of Ownership”.
2) The developed home service gateway is proof to be an open and general which would support remote service provision of different service applications without discrimination.

7.5 Summary

Two test cases have been carried out to evaluate the proposed remote service delivery platform and associated tools. The results have successfully shown the proposed methodology, developed platform and associated tools would able to support remote service provision to connected home environments, which support “On-Demand” business model and service “Mass Customisation”. The results also successfully showed the ease of “Resources Sharing” with the proposed platform, which would lower the “Total Cost of Ownership”.

Chapter 8 Conclusions and Future Work

8.1 Conclusions from this Research study

The aim of this research study is to investigate the method and associated tools for the realisation of a "Service-On-Demand" business model with the supporting of "Mass Customization" and provides "Freedom of Choice" for connected homes. The literature review revealed that there are no existing methods and tools to support a remote service delivery platform which would act as a general platform to provide remote service provision, although some existing architectures and many research projects have indicated possible solutions for part of the system.

This research study has focussed on:

1) A methodology for encapsulating home appliances, devices and environment information into Internet-accessible service objects has been developed. The proposed methodology has offered authenticated remote service providers with a set of Interfaces which remote service providers can retrieve home service resources' information, and then uses them to customise a service application to meet user requirements regardless of the difference of home environment settings.
2) The research study has adopted a widely accepted Web Service management format – BPEL to aggregate the relevant home service resources to formulate specific service applications. The proposed methodologies and associated tools support the concept of "Resource Sharing" between different service applications via "adaptive" message (i.e. data transform to messages that specific to application domain). This approach will help to lower the "Total Cost of Ownership".
3) Two types of service applications, namely remote security monitoring service and advanced information service have been developed and demonstrated with the aid of the

developed project demonstrator; to evaluate the proposed methodologies and associated tools and illustrate the proposed concepts.

8.2 Achievements

Main achievements in this research study could be viewed from the following three aspects: project demonstrator, remote service delivery platform and contribution of knowledge.

8.2.1 Project Demonstrator

A demonstrator has been developed to illustrate and evaluate the proposed remote service provision methodologies. RSO-compliant sensors developed on the demonstrator have proved to be able to auto-register and managed by the home service gateway. Two advanced service have proved to be able to deliver remotely with the support of demonstrator.

8.2.2 Remote Service Delivery Platform

The developed remote service delivery platform has proved to be able to connect remote service providers and connected home service resources with automatic management and data security via the developed demonstrator and service applications (i.e. a remote security monitoring service and an advanced information service). The demonstrator and these services have also proved the developed platform is able to support the “On-Demand” service provision with “Freedom of Choice”. The developed platform has also proved the critical ability of an open platform in supporting remote service provision of different types of services by third parties without discrimination, and support “Resource Sharing” between different services with “adaptive” messages.

8.2.3 Contribution of Knowledge

One of the contributions of this research study is to present methodologies and associated tools to encapsulate home appliances, devices and environment information into service objects that enable service interoperability in connected home environments; an essential ingredient to support remote service provision with the ability of "Mass Customisation" to meet individual needs regardless the difference of the home settings. This research study employs a generic micro-controller based hardware module to transform the selected "off-the-shelf" devices into interoperable-networked devices that are independent from their communication protocols. It also supports an advanced implementation model which use web service application to wrap the hardware module in order to provide a consistent management and standard interfaces. This approach simplifies the complicated life-cycle management of home service resources; hides the heterogeneous nature of the home service resources, and provides a systematic and integrated solution to support information rather than data interoperability between home service resources and remote service providers. Therefore, remote service providers can utilise available home service resources to create advanced services when the needs arise (i.e. "On-Demand") which could meet specific requirements of householders regardless of the difference between home settings.

Another contribution of this research study is the investigation of methods and tools for an open home service gateway which supports third party service providers to create different types of services easily and efficiently without discrimination. The widely accepted Web service management standard - BPEL is adopted to facilitate the aggregation of the relevant service resources into specified service applications, and a lightweight BPEL engine – Apache ODE is embedded into the home service gateway to avoid unnecessary interaction and data go through the remote service providers' server. The developed home service gateway supports "Resources Sharing" and "adaptive" message for different types of services. These approaches would further reduce the "Total Cost of Ownership", elevate company interest for investment, and attract customers to buy and install the necessary infrastructure support.

This research study is completed with evaluations of the remote service delivery platform that confirms its advantage in terms of performance and resource utilisation. It introduces new dimensions such as “On-Demand” and service “Mass Customisation” for remote service provision in a flexible and integrated manner.

