Adaptive service orchestration in ambient assisted living

Reference:
Sun Hong.- *Adaptive service orchestration in ambient assisted living*
Antwerpen, Universiteit Antwerpen, Faculteit Wetenschappen, Departement Wiskunde en Informatica, 2011, 116 p.
Handle: http://hdl.handle.net/10067/916960151162165141

Faculteit Wetenschappen
Departement Wiskunde en Informatica

Adaptive Service Orchestration in Ambient Assisted Living

Adaptieve Orkestraties van Diensten in Omgevings-ondersteunde Woon- en Leefomgevingen

Proefschrift tot het bekomen van de graad van
Doctor in de Wetenschappen
aan de Universiteit Antwerpen, te verdedigen door

Hong Sun

Promoter: Dr. Vincenzo De Florio

Antwerp, 2011

Abstract

We are now living in a world where we are continuously interacting with various service systems in our daily lives, ranging from ordering a pizza online to filing a tax form. Businesses, hospitals, education systems, cities and even a nation could be considered as instances of service system. Service innovation is earnestly required to improve the current service systems by combining innovations in technology, business models, social organizations and users' demand.

This thesis discusses service innovation in the domain of Ambient Assisted Living. The population of elderly people is increasing rapidly in the last decades, which becomes a predominant aspect of our societies. As such, both efficacious and cost-effective solutions need to be sought to provide services required by an ever increasing number of users. The European Union launched the Ambient Assisted Living Joint Programme in 2008, which aims to find out an efficient solution to help elderly people living independently. Many research projects are supported by the AAL programme; meanwhile, many similar projects are also carried out in the rest of the world.

Living assistance systems and assistive devices are thus developed to facilitate the daily lives of the elderly people. These technologies show promise in helping elderly people to live independently and in comfort. However, most of the research on AAL is focused on assisting the elderly people with technologies, such as assistive devices. In order to utilize all the resources and build up efficient and cost effective systems to help the elderly people independently living, it is important to take a deeper perspective of such systems with the view of service system.

This thesis proposes to build up a mutual assistance community where services from informal care-givers (e.g. families, friends, etc.) are also included in addition to those from assistive devices. Besides passively receiving services as service requesters, the mutual assistance community also provides user with the possibility to join group activities as peer participants. Such group activities may contribute to greatly save the social resources, at the same time also helping the elderly people live in an active way.

A simplified mutual assistance community simulation model was built up for simulating community behaviours of such a system. An adaptive framework was built by combining Service Oriented Architecture (SOA) and Aspect Oriented Programming (AOP). This framework could be used to monitor environment changes and orchestrate service adaptations in the mutual assistance community. Semantic service matching is investigated, and applying semantic service matching in the mutual assistance community is also carried out in this thesis. The scope of this thesis is to prove the feasibility of the mutual assisted community concept by means of a few contributions and simulations. Constructing and validating such a community is a huge task, which is not implemented in this thesis due to resource limitation.

Acknowledgement

Now that I have finished the last words of this thesis. There are so many people I would like to thank for their help and support in finishing this thesis and their support during my PhD's study. Without your help, this thesis will never become possible.

First, I would like to express my thanks to my supervisor Dr. Vincenzo De Florio, who always provide timely support and guidance during the whole period of my PhD study. I feel guilty to occupying his time for online discussions in many weekends and evenings. I would also like to thank Professor Chris Blondia for supporting my research in the PATS group. Professor Serge Demeyer provides many advices in reading the draft of this thesis, which helps greatly in improving the thesis and I am very gratitude to. Also, I would like to thank other jury members Professors Andy Tyrrell, Bart Goethals, Elke Smits for reading my thesis and giving their valuable comments.

I would also like to thank many friends in helping me to advocate the concepts proposed in my thesis to public. Therefore many thanks to IBM for their support on my research and the helps I received from Paul Van Droogenbroeck, Martine Broekaert, Marnix Gillis and Scott de Deugd. I also thank FITCE.be and Marc Verbruggen for bringing the concept of mutual assistance community to the public medium.

I would also like to thank my colleague Ning Gui, Jonas Buys for not only sharing knowledge with me but also exchanging fun experience on chocolates and beers during these years. There are also many thanks to my friends in PATS group, Johan, Bart, Erwin, Nik, Michael, Kathleen, Nicolas, for providing a great environment in work. Many thanks to Xingfang, Xianglan, Yinghuo for helping me get used to live in Antwerp, also thanks for Wei, Jiwei and Xingjie for the happy moments we shared.

Many thanks go to my family. I am grateful for the persistent support from my parents. I also thank my dearest wife Jingjing for accompanying and supporting me to finish this thesis.

Last, but not least, I would like to thank acknowledge the educational and financial support I have received from the University of Antwerp and IBBT.

Hong Sun
Antwerp, 2011

Contents

Abstract I
Acknowledgement III
Contents V
List of Figures and Tables IX
Abbreviations Glossary XI
Chapter 1. Introduction 1
1.1 Rationale 2
1.1.1 The need for information management 2
1.1.2 Main thesis 3
1.2 Contributions of this Thesis 5
1.3 Thesis Outline 6
1.4 Publications and Awards 7
References 10
Chapter 2. Background 13
2.1 The Emerging of Service Science 13
2.1.1 Service System Engineering and Service Oriented Computation 15
2.1.2 Context Aware Services and Ambient Intelligence 18
2.2 Societal Problem (Social Challenge in this Modern Society) 21
2.2.1Societal Crisis: Elderly People Need Assistance 21
2.2.2 Solution: Ambient Assisted Living 22
References 23
Chapter 3. Mutual Assistance Community for Ambient Assisted Living 27
3.1 Related Work in Ambient Assisted Living 27

3.2 Overall Structure of Mutual Assistance Community.......................31
3.3 Simulation of Human Participation 35
3.3.1 The Simulation Model 35
3.3.2Simulation Results 37
3.4 Participant Model for Group Activities..39
3.4.1 The Concept of Participant 39
3.4.2 A model for Simulation 41
3.4.3 Simulation 43
3.5 Threat to Validity 47
3.6 Conclusions..48
References...49
Chapter 4. Service Adaptation for Mobile Devices 51
4.1 Related Issues..52
4.1.1 Current Limitations..52
4.1.2 OSGi Framework...53
4.1.3 Aspect Oriented Programming (AOP) 55
4.1.4 Binding AOP with OSGi 57
4.2 A Generic Service Adaptation Framework 59
4.2.1 Scales of Adaptation Systems 59
4.2.2 Design Issues 62
4.2.3 Global Adaptation Framework..63
4.2.4 Generic Context Aware Global Adaptation Framework........65
4.3 Experiments 70
4.4 Managing the Assistive Devices...72
4.5 Conclusions...75
References...76
Chapter 5. Service Adaptation in Healthcare Community.................81
5.1 The Evolvement of Healthcare System..81

5.2 Challenges of bringing human participation in Ambient Assisted Living systems 84
5.2.1 Dynamicity of service availability 84
5.2.2 Service Mapping 85
5.2.3 Raising User Acceptance 86
5.3 Service Orchestration in Mutual Assistance Community 87
5.3.1 Integrating different services 88
5.3.2 Architectures of Service Discovery 89
5.3.3 Semantic Service Matching 94
5.3.4 OWL-S Matcher 98
5.4 Experiments 99
5.5 Scenario 104
5.6 Conclusion 106
References 107
CHAPTER 6. Conclusion and Future Work 111
References 115

List of Figures and Tables

Figure 2.1 The Service Triangle 16
Figure 2.3 Percentage of people hampered in daily activities, by age..........21
Figure 3.1 Side Effect of Over-Using Assistive Devices.............................29
Figure 3.2 Organization of a Mutual Assistance Community 32
Figure 3.3 Simulation model of mutual assistance community....................36
Figure 3.4 Impact of Initial State..38
Figure 3.5 Real-time satisfaction..38
Figure 3.6 Comparison between traditional model and participant model ...40
Figure 3.7 Process to parse a request..41
Figure 3.8 Simulation model of mutual assistance community – participant model...43
Figure 3.9 Screenshot of one setting 44
Figure 3.10 Corresponding result 44
Figure 3.11 Experienced failures...45
Figure 3.12 Experienced delays 46
Figure 4.2 Life-cycle of OSGi Bundle 48
Figure 4.3 AOP – Express Cross-cutting Concerns 55
Figure 4.4 Improving Separation of Concerns and Modularity with AOP and SOA..57
Figure 4.5 Aspect Weaving for OSGi 58
Figure 4.6 The Evolvement of Adaptation Systems 61
Figure 4.7 Architecture of Global Adaptation Framework 64
Figure 4.8 Unifying the Adaptation Models 65
Figure 4.9 Building Resource Monitor in Bundle..66
Figure 4.10 Generic Context Aware Global Adaptation Framework 67
Figure 4.11 Structure of GUI Bundle 68

Figure 4.12 Activator of the GUI Bundle 69
Figure 4.13 Retrieving GUI Object in Aspect Bundle 69
Figure 4.14 Experiment Structure 70
Figure 4.15 Graphic User Interface 71
Figure 4.16 Adaptation Thresholds 72
Figure 4.17. Sample Electronic Health Record of a Patient's health status..73
Figure 4.18 Managing Assistive devices 75
Figure 5.1 The Evolvement of Healthcare System 83
Figure 5.2 Organization of a Mutual Assistance Community 87
Figure 5.3 Structure of service integration 88
Figure 5.4 Centralized Architecture for Service Discovery 90
Figure 5.5 The process of engaging a web service 91
Figure 5.6 Peer-to-peer Architecture for Service Discovery.........................92
Figure 5.7 Hybrid Architecture for Service Discovery 93
Figure 5.8 The Top Level Organization of OWL-S Service 95
Figure 5.9 A Sample of Service Profile Attributes 96
Figure 5.10 Algorithm for output matching..96
Figure 5.11 Hierarchical Classes in Protégé...100
Figure 5.12 Excerpts from the Advertised Service in OWL-S..................101
Figure 5.13 OWL-S Matcher Interface and Matching Result 102
Figure 5.14 Matching Result II 103
Figure 5.15 A Scenario of Organizing a "Participant Activity" 105
Figure 6.1 The Social Interaction Context of the elderly 112
Figure 6.2 Technologies to Construct Mutual Assistance Community 113

Listing 4.1 A Simple Example of Monitor and Adaptation 56

Table 6.1. Matching Degree...91
Table 6.2 Service Matching 105

Abbreviations Glossary

AAL – Ambient Assisted Living

AJDT – AspectJ Development Tools

AMI – Ambient Intelligence

AOP – Aspect Oriented Programming

AOSD – Aspect Oriented Software Development

BPEL – Business Process Execution Language

DL – Description Logic

ECG – Electrocardiogram

ESB – Enterprise Service Bus

GUI – Graphic User Interface

HCI – Human Computer Interface

ICT – Information and Communication Technologies.

JAR – JAVA Archive

OSGi – Open Service Gateway Initiative

OWL – Ontology Web Language

OWL-S – Ontology Web Language for Services

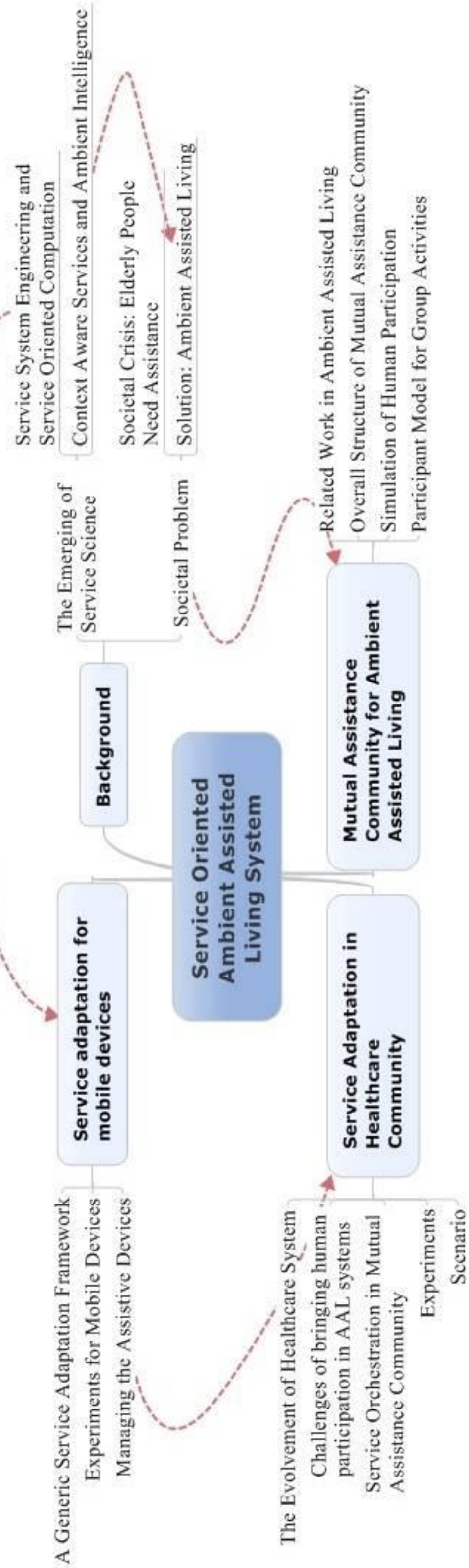
QoE – Quality of Experience

P2P – Peer-to-Peer

RFID – Radio Frequency Identification

UDDI – Universal Description Discovery and Integration

WSDL – Web Service Description Language



Note: the arrows indicate where the concepts are applied or problems are addressed.

Figure I. An Overview of the Main Topics Covered in this Thesis

Chapter 1. Introduction

Mobile digital devices such as mobile phones, PDAs, wireless sensors, etc. are now available with very low prices, and are widely used. Smart phones, which provide powerful computational ability, have also gained wide-spread popularity. These devices will increasingly be networked together to share information and provide better users' experience. In order to better share and process information between different digital devices, the concept of pervasive computing [Satyanarayanan, 2001] was proposed to build up environments where many computational devices or systems are simultaneously engaged in, communicating with each other in an unobtrusive way. The purpose is to provide the user with comfortable environments without the feeling of intrusion, thus improving the quality of our working and daily lives.

Progresses that we human beings have achieved in the past century are not limited computer technology, but also e.g. the medical sector. The advances in medical sectors help people live a longer life: the proportion of the elderly people keeps increasing in the past decades and life expectation is also rising continuously [EUROSTAT, 2004]. Prolong our lives is a great achievement, however, the increasing number of the elderly people also calls for huge amounts of resources and services to help them live in a decent way despite the increasing inconvenience that come with aging. Such requirements pose great challenges to our society's welfare system. Hence, solutions both efficacious and cost-effective need to be sought to provide the elderly people with the services they need.

Ambient intelligence, which proposes to connect smart devices together to build up a safety environment around the assisted people with the concept of pervasive computing, is considered as a promising approach to meet the challenge of the increasing number of the elderly people. Much research is being carried out with this approach, which is bringing encouraging results in assisting elderly people living safely and independently.

This thesis discusses the challenges of providing efficient and effective assistive services to the elderly people. This thesis also emphasizes the importance to integrate computer services with the services from human side so as to build a community with all the available resources, and share these resources in an effective way.