8.3 Recommendations and Future Work

Although this research project has successfully demonstrated interoperability between connected home service resources and remote service providers can be significantly improved with the proposed architecture, methodology and developed toolset, there are still a number of research areas requiring further attention and efforts to complete the picture.

The proposed RSO enhanced model supports service interoperability via service-based modelling which can hide the heterogeneous nature of home service resources. However, the lack of widely accepted ontology dictionaries for connected home environments and service domains (e.g. security and healthcare) will limit the openness of the proposed model. Further research works are required to establish a definitive and exhaustive classification of home service resources, which would ensure the openness of the proposed model to realise the service level interoperability.

The proposed concept of “adaptive” message has proved to be effective and efficient in supporting “Resource Sharing” between different types of service applications. Nowadays, “Resource Sharing” and service interoperability across heterogeneous platforms is mainly built upon on neutral “languages” (i.e. XML, SOAP, and HTTP) and Web services. However, the poor performance and transaction support of Web services standards result in the difficulty of service interaction. Hereby, a set of strict binding mechanisms have to be defined to ensure the cross-platforms interoperability between connected home service resources and service providers. One of the on-going research subjects is how to dynamically integrate Software Agents with Web services to support autonomous transaction and interaction among different service systems and platforms. Future research in this area to further improve the interoperability between connected home service resources and remote service providers are highly desirable.

Definition of Terms

Components – A component is a physical unit of implementation with well-defined interfaces that is intended to be used as a replaceable part of a system.

Context – Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves.

Platform – a Platform is a defined support structure in which another software project can be organized and developed.

A technology platform is a collection of things. It can include one or more architectures, technologies, concepts, models, and even sub-platforms.

Householder – One who occupies or owns a house.

HTML (Hyper-Text Mark-up Language) – A simple mark-up and formatting language for text, with links to other objects, used with the web.

HTTP (Hyper-Text Transport Protocol) – The basic protocol of the web, used for communication between browsers and web sites.

Information and communications technologies (ICTs) – ICT includes technologies such as desktop and laptop computers, software, peripherals and connections to the Internet that are intended to fulfil information processing and communications functions.

Infrastructure – The underlying base or foundation for a system. The basic facilities and services for the functioning of a system.

Interface – An interface is a descriptor for the externally visible operations of a class, component, or other entity (including summarization units, such as packages) without specification of internal structure. An interface is essentially equivalent to an abstract class with no attributes and no methods and only abstract operations.

Message – Message in its most general meaning is the object of communication. In computing, under certain object-oriented programming languages such as Objective C, a message is an instruction to an object to perform some task.

Model - A model is a representation in a certain medium of something in the same or another medium. The model is purpose to capture and precisely state requirements and domain knowledge so that all stakeholders may understand and agree on them.

Module – A software unit of storage and manipulation. Modules include source code modules, binary code modules, and executable code modules.

Operating system – In computing, an operating system is the system software responsible for the direct control and management of hardware and basic system operations. Additionally, it provides a foundation upon which to run application software such as word processing programs, web browsers and others.

Platform – The type of hardware on which a given operating system or application program runs.

Protocol - A set of rules that describe the sequence of messages sent across a network, specifying both syntax and semantics.

Proxy Server - A computer that acts as a bridge between two computer systems that use different standards, formats, or protocols.

Raw Data – Raw data, sometimes called source data or atomic data is data that has not been processed for use. A distinction is sometimes made between data and information to the effect that information is the end product of data processing. Raw data that has undergone processing is sometimes referred to as cooked data.

Although raw data has the potential to become “information”, it requires selective extraction, organization, and sometimes analysis and formatting for presentation.

Reference Architecture – A reference architecture is a high-level system design free of implementation details and consists of high-level description of system components, definitions of relationships between components, definitions of relationships between system components and elements external to the system, and identification of performance drivers and capacity requirements.

Residential Gateway – A hardware device integrates network-enabled devices and connects them to the Internet.

Service - A service is a coarse-grained processing unit that consumes and produces sets of objects passed-by-value. The service consists of a connection of components that is able to deliver business functions that service represents.

The concern address by a service can be small or large. Therefore, the size and scope of the logic represented by the service can be vary. Further, service logic can encompass the logic provided by other services. In this case, one or more services are composed into a collective.

Service Provider – The organization that procures or develops Service Applications and deploys these applications via a Service Deployment Manager on Service Platforms.

Smart Home – A dwelling incorporating a communications network that connects the key electrical appliances and services, and allows them to be remotely controlled, monitored or accessed.

System – A collection of connected units organized to accomplish a purpose. A system can be described by one or more models, possibly from different viewpoints.

XML – XML (Extensible Markup Language) is a W3C initiative that allows information and services to be encoded with meaningful structure and semantics that computers and humans can understand. XML is great for information exchange, and can easily be extended to include user-specified and industry-specified tags. [www.orafaq.com/glossary/faqglosx.htm]

Virtual community – A virtual community is a community of people sharing common interests, ideas, and feelings over the internet or other collaborative networks.

Reference

Abrach, H. S. Bhatti, ect. (2003) “MANTIS: System Support For MultimodeAI Networks of In-situ Sensors,” *Proc. Of 2nd ACM International Workshop on Wireless Sensor Networks and Applications*, 2003, pp. 50-59.

ACPO (2006) ACPO policy on police response to security systems, April 2006. Available: http://www.ciaalarms.co.uk/downloads/acpo_alarms2006.pdf

ADT. Monitored Home Security Systems, Available: http://www.adt.co.uk

Aglets. Available: http://aglets.sourceforge.net/

Ahmed, R. Boutaba, R. etc. (2005) “Service Discovery Protocols: A Comparative Study,” *IFIP/IEEE International Symposium on Integrated Network Management Application Sessions*, Nice, France, 2005

Alison, Nicholl and Perry, Mike. (2008) Smart home systems and code for sustainable homes, Available: http://www.bre.co.uk/filelibrary/ibexcellence/ibexcellence_smart_home_systems.pdf

Amigo. Available: www.amigo-project.org

Amigo (2005) “Open development platforms for software and services,” Ambient Intelligence for the networked home environment- short project description, January 2005.

Ankolekar, A. (2002) “DAML-S: Web Service Descriptioon for the Semantic Web,” *Proc. ISWC’02. Number 2342 in LNCS*, Springer (2002) 348-363

Baek, SH., Choi, EC (2007) “Sensor Information Management Mechanism for Context-aware Service in Ubiquitous Home,” Consumer Electronics, IEEE Transactions on Volume 53, Issue 4, Nov, 2007 Page(s): 1393-1400

Baker, Sean. (2003) "Web Services, CORBA and other Middleware," *Web Services for the integrated enterprise, OMG Workshop,* Munich Feb 2003.

Barreto, Charlton., Bullard, Vaughn., Erl, Thomas., etc. (2007) "Web Services Business Process Execution Language Version 2.0," 9 May 2007. Available at: http://www.oasis-open.org/committees/download.php/23964/wsbpel-v2.0-primer.htm#_Toc166509757

Baumann, J., Hohl, F, and Steedman, D. A. (1998) "Mole 3.0: A middleware for Java-based mobile software agents." *Proc. Middleware 1998: IFIP Int. Conf. Distrib. Syst. Platforms Open Distrib, Process.*, 1998, pp. 355-372

Beek, MH. Bucchiarone. And Gnesi, S. (2007) "Formal Methods for Service Composition," Annals of Mathematics, Computing & Teleinformatics, Vol 1, No 5, 2007, pp. 1-10.

Bonino, Dario and Corno, Fulvio. (2008) "DogOnt – Ontology Modelling for Intelligent Domotic Environments," *Springer-Veriag Berline Heidelberg*, LNCS 5318, pp(s) 790-803

Booth, David. Haas, Hugo. McCabe, Francis. (2003) "Web Services Architecture", August 06, 2003. Available at: http://www.w3.org/TR/2003/WD-ws-arch-20030808/

Borges, B. Holley, K. and Arsanjani, A,. (2005) Delving into Service-Oriented Architecture. Available at: http://www.developer.com/java/ent/article.php/10933_3409221_3

Bourcier, Johann. etc. (2006) "A Dynamic-SOA Home Control Gateway," *IEEE International Conference on Services Computing (SCC'06)*, May 2006

Brown, Paul. (2007) "An Introduction of Apache ODE," Sep, 2007. Available at: http://www.infoq.com/articles/paul-brown-ode

Braun, P and Rossak, W,. (2005) "Mobile Agents – Basic Concepts, Mobility Models, and the Tracy Toolkit. San Mateo, CA: Morgan Kaufmann, 2005

Bronsted, J. Hansen, KM, Ingstrup, M. (2007) " A Survey of Service composition Mechanisms in Ubiquitous Computing," In proceedings of UbiComp 2007 Workshop, June 2007, Vol. 4717, No. 9, pp. 87-92, Innsbruck, Austria.

Chakraborty, Dipanjan. (2004) "Service Discovery and Composition in Pervasive Environments, "2004

Channabasavaiah, K. Holley, K. Tuggle, E. (2003) "Migrating to a Service-Oriented Architecture," *IBM Developer Works*, Dec, 2003.