The concept and design of a hybrid Service-oriented Community, where devices and human beings with different capabilities, background, and information could be optimally orchestrated to devise intelligent responses to different situations, are introduced in this thesis. Such a community could be configured to fit for different purposes of applications, however, in this thesis, I focus on its application on building an assistance community for helping the elderly people.

1.1 Rationale

1.1.1 The need for information management

Digital devices are widely applied nowadays, which brings to new ways of expressing various contents in digital forms. Together with the increased usage of mobile devices, such digital information could be shared between connected users. Such sharing and exchanging of knowledge could assist people’s work and daily lives; however, the massive use of these digital products brings an explosion of information/digital contents, which makes it hard for people to retrieve the right information they would want or need, when they do need it. Very often such information is lost or scattered among devices and human beings that neither know how to share it nor how to put it to use. This prevents to timely orchestrate intelligent response to societal needs. Lacking this intelligence, the costs that societies sustain to guarantee their services are becoming unsustainably large – especially since the growth in age and number of our populations.

Service orientation architecture (SOA) permits to represent, publish, and look up different applications in heterogeneous service oriented structure. It builds up hybrid architectures of people and devices, each with their own capabilities and requirements; this can be used not only to intelligently manage the huge available information but also to provide people with easy access to manipulate these digital contents.

In order to better utilize advanced digital devices and fully express their potential in our work and daily lives, it is important to dynamically configure resources from people, digital devices, and organizations, to share information and deliver value to the involved people. Such dynamic configuration and organization of resources are the origin of the concept of Service System, proposed by IBM [IBM, 2008].

Inside a service system, innovative methods are required to best organize the resources. With the wide application of mobile devices, the resources inside a service system are often floating from here to there with high dynamicity. Adaptive policies are earnestly desired to cope with this dynamicity, which is to monitor the changing context, and take corresponding reactions.

In general, we require adaptive service systems which could flexibly orchestrate the resources around us, coping with the frequently changing environment, and provide the needed services in time. This thesis covers the topic of how to build up adaptive service systems, especially in the ambient assisted living domain.

1.1.2 Main thesis

The population of elderly people is increasing continuously. Thus there is an increasing demand for help from our society to maintain the independent living of their elderly ones. Much research was carried out on building intelligent environments around people, which aimed at providing assistive services in pervasive environments, constructing better environments and designing the means to support independent living. However, many fundamental issues in assisting the elderly people living independently remain open.

This thesis tries to address some fundamental problems of Ambient Assisted Living: contrarily to other approaches, which merely address the safety of the elderly ones, our focus also covers quality-of-life aspects (e.g. eInclusion, personalized services…). Moreover, it includes individual well-being as well as societal concerns. In real-life scenarios, the requests for and the availabilities of services are changing dynamically and abruptly together with changes of constraints and context environments. In order to match this very complex matter, we believe that (1) the flexibility of Service Oriented Architecture to carry out dynamic service orchestration and (2) the versatility of human tasks in providing personalized and economized services should be coupled into an adaptive hybrid socio-technical organization. We provide an example of such organization, which we call the Mutual Assistance Community, and we prove by simulation that such a system may indeed be able to match the above stated requirements.

The deployment of assistive devices in the past decades has greatly helped the elderly people to maintain independent living by monitoring their health status, providing assistive services, and in general building a safe environment around them. However, the ambient assisted living systems of the 21^{st} century must not only take care of the physical health of the elderly people, but also be aware of their psychological well-being. The usage of assistive devices helps to transfer the dependence from the human side to the machinery side, thus establishing some degree of independence. However, the dependence on the assistive devices unconsciously reduces the social connections of the assisted people. The human services could fill this gap and provide a means to strengthen the social connections between the assisted people thus helping to keep them psychologically healthy.

Hybrid approaches integrating the human services and device applications could provide services to the elderly people in an optimized way by choosing the most appropriate service providers – human or otherwise. Our social resources could also be greatly saved with such an approach. Currently, medical services are mainly carried out by professional medical caregivers. These medical resources are under great pressure with the ever-increasing population of the elderly people. Integrating human services with assistive devices could provide an economized solution: parts of the medical services could be accomplished by informal caregivers with assistive medical devices, thus the professional medical resources would be greatly saved.

Building a hybrid system integrating the human services and device applications is a complex task considering the dynamic nature of human services and the continuously changing environment of the device application. We believe the flexibility of Service Oriented Architecture could help us to implement such integration to cope with the above mentioned dynamic changes. By combing the Service Oriented Architecture and Aspect Oriented Programming, we are able to adjust the services according to the context environment to deliver better performance; connecting the web service technology with semantic service representation and matching, we are able to find appropriate services for service requester from the web.

This thesis presents the prototype of a mutual assistance community to integrate the human service and assistive devices. The prototype aims to construct a more effective ambient assisted living system for the elderly people. The system has not been implemented in real life yet due to its complexity and other constraints (a full-fledged implementation of such a community is a complex project which involves multi-disciplinary research and participation of different parties). Nevertheless, it represents my efforts of advocating a new concept in ambient assisted living. This thesis is an attempt in using innovative service models and new technologies to provide cost effective solutions in ambient assisted living. The scope of this thesis is to prove the concept of mutual assisted community by means of a few contributions and simulations. Constructing and validating such a community is a huge task beyond this thesis.

1.2 Contributions of this Thesis

The contribution of this thesis is to introduce an effective architecture to provide prompt services to the ever increasing population of elderly people. My research during my PhD study contributed on two directions: 1) Building adaptive service systems to flexibly orchestrate resources and take quick responses to the changing context. 2) Bringing service innovation into Ambient Assisted Living, where a so called mutual assistance community is proposed to seamlessly integrate services from human side and applications from assistive devices, to provide timely help to the elderly people.

This thesis presents the prototype of a mutual assistance community to integrate the human service and assistive devices to construct a more effective ambient assisted living system for the elderly people. The system is not yet implemented in real life due to its complexity and other constraints (a full fledged implementation of such a community is a complex project which involves multi-disciplinary research and participation of different parties). Nevertheless, it represents my efforts of advocating a new concept in ambient assisted living. Currently, our society is spending too much in providing care to the elderly people. As their population is increasing, governments are not able to continue providing services in conventional ways, and are required to adapt to cost effective solutions with new service modes. This thesis is an attempt in using innovative service mode and new technologies to provide cost effective solutions in ambient assisted living. The scope of this thesis is to prove the concept of mutual assisted community by means of a few contributions and simulations. Constructing and validating such a community is a huge task beyond this thesis.

The results achieved in the above mentioned directions are as follows:

A simplified mutual assistance community model was built up which could be used for the simulation of community behaviours of a mutual assistance community [Sun, 2006]. Based on such a model, I have proposed a concept named as “Participant”, to optimally help the elderly people join group activities as peer participants and reduce their dependence on social resources [Sun, 2007a]. In the view of software architecture, how to build up a mutual assistance community, the representation and processing of services are analyzed in [Gui and Sun, 2007].

In implementing the mutual assistance community, human service expression and matching in such a community is studied and implemented with the OWL-S Matcher [Tang, 2006] [Sun, 2007c]. Tentative ideas of building up the proposed mutual assistance community with serious games in the form of virtual reality are envisaged in [Sun, 2008a], where inter-reality communication were imagined.

In building adaptive system, an adaptation framework is implemented in this thesis [Sun, 2007b] [Sun, 2009b]. Such a framework flexibly combines the process of monitoring the changes and taking corresponding reactions in service oriented approach. By using Aspect Oriented Programming (AOP) into SOA [Lippert, 2008], the adaptation framework implemented in this thesis can observe changes in monitored functions and take corresponding reactions by changing settings of certain services according to predefined rules. The adaptive framework could be applied in monitoring environment changes and orchestrate service adaptations in the mutual assistance community.