Chen, H., Tim Finin, etc. (2004) "Intelligent Agents Meet the Semantic web in Smart Spaces", *IEEE Computer Society*, November 2004

Chen, X. Wong, C.B. (2005) "eService-On-Demand for connected homes – Final Report", December 2005

Christensen, Erik., Curbera, Francisco., Meredith, Greg., Weerawarana, Sanjiva. (2001) "Web Services Description Language (WSDL) 1.1," *W3C Note* 15 March 2001.

Chu, S.C. (2005) "From component-based to service oriented software architecture for healthcare, Enterprise networking and Computing in Healthcare Industry." *HEALTHCOM 2005. Proceedings of 7th International Workshop* on 23-25 June 2005 Page(s):96 – 100.

Compare Monitoring Systems, Available: http://www.smarthome.com/alarm.html

Connected Home News, Available: www.connected-home-news.com

Cook, D. J., Michael youngblood, etc. (2003) "MavHome: An Agent-based Smart Home", *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, June 2003

Cory D. Kidd Robert J. Orr, etc, (1999) "The Aware Home: A Living Laboratory for Ubiquitous Computing Research, " Proc. Of the Second International Workshop on cooperative Buildings – CoBuild'99, 1999

Das, S,K. Cook, D,J.etc. (2002) "The Role of Prediction Algorithms in the MavHome Smart Home Architecture, " IEEE Wireless Communications, Vol.9, No. 6, pp. 77-84, Dec, 2002.

Dey, A. K. (2000) "Providing Architectural Support for Building Context-Aware Application", Doctor of Phiosophy in Computer Science, November 2000.

Dey, A.K., et. (2001) "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications." *A Special Issue on Context-Aware Computing, Human-Computer Interaction (HCI) Journal*, Vol. 16. (2001)

Dimitrov, Todor., Naroska Edwin., Schmalenstroer Jorg. (2007) "Amigo Training Document: Using Web-services." IST Amigo Project Tutorial, June 2007.

Dumitru, Roman., Uwe Keller., Holger, Lausen. (2005) "Web service Modelling Ontology," Applied Ontology, 1(1) 77-106, 2005.

Endrei, M. Ang, J.. Arsanjani, A. Chua, Sook. Comte, Philippe. Krogdahl, Pal. Luo, Min. and Newling, Tony. (2004) Patterns: Service-oriented Architecture and Web Services. IBM Redbook, ISBN 073.845317X.

Engelstad, Paal. Yan zhen, etc. (2003) "Service Discovery and Name Resolution Architecture for On-Demand MANETs, *Proceeding of the 23rd International Conference on Distributed Computing Systems Workshops*, June 2003

Erl, Thomas. (2004) Service-oriented Architecture: A Field Guide to Integrating XML and Web Services. April 6, 2004, Pearson Education, Inc.

Erl, Thomas. (2005) Service-Oriented Architecture: Concepts, Technology, and Design. July 2005, Pearson Education, Inc.

Etter, Richard and Roecker, Carsten. (2007) Social radio - a musical based approach to emotional awareness mediation IUI07 , Jan 2007.

Farlex, (2005) “The free dictionary”, 2005, Available: http://www.thefreedictionary.com/householder

Gbdirect Ltd. (2009) “Benefits of using open source software,” Available at: http://open-source.gbdirect.co.uk/migration/benefit.html

Gilbert, Martyn, (2004) “TAHI Open Architecture, Definitive Description”, November 19, 2004.

Griffin, D. Pesch, D. (2007) “A Survey on Web Services in Telecommunications,” *IEEE Communications Magazine*, July 2007, Vol. 45, No. 7, PP. 28-35.

Gulbrandsen, A. Vixie, P. Esibov, L. (2000) “A DNS RR for specifiying and location of service (DNS SRV)”, RFC 2782, *Internet Engineering Task Force (IETF)*, February 2000.

Guttman, E. Perkins, C. etc. (1999) “SLP, Servcie Location Protocol”, June, 1999. Available: http://tools.ietf.org/html/rfc2608.

Hamadi, R., Benatallah, B. (2003) “ A Petri Net-based Model for Web Service Composition,” In: ADC’03. Number 17 in CRPIT (2003) 191-200.

Han, CC. Kumar, R.etc. (2005) “A Dynamic Operating System for Sensor Nodes,” *Proc. Of MobiSys*, 2005, pp. 163-176.

Helal, Sumi, etc. (2005) Smart House: A Programmable Pervasive Space, IEEE Computer Society, 0018-9162/05, 2005

Heiman, Richard. (2009) “Worldwide software 2009-2013 forecast summary,” Jun 2009, iDC No. 218938. 2009. IDC Report.