Another contribution of this thesis is to raise the social awareness of the important role that human participation displayed in providing quality services in Ambient Assisted Living (AAL) systems [Sun, 2010a]. Social impact of such a mutual assistance community is illustrated in this thesis: it could help the elderly people to keep social connections and live in active ways [Sun, 2008b], which not only reduces their dependence on the social security system, but at the same time improves their psychological health.

Finally, it is worth to remark how disseminating the concept of mutual assistance, signifying the importance of human participation in AAL system, raising the social awareness that the mutual assistance activities could not only save social resources, but also help the elderly people live in active ways are also the contribution of my work. I have presented the importance of human participation and the concept of mutual assistance community in my publications and public events in the past few years. More and more researchers are now aware of the importance of the social interaction in assisting the elderly people. Once completely overlooked, exploiting the possibilities to promote intergenerational activities and make use of the knowledge/experience of the elderly people is becoming now an invaluable asset, as explicitly stated in the second call for proposals of the AAL Joint Programme [AAL, 2009].

1.3 Thesis Outline

The remainder of this thesis is structured as follows. In Chapter 2, I provide more details on the revolution of service science, the requirement for dynamic service adaptation and context awareness, and how ambient intelligence is considered as an approach to meet such needs. I also introduce the societal problem of assisting the aging population, and the efforts from the society level.

In Chapter 3, I introduce some of the related projects that focus on helping the elderly people living independently, and signify the importance to bring human participation into AAL systems. The mutual assistance community is introduced in this chapter. Its structure and organization is presented in a general way and some basic simulations of the community behaviours are presented.

In Chapter 4, I introduce the service adaptation for mobile devices. A generic context aware global adaptation framework is presented in this chapter. By coupling OSGi and AOP, the framework can flexibly manage the monitoring and adaptations across networked devices.

In Chapter 5, I introduce the service orchestration in the proposed mutual assistance community. Web service representation and service matching techniques are introduced, experiments are carried out. A scenario is also given to show how to organize group activities in such a mutual assistance community.

Conclusions are given in Chapter 6, in which also my visions on the future of such a system are presented.

1.4 Publications and Awards

Publications

Book Chapter

Building Mutual Assistance Living Community for Elderly People
H. Sun, V. De Florio, N. Gui & C. Blondia. in Jeffrey Soar, Ed., Intelligent Technologies for the Aged: The Grey Digital Divide, 2010(b). p207-219

This book chapter summarizes my efforts on building mutual assistance community for elderly people, the content of this paper are contained in Chapter 3 and Chapter 5.

Journal

A Generic Adaptation Framework for Mobile Communication
H. Sun, V. De Florio, N. Gui & C. Blondia. In International journal of adaptive, resilient and autonomic systems, volume 2, issue 1, 2011 , p. 46-57

This paper introduces a generic adaptation framework together with its application in mobile communication. The content of this paper is presented in this thesis in Chapter 4.

Toward architecture-based context-aware deployment and adaptation
N. Gui, V. De Florio, **H. Sun** & C. Blondia. In Journal of Systems and Software, Elsevier, Volume 84 Issue 2, 2011. P.185-197

An Architecture-Based Adaptation Framework for Soft Real-Time Applications
N. Gui, **H. Sun**, V. De Florio & C. Blondia. In International journal of adaptive, resilient and autonomic systems, 1:4(2010), p. 12-25

The Missing Ones- Key Ingredients Towards Truly Effective Ambient Assisted Living Systems
H. Sun, V. De Florio, N. Gui & C. Blondia. in the Journal of Ambient Intelligence and Smart Environments, volume2, number 2, 2010(a). p.109-120.

This paper emphasized the importance of human participation in constructing effective Ambient Assisted Living. The content of this paper is presented in this thesis in Chapter 3 and Chapter 5.

Towards Longer, Better, and More Active Lives - Mutual Assisted Living Community for Elder People
H. Sun, V. De Florio, N. Gui & C. Blondia. in Journal of the Institute of Telecommunications Professionals, 2:4(2008), p. 29-33. (Published in the 47th FITCE Congress firstly)

This paper draws a vision of using mutual assistance to help the elderly people have longer, better, and more active lives. The content of this paper is presented in this thesis in Chapter 3.

Conference Proceedings

ACCADA: A Framework for Continuous Context-Aware Deployment and Adaptation,
N. Gui, **H. Sun**, V. De Florio, & C. Blondia**.** in the Proceedings of the 11th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS 2009), Lecture Notes in Computer Science, Springer, 2009.

Adaptation Strategies for Performance Failure Avoidance
H. Sun, V. De Florio, N. Gui & C. Blondia. in the Proceedings of the 3rd IEEE International Conference on Secure Software Integration and Reliability Improvement (SSIRI 2009), IEEE Computer Society, 2009(b).

This paper introduces an user case of using the generic adaptation framework to avoid performance failure in communication. The content of this paper is presented in this paper in Chapter 4.

Promises and Challenges of Ambient Assisted Living Systems

H. Sun, V. De Florio, N. Gui & C. Blondia. in the Proceedings of the 6th International Conference on Information Technology: New Generations (ITNG 2009), Las Vegas, Nevada, USA, 2009(a).

This paper reviewed the promising and challenging aspects in constructing an Ambient Assisted Living System. The content of this paper is presented in section 5.1 and 5.2.

Adaptive Robot Design and Applications in Flexible Manufacturing Environments
N. Gui, V. De Florio, **H. Sun** & C. Blondia. in the 13th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 09), 2009.

A framework for adaptive real-time applications: the declarative real-time OSGi component model
N. Gui, V. De Florio, **H. Sun** & C. Blondia, in the Proceedings of the 7th Workshop on Adaptive and Reflective Middleware (ARM 08), ACM/IFIP/USENIX, 2008.

Towards Longer, Better, and More Active Lives - Building Mutual Assisted Living Community for Elder People
H. Sun, V. De Florio, N. Gui & C. Blondia .in the Proceedings of the 47th European FITCE Congress, FITCE, London 2008(b).

This paper draws a vision of using mutual assistance to help the elderly people have longer, better, and more active lives. The content of this paper is presented in this thesis in Chapter 3.

Towards Building Virtual Community for Ambient Assisted Living
H. Sun, V. De Florio, N. Gui & C. Blondia, in the Proceedings of 16th Euromicro International Conference on Parallel, Distributed and network-based Processing (PDP2008). IEEE Computer Society, 2008(a).

This paper investigates using virtual online community to create an inter-reality community to realize the targets for mutual assistance community. The content of this paper is presented in Chapter 5 and Chapter 6.

A Hybrid realtime component model for reconfigurable embedded systems
N. Gui, V. De Florio, **H. Sun** & C. Blondia, in the Proceedings of the 23rd Annual ACM Symposium on Applied Computing (SAC2008). ACM, 2008.

Service Matching in Online Community for Mutual Assisted Living
H. Sun, V. De Florio, N. Gui & C. Blondia, in the Proceedings of 3rd International Conference on Signal-Image Technology & Internet Based Systems (SITIS' 2007). IEEE Computer Society, 2007(c).

This paper carries out experiments of matching web services in Ambient Assisted Living domain. The content of this paper is presented in Section 5.3 and 5.4

Global Adaptation Framework for Quality of Experience of Mobile Services
H. Sun, V. De Florio, N. Gui & C. Blondia, in the Proceedings of the 2007 IEEE Three-Rivers Workshop on Soft Computing in Industrial Applications, 2007. IEEE Systems, Man and Cybernetics Society, 2007(b).p.121-126

This paper presented the preliminary research in constructing global adaptation framework for mobile services. An improved version of the generic adaptation framework is published in IJRAS in 2011. The content of this paper is presented in Chapter 4.

Participant: A New Concept for Optimally Assisting the Elder People
H. Sun, V. De Florio, N. Gui & C. Blondia, in the Proceedings of the 20th IEEE International Symposium on Computer-Based Medical Systems (CBMS-2007), IEEE Computer Society, 2007(a).p.295-300

This paper presented a participant model for organizing group activities for the elderly people, the content of this paper is presented in Chapter 3.

A Service-oriented Infrastructure Approach for Mutual Assistance Communities
N. Gui, **H. Sun**, V. De Florio & C. Blondia, in the Proc. of the First IEEE WoWMoM Workshop on Adaptive and DependAble Mission- and bUsiness-critical mobile Systems (ADAMUS 2007), IEEE Computer Society, 2007.p.1-6

A Design Tool to Reason about Ambient Assisted Living Systems
H. Sun, V. De Florio & C. Blondia, Proceedings of the International Conference on Intelligent Systems Design and Applications (ISDA 06), IEEE Computer Society, 2006.p.1136-1144

This paper presented a simulation tool to simulate the behaviours of an ambient assisted living community.

AWARDS RECEIVED

2009-2010 IBM PhD Fellowship
For the concept of building up mutual assistance community.

'FITCE.be Young ICT Personality 2008' competition, Number One Winner
Presentation Title: Towards Longer, Better, and More Active Lives - Mutual Assisted Living Community for Elder People

IBM Service Innovation Student Contest 2007,
Number Winner of the PhD Category, Belgium & Luxembourg

References

AAL (2009), Ambient Assisted Living (AAL) Joint Programme, Call for Proposals AAL-2009-2, ICT based solutions for Advancement of Social Interaction of Elderly People, http://www.aal-europe.eu/aal-2009-2

EUROSTAT (2004), http://epp.eurostat.ec.europa.eu: ECHP - UDB, July, 2004.

IBM (2008). Succeeding through Service Innovation: A Service Perspective for Education, Research, Business and Government. Cambridge: University of Cambridge Institute for Manufacturing.

Lippert, M. (2008)**,** Aspect weaving for OSGi, in proceedings of Conference on Object Oriented Programming Systems Languages and Applications, Companion to the 23rd ACM SIGPLAN conference on Object-oriented programming systems languages and applications.

Satyanarayanan, M., Pervasive Computing: Vision and Challenges, *IEEE Personal Communications*, vol. pp. 10-17, 2001.

Tang, S., & Liebetruth, C. (2006). *The TUB OWL-S Matcher*. Retrieved April 24, 2011 from http://owlsm.projects.semwebcentral.org.

Chapter 2. Background

In this chapter, I extend the rationales introduced in the previous section. I firstly introduce the emerging of service science and service systems; then I introduce some of their application domains, especially in meeting the social challenges brought by the demographical changes.

2.1 The Emerging of Service Science

The computational power keeps increasing for decades; following this trend, the complexity of software system is also constantly expanding, which brings difficulties in maintaining software system. To tackle this problem, the common approach is to decompose the complexity by partitioning it into different system layers, hide and process information in each layer. However, with the wide application of mobile computation, software system nowadays is quite a dynamic system which is highly dependent on the environment. Thus new software architecture is required to allow sharing the hidden intelligence and open for adaptation.

At the same time, due to globalization, business growth, technology developments and demographic changes; the scale, complexity and interdependence of today's organization bodies have be driven to an unprecedented level. Still, like the software systems, those organizations also used to break the complexity by dividing the organizations into sub-systems. Innovation approaches are required to share the hidden knowledge among the different subsystems and coordinate the collaborations between different departments.

Service science is an emerging discipline to cope with the above mentioned challenges to manage the knowledge, communication and resources inside a system. It is an interdisciplinary research domain that seeks to bring together knowledge from various areas to improve the operations, performance, and innovation either inside a software system, or for an organization [Paulson, 2006]. Service science has its stakeholders from various domains, ranging from individuals depend on complex service systems to manage those applications; companies that want to improve their revenues by re-shaping their business processes; and non-profit organizations or different level governments that wish to deliver better services by more efficiently organizing their resources, etc.

The work presented in this thesis focused on the application of service science in the domain of mobile computation and assisting the independent living of elderly people. As the dynamic nature of service oriented computing, which is easy to migrate between different systems, the architecture that developed in the mobile application scenario could also be applied in the later application to help manage the assistive devices.

Due to the fast growth in computational power and increased connectivity, mobile services are now widely available around us, but how to effectively utilize these services is still a big challenge. The great variability of resources and conditions that mobile applications have to live with calls for service science methods which are able to express such availability, optimally organize such resources and select the most appropriate services/resources for users. Service science in this specific field focuses on connecting the mobile services, discovering and orchestrating available applications/resources, and creating interactions between mobile computers and human beings, thus turning the mobile devices into a connected service system, and share the information and resources inside such a system in an efficient way.

The application of managing mobile computation is one example of applying service management in the digital regime, yet there are many applications on other fields. One of such application is restructuring the pervasive healthcare system in the social domain. The current healthcare system is actively providing the requested care services to the people in need; however, it is costly and consumes tremendous resources. With the growing number of elderly people, our healthcare system is under great pressure and is calling for an efficient solution to reorganize and save the social resources. Service science could help to reengineer the healthcare services from the service oriented perspective, which is to represent and orchestrate the resources as assets of a service system in a cost effective way. It could help to create innovative ways to deliver cost effective healthcare services, thus greatly save our social resources while still providing high quality services to people in need.

In the followings of this section, I will first introduce service system engineering and service oriented computing, which is about organizing resources with the concept of services. Then I will move to introduce context aware services and ambient intelligence, which are technologies dealing with discovery and utilization of services/resources in our surrounding environment.

2.1.1 Service System Engineering and Service Oriented Computation

2.1.1.1 Service System Engineering

Service system is the essence of service science. In [Maglio, 2008], Maglio and Spohrer signified the service system as the essence of service science by stating: "Service is the study of service systems". Service system engineering is about optimally organizing resources: "dynamic configurations of people, technologies, organizations and shared information that create and deliver value to customers, providers and other stakeholders" [IfM and IBM, 2008]. We are now living in a world where we are continuously interacting with various service systems in our daily lives, ranging from ordering a pizza online, to filing a tax form. Businesses, hospitals, education systems, cities and even a nation could be considered as instances of service system. Developing smart service system could help to optimize and efficiently save the resources inside the system meanwhile also improving service users' quality of experience.

The study of service systems aims to create a body of knowledge that describes, explains, predicts, and improves the value co-creation between interacting entities. Such methods and standards could help to break down the complexity of a service system and provide us with a clear understanding of the organization of service system, hence they drive for service innovation. Service innovation aims at improving the current service system; it combines innovations in technology, business model, social organizational and demand. Service innovation aims to have a novel combination of existing service elements to build up a new system to meet certain goals (e.g. save cost, speed up reactions, improve users' quality of experiences).

The study of service system is a new research field which involves research from multiple disciplines. The study of service system engineering could be generally considered as a combination of business strategy study and its implementations with software engineering. From the business perspective, there are studies on strategies for service innovation related on business re-engineering, supply chain, service quality, effectiveness, and change, etc.; while focusing on emerging technologies in software engineering, Web Services, Service Oriented Architecture (SOA), and Enterprise Service Bus (ESB) are the most used technologies in deploying and delivering services over the Web.