Holmes, D., Stocking, R., (2009) “Augmenting agent knowledge bases with OWL ontologies**.”** *Aerospace conference, 2009 IEEE* 7-14 March 2009 Page(s):1 - 15

Hsu, CC., and Wang, LZ. (2008) "A Smart Home Resource Management System for Multiple Inhabitants by Agent Conceding Negotiation." *IEEE System, Man and Cybernetics, 2008. SMC 2008.* October 2008.

Hsu, CC., and Chien, YY. (2009) "An Intelligent Fuzzy Affective Computing System for Elderly Living Alone." *Ninth International Conference on Hybrid Intelligent systems Conference* Volume 1, 12-14 Aug. 2009 Page(s):293 - 297

IBM and SAP. (2005) WS-BPEL extension for people – BPEL4People, July 2005. Available at: http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel4people/BPEL4People_white_paper.pdf

IEEE1451 committee. (2000) "IEEE Standard for a Smart Transducer Interface for Sensors andActuators-Network Capable Application Processor (NCAP) Information Model." *TC-9 Committee on Sensor Technology of the IEEE Instrumentation and Measurement Society, USA*, 2000

Inclusion (2006) Independent Living research activities (the United Kingdom), Available: http://www.einclusion-eu.org/ShowCase.asp?CaseTitleID=1889

iptvWorldforum, (2008) Available: www.the-connected-home.co.uk March 2008, London

Istepanian, R., Laxminarayan, S and Pattichis, C. (2007) "Remote Monitoring for Healthcare and for Safety in Extreme Environments", 978-0-387-26558-2, Springer US, January 04, 2007

Jini. (2005) Jini Specifications Archive - v2.0, 2005 Available: http://java.sun.com/products/jini/2_0index.html

Johansen, D., (1998) "Mobile agent applicability." Proc. 2nd Int. Workshop Mobile Agents, Sep 1998, pp. 80-89.

Kalasapr, S., Kummar, M., Shirazi, B. (2007) "Dynamic Service Composition in Pervasive Computing." *Parallel and Distributed Systems, IEEE Transactions* on Volume 18, Issue 7, July 2007 Pages:907-918

Kavantzas, Nickolaos., Burdett, David., Ritzinger, Greg. (2004) Web Service Choreography Description Language Version 1.0, W3C Working Draft, April, 2004.

Kiani Saad L, Maria Riaz, Sungyong Lee, Young-Koo Lee, (2005) "Context Awareness in Large Scale Ubiquitous Environments with a Service Oriented Distributed Middleware Approach", *Processding of the Fourth Annual ACIS International Conference on Computer and Infomation Science*, 2005 IEEE.

King, Nicola. Avenue, Davy. (2003) "Smart Home – A Definition," September 2003, Available: http://www.changeagentteam.org.uk/_library/docs/housing/smarthome.pdf

Kohn Dan. (2008) "The Linux Driver Model: A Better Way to Support Devices," June 2008, Linux Foundation. Available: https://www.linuxfoundation.org/collaborate/publications/linux-driver-model

Laerhoven, K. Van. (2005) "The Pervasive Sensor", *Invited Talk, In Ubiquitous Computing Systems, Second International Symposium, UCS 2004, Tokyo, Japan,* November 8-9, 2004, Revised Selected Papers. Springer LNCS 3598, Springer 2005, pp 1-9.

Lautenbacher, F. Bauer, B. (2007) "A Survey on Workflow Annotation & Leutnant Composition Approaches", In Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SemBPM), Innsbruck, Austria,, June 2007, pp.12-23.

Leutnant Volker., Schmalenstroeer Joerg., Poortinga Remco. (2007) "Context Management Service." IST Amigo Project tutorial, October 2007.

Levis, P. S. Madden, D. Gay, J. Polastre, R. Szewczyk, etc. (2004) "The Emergence of Networking Abstractions and Techniques in TinyOS," *Proc. Of the First USENIX/ACM Symposimum on Networked Systems Design and Implementation*, 2004

Look Gary, etc (2005) "A Location Representation for Generating Descriptive Walking Directions. *In IUI 05: Proceedings of the 10th International Conference on Intelligent User Interfaces*, pp. 122-129. 2005

Look Gary & Shrobe Howard. (2007) "Toward Intelligent Mapping Applications: A Study of Elements Found in Cognitive Maps." *In IUI 07: Proceedings of the 12th International Conference on Intelligent User Interfaces*, pp. 309-312. New York, NY, USA, 2007.