In this thesis, the proposed mutual assistance community is a blueprint of service innovation which targets on the current medical care service system for elderly people. The architecture of the system is to be implemented with service oriented approach and it aims to save the cost in providing services to the elderly people, meanwhile also improve the quality of the lives of the elderly people.

2.1.1.2 Service Oriented Computing

Service system needs to find out solutions in how to connect the different resources together and let them communicate with each other. Services are scattered around corners of the service system with various forms, e.g. applications from mobile devices, services from human beings, business processes, etc. This calls for a common language to express the services, and let them communicate with each other. Service Oriented Computing is a new computing paradigm that uses services as fundamental elements for developing applications and provide support of interconnecting services into service systems. [Papazoglou, 2003].

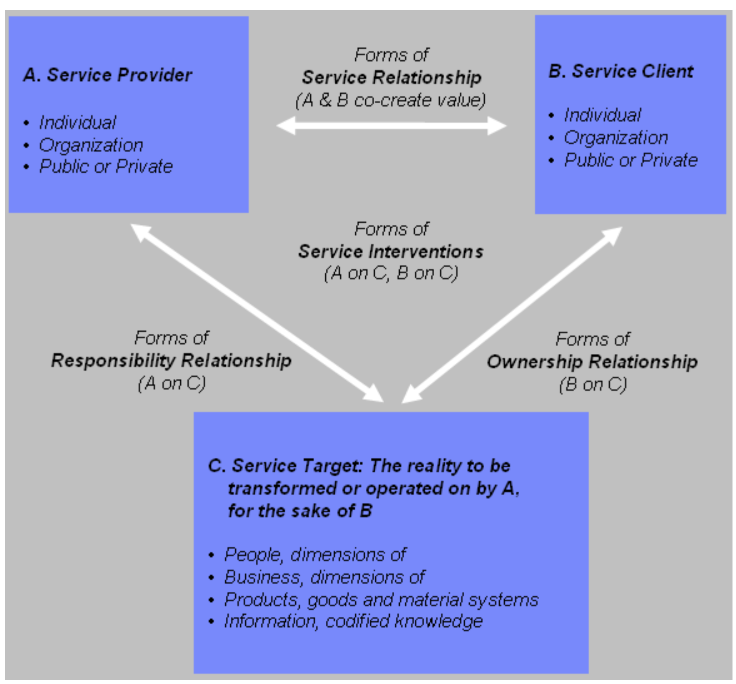


Figure 2.1 The Service Triangle [Gadrey, 2002]

Figure 2.1 categorizes the players inside a service system and summarizes the relationships between them. In Fig 2.1, a service is considered as an operation provided by A (Service Provider), transformed (or could be considered as published and customized) at C, and carried out by A to B (Customer) [Gadrey, 2002]. Gadrey pointed out that the model in Fig 2.1 should clarify: firstly, how C could transform the service from A to meet the request of B; secondly, how A and B could interact or play a complementary role in carrying out the service and bringing out the final result.

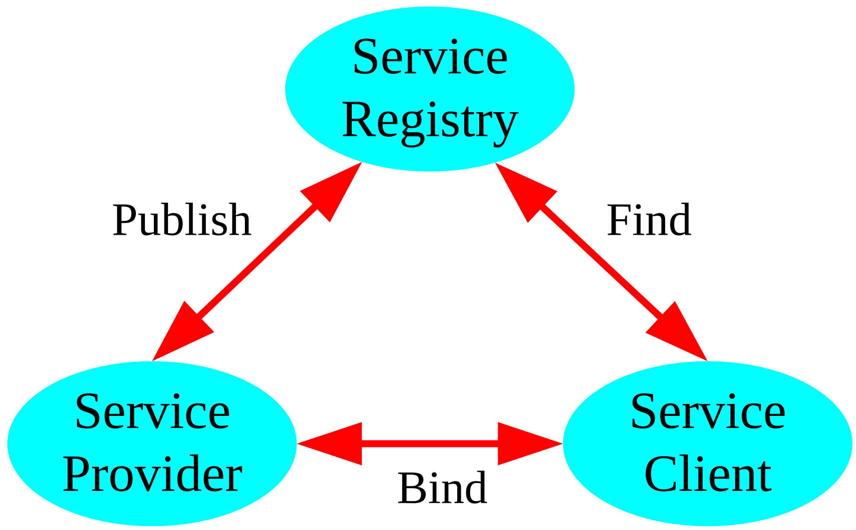


Figure 2.2 The basic service oriented architecture

Service Oriented Architecture (SOA) is a logical way of designing software systems to enable flexible development of systems based on a collection of services. Figure 2.2 shows the three basic components in service oriented architecture and their relationships. Service provider and service client are quite like their counterparts in Fig 2.1 while the service registry carries out the function as "service target". A service provider publishes its service in the service registry; a service client looks up its needed service in the service registry too. Once the published service and requested service are matched, the service provider and service client will be bond together, and service will be delivered from provider to client.

The fundamental components in SOA are services, which are well defined and self contained software entities with a discoverable and invocable interface to provide certain functions. Services in SOA are different from modules, objects, or components in other programming methods as they represent complete business function; they are also able to be assembled together to provide composite function. SOA aims to flexibly organize services which are distributed in different locations. To satisfy this goal, services in SOA are required to meet the following requirements:

- Technology neutral: Services should be easily invocable through standardized, widely accepted communication protocols.

- Loosely coupled: Services should not require knowledge or any internal structures on the client side or server side.
- Location transparent: Services should be published in repositories, from where clients can locate and invoke the services according to the service descriptions, irrespective of their location.

2.1.2 Context Aware Services and Ambient Intelligence

Key aspects in reengineering the mobile computing and restructuring the healthcare system is that they aware of the changing environment. It is important that the applications could be integrated in our daily lives in unobtrusive way and take adaptations autonomously to best meet the users' special requirements.

In "The Computer for the 21st Century" [Weiser, 1991], Weiser stated that "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." Such concept is the origin of ubiquitous computing, which could be considered as a model of human computer interaction, in which information processing has been thoroughly integrated into everyday objects and activities. In my point of view, the essence of ubiquitous computing is not about particular computational devices, but rather focused on services – it is about delivering services through computational devices in our daily lives in a seamlessly way.

Weiser noted three products as start up to build up the ubiquitous world, Tabs ("inch-scale machines that approximate active Post-It notes"), Pads ("foot-scale ones that behave something like a sheet of paper (or a book or a magazine)"), and Boards ("yard-scale displays that are the equivalent of a blackboard or bulletin board"). After twenty years' development in technology, we have a wide range of available devices and advanced wireless communication: computational powers of mobile devices are greatly improved, the sizes of computers are largely reduced, and the communication technologies are also well developed which connect the computers together into a big network. The wide applications of mobile computational devices are bringing us close to the visions that Weiser had some twenty years ago, but only if we can solve the following challenges on interaction between human, application and environment: Firstly, how to adjust applications to fit their changing environments, to get better performance? Secondly, how to adjust the environment with the presence of people, meanwhile eliminating the sense of intrusion?

2.1.2.1 Context Aware Services

Context Aware Computing is an answer towards the first question, "*how to adjust services to fit the environment?*"

Mobile computing technology is developing rapidly: mobile devices are now more with powerful computation abilities and equipped with many sensors to detect the surrounding environment to deliver tailored services. Context-aware computing is a paradigm in which applications can discover and take advantage (take appropriate adaptations/reactions) of contextual information [Chen, 2000]. Context awareness refers to the ability to adapt operations or applications to the current context environment [Dey, 2000a]. Such adaptations are expected to be carried out dynamically without explicit user intervention and thus aim at increasing the usability and effectiveness by taking account the environmental context [Baldauf, 2004].

The word "Context" in general refers to the "whole situation, background or environment relevant to some happening or personality" (Webster's New Twentieth Century Dictionary, 1980). Such a definition is too general for the application of the context-aware computing. To clarify the meaning of "context" for context-aware computing, Dey has made a highly referenced definition as follows:

> (Context is) "any information that can be used to characterize the situation of entities (i.e., whether a person, place, or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves. Context is typically the location, identity, and state of people, groups, and computational and physical objects."

This definition is more detailed compared with the definition in the Webster dictionary, however, it is still general because context awareness represents a generalized way of input, allowing many applications to be considered as context aware as they react to input. With the wide application of mobile computation, there is a trend, or rather say research effort, to coordinate the mobile devices collaboratively working together, e.g. social computing. This movement also posts a request of context awareness of social connections. In [Eugester, 2009], Eugster et.al proposed to divide the context information into two main categories:

- **Individual Context**: which gathers all parameters representing information about its environment directly accessible to the object, without interactions with other peers. It contains knowledge such as: *Where do I stand? (location) What is my profile? (resources)*
- **Social Context**: which represents awareness of the existence of other peers and context of social interactions. It contains knowledge such as: *Who is around? How close are they?*

There are many middleware platforms developed for the application of context aware computing. ContextToolkit [Dey, 2000b] uses an object-oriented approach and supports the awareness of locations of the application and its peers. CARISMA [Capra, 2003] is a context-aware reflective middleware system which focuses on detecting the system resources and remove conflicts in adaptation. SOCAM [Gu, 2004] uses service oriented approach and process context information as ontologies – it mainly focuses on processing individual contexts such as location and device resources.

2.1.2.2 Ambient Intelligence

Ambient Intelligence is an answer towards the second question, "*how to make environment adjust to the presence of people?*"

Ambient Intelligence (AmI) is a new paradigm that stems from pervasive and context aware computing, where technologies are deployed to make computers disappears in the background while moving the human users to the foreground in complete control of the augmented environment [Remagnino, 2005]. In other words, AmI refers to electronic systems that are sensitive and responsive to the presence of people, and are unobtrusively integrated into our daily environment [Aarts, 2001]. In an AmI system, distributed devices are massively embedded in the environment, which delivers information and intelligence in the interconnection network in seamless, trustworthy and natural manner.

In 2001, the European Commission launched the AmI challenge with the report of "ISTAG Scenarios for Ambient Intelligence in 2010" [ISTG, 2001], which aims to describe that living with 'Ambient Intelligence' might be likely for ordinaries people in 2010. It is stated that "The emphasis of AmI is on greater user-friendliness, more efficient services support, user-empowerment, and support for human interactions." Ambient Intelligence technologies are now widely applied in office environment to assist people finding right information [Dey, 2009], and home domain to construct smart home environment [Friedewald, 2005]. The latter application is also often accompanied with services for monitoring people's safety, and other assistance services to help the elderly people living independently [Aware Home, 2008] [I-living, 2007].

Technologies of Context Aware Computing and Ambient Intelligence could improve the interactions between the human beings and environment, thus building up a better environment to improve their quality of life. Such technologies could also be applied to building a safety environment around the user, thus are considered as solutions to tackle social problems of helping the increasing number of elderly people living independently and safe. In the remainder of this chapter, we will discuss the challenge brought by the ever increasing population of elderly people, and with introductions of how Context Aware and Ambient Intelligence, together with other approaches could make their contributions to meet such a challenge.

2.2 Societal Problem (Social Challenge in this Modern Society)

The population of elderly people is increasing rapidly, which becomes a predominant aspect of our society. As such, both efficacious and cost-effective solutions need to be sought to provide the services required by an ever increasing number of users.

2.2.1Societal Crisis: Elderly People Need Assistance

A well known trend and a predominant aspect of our society is the rapid increase of the elderly population. Several statistical reports revealed that the proportion of elderly people has kept increasing since the end of last century, e.g. EUROSTAT (EUROSTAT, 2004) indicated that “the share of the total European population (EU 15) older than 65 is set to increase from 16.3% in 2000 to 22% by 2025 and 27.5% by 2050.”

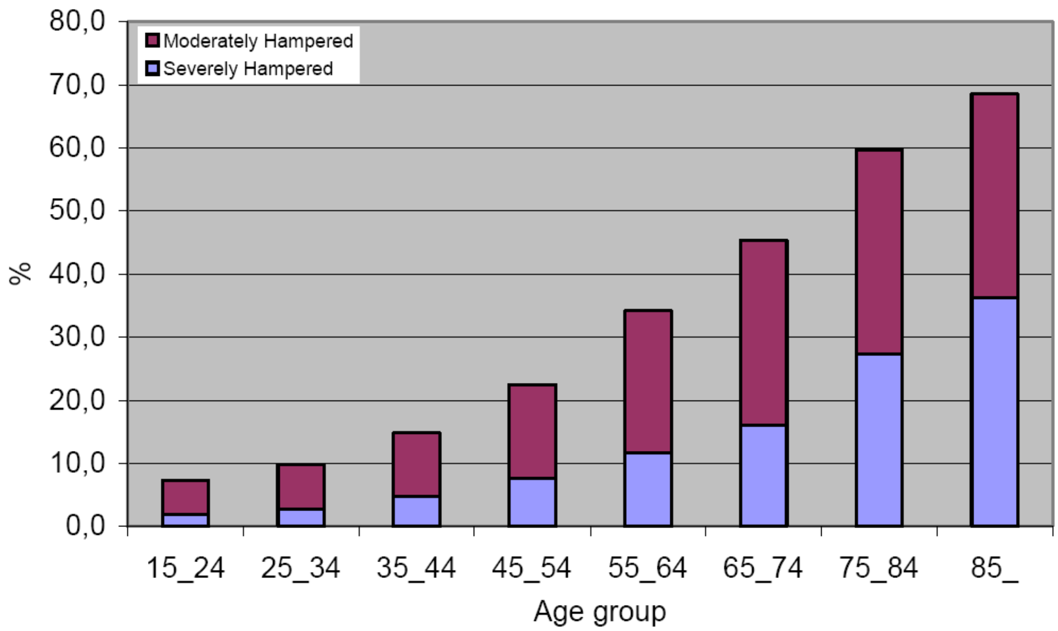


Figure 2.3 Percentage of people hampered in daily activities, by age

Independent living is indeed regarded as a key request from the elderly people. Studies of Counsel and Care in the UK found out that elderly people would prefer to live in their own homes rather than in nursing homes; thus they need support to remain independent in their home [Counsel and Care, 2005]. Figure 2.3 shows how people's daily lives are hampered by aging [EUROSTAT, 2004]. In order to improve the quality of the life for the elderly and disabled people, it is important to guarantee that assistance to those people should be timely arranged in case of need.

Research has also shown that remote clinical therapy at home will not bring negative effects to the therapy process [Deutsch, 2007]. However, one significant characteristic for the elderly and disabled people is the reduction of their mobility and social contact. The reduction of mobility also makes simple tasks, like feeding a pet or mowing the lawn, more difficult. In order to improve the quality of life for the elderly people, it is important to guarantee that timely assistance to those people is arranged when they are in request.

2.2.2 Solution: Ambient Assisted Living

The European overview report of Ambient Assisted Living (AAL) investigated the trend of increasing population of elderly and aims to find out an efficient solution to help these elderly people independently living [Steg et al., 2006]. Several research programmes that focus on AAL have been started, such as the Ambient Assisted Living Joint Programme launched by the European Union in 2008, which aims to find out an efficient solution to help elderly people living independently.