Louati, Wassef. Marc Girod Genet, etc. (2005) "UPnP Extension for wide-area Service Discovery using the INS/Twine Framework*", 2005 IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communicat*ions, May, 2005

Lv, QC., Zhou, JL. (2009) "Service Matching Mechanisms in Pervasive Computing Environments," *Intelligent Systems and Applications, 2009. ISA 2009 International Workshop* on 23-24 May 2009 Page(s): 1-4

McGuire, Tammy J. (2009) False Alarm Reduction Unit, Available: http://www.co.cal.md.us/residents/safety/falsealarm/

Meixell, M.J., Shaw, N.C., Tuggle, F.D. (2008) "A Methodology for Assessing the Value of Knowledge in a Service Parts Supply Chain," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* Volume 38, Issue 3, May 2008 Page(s):446 - 460

Merck Company Foundation, (2008) The State of Aging and Health in America 2008, Available: http://www.cdc.gov/aging/saha.htm

Mitsuhiro OGAWA, and Toshiyo TAMURA,. (1997) "Fully Automated Biosignal Acquisition System for Home Health Monitoring", *Proceedings – 19th International Conference – IEEE/EMBS* October 30- November 2, 1997 Chicago, IL, USA

Mockapetris, P. (1987) "Domain names – concepts and facilities", *RFC 1034, Internet Engineering Task Force (IETF)*, November 1987.

Moore, Philip.(2005) “eServices-on-demand for connected homes Seminar Report.” TAHi December 2005 Newsletter, December 2005.

Moyer, S. (2009) “Enabling Service Aggregation for Providers of Consumer Services and Applications”, *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE***,** Jan, 2009

National Statistics. (2008) Internet Access 2008- Households and Individuals, 28, August, 2008

National Statistics, (2008) Ageing – 16% of UK population are aged 65 or over, 22, August, 2008, Available: http://www.statistics.gov.uk/cci/nugget.asp?ID=949

Nicholl, Alison and Perry, Mike. (2008) Smart home systems and code for sustainable homes, Available: http://www.bre.co.uk/filelibrary/ibexcellence/ibexcellence_smart_home_systems.pdf

Nokia. (2005) White Paper – Service Delivery Platform, February 2005.

Noury, N. (2005) “AILISA : experimental platforms to evaluate remote care and assistive technologies in gerontology.” *Enterprise networking and Computing in Healthcare Industry, 2005. HEALTHCOM 2005. Proceedings of 7th International Workshop on* 23-25 June 2005 Page(s):67 – 72.

Noy, NF. (2004) “Semantic integration: a survey of ontology-based approaches,” ACM SIGMOD Record, Vol. 33, No. 4, December 2004, pp. 65-70.

Oh, Seog-Chan. Lee, Dongwon and Kumara, Soundar. (2005) “A Comparative Illustration of AI Planning-based Web Services Composition, “ ACM SIGecom Exchanges, Vol.5, No. 5, 2005, pp. 1-10.

OSGi. (2009) “OSGi Service Platform Core Specification”, Release 4, Version 4.2, June, 2009.

Othman, Z.A., Abu Bakar, A., Hamdan, A.R., Omar, K., Shuib, N.L.M., (2007) "Agent based preprocessing." *Intelligent and Advanced Systems, 2007. ICIAS 2007. International Conference on* 25-28 Nov. 2007 Page(s):219 – 223

Parliamentary Office of Science and Technology, (2006) Pervasive computing, May 2006

Park, S. Kim, J. etc. (2006) "Embedded Sensor Networked Operating System," *Proc. Of 9th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing*, 2006

Park, Jung Wook. Yoo Suk Jung. Hui Jung Park. Soon Dong Kim. We-Duke Cho. (2009) "WIS : A well-being index based health care system in smart home." *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on* 9-13 March 2009 Page(s):1 - 3

Parks Associates (2008) White paper, Europe: Home Network Update, 2008

Percivall, George. Reed, Carl. (2006) "OGC Sensor Web Enablement Standards", *Sensors & Transducers Journal*, Vol. 71, Issue 9, September 2006, pp. 698-706.

Pessoa, R.M., Silva, E., van Sinderen, M., Quartel, D.A.C., Pires, L.F. (2008) "Enterprise interoperability with SOA : a survey of service composition approaches." *Enterprise Distributed Object Computing Conference Workshops, 2008 12th* 16-16 Sept. 2008 Page(s):238 – 251

Platonov, Ilya., Papkov, Artem., Smith, Jim. (2008) "Develop and execute WS-BPEL V2.0 business processes using the Eclipse BPEL plug-in," IBM DeveloperWorks, 25, Mar, 2008.

Pollice, Gary. (2005) "Why I teach Eclipse," *IBM developerWorks*, Jun, 2005.

Quitadamo,R., Zambonelli, F., Cabri, G. (2007) "The Service Ecosystem: Dynamic Self-Aggregation of Pervasive Communication Services." *Software Engineering for Pervasive*

Computing Applications, Systems, and Environments, 2007, SEPCASE'07. First International Workshop on 20-26 May 2007 Pages: 1-1.

Redondo, Diaz., Vilas, R.P., Cabrer, A.F., Pazos, M.R., Marta Rey Lopez. (2007) "Enhancing Residential Gateways: OSGi Service Composition," *Consumer Electronics, IEEE Transactions* on Volume 53, Issue 1, February 2007 Page(s):87 - 95

Potamianos, G., Jing, Huang., Marcheret, E. (2008) "Far-Field Multimodal Speech Processing and Conversational Interaction in Smart Spaces," Hands-Free Speech Communication and Microphone Arrays, 2008, HSCMA 2008, May 2008 Page(s): 119-123

Ricquebourg, V, Menga D, etc, (2006) "The Smart Home Concept: Our immediate future." *E-Learning in Industrial Electronics, 2006 1ST IEEE International Conference*, Dec, 2006.

Riou, Matthieu. (2008) Web Services, Orchestration and Apache ODE, ApahceCon US 2008. Available at: www.eu.apachecon.com/presentation/materials/40/ApacheODE_ApacheConUS2008.pdf

Royon, Y. Frenot, S. (2007) "Multiservice home gateways: business model, execution environment, management infrastructure." *Communications Magazine, IEEE* Volume 45, Issue 10, October 2007 Page(s):122 – 128

Rubini, A. (1998) "Linux Device Drivers," O'Reilly & Associates, Inc., 1998

Sahai, Akhil. (2000) "Managing Next Generation E-Services." HP Labs Technical Reports, 29,09,2000

Salaun, G., Bordeaux, L., Schaerf, M. (2004) "Describing and Reasoning on Web Service using process Algebra. In: *Proc. ISOLA'* (2004) 43-50.

Saravanan, P., Reuter, E., Verma, S. (2008) "Enhancing enterprise network management using SMART." *India conference, 2008. INDICON 2008*, Annual IEEE, Volume 2, 11-13 Dec. 2008 Page(s): 343-348

Satoh, F., Yamaguchi, Y. (2007) "Generic Security Policy Transformation Framework for WS-Security." *Web Services, 2007. ICWS 2007. IEEE International Conference* on Digital Object Identifier: 10.1109/ICWS.2007.92, Page(s): 513-520

Schilit, Bill N. and Marvin M. Theimer (1994). Disseminating active map information to mobile hosts. IEEE Network 8(5): pp. 22-32. September/October 1994.

Selcuk, K. Li, Wen-Syan. Phan, Thomas. Zhou, Minqi. (2009) "Frontiers in Information and Software as Services*." IEEE International Conference on Data Engineering*, March 2009.

Shang, ZM., Wang HY. (2008) "Exception Handling in Smart Process-based Applications in Pervasive Computing Environments." *Pervasive Computing and Applications, 2008. ICPCA 2008. Third International Conference* on Volume 2, 6-8 October 2008 Page(s): 820-825.

Smarthome. (2009) Home Automation Superstore - Alarm System. Available at: http://www.smarthome.com/_/Security/Alert_Systems/_/H/1Rv/nav.aspx

Steele, Robert., etc. (2005) XML-based Mobile Agents, Proceeding of the International Conference on Information Technology: Coding and Computing, IEEE, March 2005

Sun. (2005) Jini Specifications Archive - v2.0, 2005 Available: http://java.sun.com/products/jini/2_0index.html

TenWolde,E. (2007) "Worldwide software on demand 2007-2011 forecast: A preliminary look at delivery model performance," 2007, iDC No. 206240. 2007. IDC Report.

UN (2008) United Nations Department of Economic and Social Affairs, Population Division, 2008, Available: http://www.un.org/esa/population/publications/wpp2008/wpp2008_text_tables.pdf

UPnP Forum. (2008) "UPnP Device Architecture 1.0", Version 1.0, 15 October 2008, Available at: http://www.upnp.org/specs/arch/UpnP-arch-DeviceArchitecture-v1.0.pdf

Venners, Bill. (1999) "The Jini Technology Vision", August 1999, Avilable at: http://java.sun.com/developer/technicalArticles/jini/JiniVision/jiniology.html

Vogel, Lars. (2009) "Eclipse Java IDE – Tutorial," July 2009. Available at: http://www.vogella.de/articles/Eclipse/article.html

Wang, Xiaohang. et al. (2004) " Semantic Space: An Infrastructure for Smart Spaces ", *Pervasive Computing*, July-September 2004

Wang, SQ., Chen, CF., Sung, jae. (2009) "A Framework for Wireless Sensor Network Based Mobile Mashup Applications", *Computer Science and Information Engineering, 2009 WRI World Congress* on Volume 5, March 31 2009- April 2 2009 Page(s): 683-686

Wang, TH. Park, H. Chung, JW,. (2006) "Design and Implementation of the Home Service Delivery and Management System Based on OSGi Service." *Consumer Electronics, 2006. ICCE '06. 2006 Digest of Technical Papers*. International Conference on, Jan. 2006

Wassermann, B., Ludwig, H., Laredo, J. (2009) "Distributed Cross-Domain Change Management." Web Services, 2009. ICWS 2009. IEEE International Conference on 6-10 July 2009 Page(s): 59-66

Weerasinghe, D. (2009) "Health@Home - An e-Service Model for Disease Prevention and Healthcare in the Home," 978-3-642-00412-4, Springer Berlin Heidelberg, February 14, 2009

Weiser, M. (1991) "The Computer for the Twenty-First Century," Scientific American, pp. 94-104 September 1991.

White, J. (1996) "Mobile Agents White Paper. " General Magic, Inc. Sunnyvale, CA, White Paper, 1996

White, J., Helgeson, C.S, etc. (1997) "System and method for distributed computation based upon the movement, execution, and interaction of processes in a network." U.S. Pat. Off., Washington, DC, U.S. Patent 5,603,031,1997

White, Mike. (2007) "High Tech Devices Support the elderly Living Alone," June 2007. Available at: http://www.associatedcontent.com/article/279334/high_tech_devices_support_the_elderly.html?cat=12

Woods, Stan P. Janusz Bryzek, etc. (1999) " IEEE-P1451.2 Smart Transducer Interface Module, " *Mixed-Mobile Communication Working Group of the Technical Committee on Sensor Technology TC-9 of the IEEE Instrumentation and Measurement Society*, June 1999.

Wu, CL., Liao, CF, and Fu, LC. (2007) "Service-Oriented Smart-Home Architecture Based on OSGi and Mobile Agent Technology." *IEEE Transactions on System, Man and Cybernetics – Part C: applications and reviews*, Vol, 37, No. 2, March 2007

Xie, ChangWen. (2001) Life cycle data acquisition methods and devices for consumer products and machines (WhiteBox Project), May, 2001

Yamato, Y., Nakano, Y., Sunaga, H. (2008) "Study and Evaluation of Context-Aware Service Composition and Change-Over Using BPEL Engine and Semantic Web Techniques," *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE* 10-12 Jan. 2008 Page(s):863 – 867

Yamazaki, T. (2006) **"**Beyond the Smart Home, Hybrid Information Technology, "2006. *ICHIT '06. International Conference* on Volume 2, 9-11 Nov. 2006 Page(s):350 – 355

Yang, J., and Pidgeon, P. (2007) "An Agent-based Decentralized Process Management Framework for Web service Composition," *Infoscale 2007*, June 6-8, 2007, Zuzhou, China

Yang, XY., and Moore, P. (2009) "Intelligent Products: From lifecycle data acquisition to enabling product-related services," Computers in Industry, Volume 60, issue 3, Page(s): 184-194, April 2009

Zaliva, Vadim. (2004) "Distributed Builds with Rendezvous and DistCC,", June 2004, Available at: www.crocodile.org/lord/DistCCRendezous.pdf.

Zeichick, Alan. (2008) "Tomcat, Eclipse named the most popular in SDTimes study," Software Development on the Web, April, 2008. Available at: www.sdtimes.com/link/31882

Zhang, Dacheng. Xiu, Jie. Li, Xianxian. (2007) "Dynamic Cross-Realm Authentication for Multi-party Service Interactions," *Dependable Systems and networks, 2007. DSN'07. 37th Annual IEEE/IFIP International Conference* on 25-28 June 2007 Pages:440-449

Zhang, X. Gracanin, D. (2008) "Service-Oriented-Architecture based framework for multi-user virtual environments," *Simulation Conference, 2008. WSC 2008*. 7-10 December, 2008, pages 1139-1147.

Zhang, Y. Gu, Y. etc. (2004) "Progress of Smart Sensor and Smart Sensor Networks," Intelligent Control and Automation, 2004, June 2004, pp.3600-3606

Zhu, Feng and Matt W. Mutka. (2005) "Service Discovery in pervasive computing environments", *IEEE CS and IEEE comSoc*, P81-90, 2005.