AAL aims at extending the time that elderly people can live in their home environment by increasing their autonomy and assisting them in carrying out daily life activities through the use of intelligent products and the provision of remote services including care services. As mentioned in the previous section, Ambient Intelligence put people in the center and embedded devices in the background environment to obtrusively provide services; such feature is very welcomed in building safety and responsive environment around the elderly people. In the existing projects of assisting the elderly people independent living, most efforts towards building ambient assisted living systems for the elderly people are based on developing pervasive devices and using Ambient Intelligence to integrate these devices to construct a safe environment.

Living assistance systems and assistive devices are thus developed to facilitate the daily lives of the elderly people. These technologies show promise in helping elderly people to live independently and in comfort. However, most of the research on AAL is focused on assisting the elderly people with technologies, such as assistive devices. In order to utilize all the resources and build up efficient and cost effective systems to help the elderly people independently living, it is important to take a deeper perspective of such systems with the view of service system. A service system is not only consists with technologies and devices, but also lives with the existence of human participation and those interactions and knowledge sharing between the involved parties. Similarly, a good AAL system should not only reply on the technologies and assistive devices, but also need to effectively express the power of human beings and establish good social connections and activities. Constructing the AAL system could help to better utilize the available resources bring in more services from human side and enhance the social connections of the assisted people. The rests of this thesis will present my research of building a service oriented community, which integrate applications from ambient devices and services from human beings, to effectively provide services to the people in request, and help the elderly people actively living.

References

Aarts, E., Harwig, R. and Schuurmans, M. ‘Ambient Intelligence’ (2001). In: *The Invisible Future: The Seamless Integration of Technology into Everyday Life*. McGraw-Hill Professional.

Aware Home (2008). Aware Home Research Initiative Project, Georgia Institute of Technology. Retrieved April, 2010. from http://awarehome.imtc.gatech.edu/

Baldauf, M. and Dustdar, S (2004). A survey on context-aware systems. Technical Report TUV-1841-2004-24, Technical University of Vienna.

Capra, L., Emmerich, W. and Mascolo, C (2003). CARISMA: Context-Aware Reflective Middleware System for Mobile Applications. *IEEE Transactions on Software Engineering*, 29(10):921–945.

Chen, G and Kotz, D (2000). A survey of context-aware mobile computing research. Technical Report TR2000-381, Dartmouth College, Computer Science, Hanover, NH.

Counsel and Care (2005), Community Care Assessment and Services.

Deutsch, J., Lewis, J., & Burdea, G (2007). Technical and patient performance with a virtual reality-integrated telerehabilitation system, *IEEE Transactions on Neural Systems and Rehabilitation Engineering, Vol 15*(1), (pp. 30-35).

Dey, A. and Abowd, G (2000a). Towards a better understanding of context and context-awareness. In Workshop on The What, Who, Where, When, and How of Context-

Awareness, as part of the 2000 Conference on Human Factors in Computing Systems (CHI 2000), The Hague, The Netherlands.

Dey, A. and Abowd, G (2000b). The context toolkit: Aiding the development of context-aware applications. In Workshop on Software Engineering for Wearable and Pervasive Computing, Limerick, Ireland.

Dey, A (2009). Modeling and intelligibility in ambient environments. Journal of Ambient Intelligence and Smart Environments, 1(1), pp. 47-62.

Eugster, P., Garbinato, B. and Holzer, A (2009). Middleware support for context-aware application, in Middleware for Network Eccentric and Mobile Applications, Springer Berlin Heidelberg.

EUROSTAT (2004), http://epp.eurostat.ec.europa.eu: ECHP - UDB.

Friedewald, M., Da Costa, O., Punie, Y., Alahuhta, P., Heinonen, S (2005). Perspectives of ambient intelligence in the home environment. Telematics and Informatics 22 (3), 221–238.

Gadrey, J. (2002). The misuse of productivity concepts in services: Lessons from a comparison between France and the United States. In J. Gadrey & F. Gallouj (Eds). *Productivity, Innovation, and Knowledge in Services: New Economic and Socio-economic Approaches.* Cheltenham UK: Edward Elgar, page 42.

Gu, T., Pung, H.K. and Zhang, D.Q (2004). Toward an OSGi-based infrastructure for context-aware applications, *IEEE Pervasive Computing.*, vol. 3, no. 4, pp. 66–74.

IfM and IBM. (2008). Succeeding through service innovation: A service perspective for education, research, business and government. Cambridge, United Kingdom: University of Cambridge Institute for Manufacturing.

I-living (2007). I Assisted Living Project, Retrieved on April, 2010 from http://lion.cs.uiuc.edu/assistedliving/

ISTAG (2001). Scenarios for Ambient Intelligence in 2010; Final Report, Feb 2001, European Commission.

Maglio, P. P. and Spohrer, J (2008). Fundamentals of Service Science. Journal of the Academy of Marketing Science 36(1) 18-20.

Paulson, L (2006). Services Science: A New Field for Today’s Economy. IEEE Computer
39(8) 18–21.

Papazoglou, M. P. (2003). Service-Oriented Computing: Concepts, Characteristics and Directions. In Proceedings of the Fourth International Conference on Web Information Systems Engineering.

Remagnino, P., and Foresti, G (2005). Ambient intelligence: A new multidisciplinary paradigm. *IEEE Transactions on Systems, Man and Cybernetics 35*, 1–6.

Steg, H., Strese, .H., Loroff, C., Hull, J., & Schmidt, S (2006). Europe is facing a demographic challenge: Ambient assisted living offers solutions. *Ambient Assisted Living – European overview report. VDI/VDE/IT,* Berlin, Germany.

Weiser, M (1991). The Computer for the 21st Century. Scientific American.

Chapter 3. Mutual Assistance Community for Ambient Assisted Living

The last chapter introduced the emerging of service science and service system as solutions to orchestrate resources. It was also suggested to apply the concept of service system in developing the ambient assisted living system to help the ever increasing elderly people living independently.

In this chapter, I first introduce the promises and challenges in the current approaches in building ambient assisted living system. Then I introduce the general structure of a Mutual Assistance Community for ambient assisted living together with some simulation results. Finally I present a "participant model" for organizing group activities. The implementation of the mutual assistance community is the topics of the following chapters.

3.1 Related Work in Ambient Assisted Living

Much research has been carried out on building intelligent environments around people with assistive devices, such as Aware Home [Aware Home 2007], I-Living [I-Living 2007], etc. These studies built up "smart houses" with ambient intelligence, which improved the independence of elderly people, and reduced manual work in assisting them. Devices such as RFID (Radio Frequency Identification) and motion detectors were used to monitor the status of the elderly people and the surrounding environment, detect dangerous situations, and assist elderly people in different activities of daily living. The Aware Home project included a living laboratory where user acceptance of technology was tested, and a framework for universal device interoperability in pervasive systems was built. The project also assessed devices for pervasive computing environments, medical monitoring, and human computer interaction interfaces. The mission of I-Living is similar to that of Aware Home: developing an assisted-living, supportive software infrastructure that allows disparate technologies, software components, and assistive devices to work together. Tasks provided in I-Living included activity-reminding, health monitoring, and location detection.

Applications developed in those projects are promising in providing assistive services in pervasive technology environments. They offer people better lives by assisting them with daily activities and keeping them safe by monitoring aspects of their health status and taking corresponding reactions. However, in the early days of the research in this field, the services included in those assistive environments were simple and limited; there were few communications between the involved devices/services. In such circumstance, scenarios presented in the projects at those early stages are not concrete enough to meet elderly people's needs in their daily lives and help them maintain satisfactory independent lives.

In a later stage, research efforts were made to connect different services together and coordinate them by some intelligent gateway in the smart environment to provide better services. Take Aware Home and I-Living as an example, applications reported by these two projects in the early days are rather simple, e.g. using RFID to localize personal belongings, remind patients to take medicine, etc. In the later reports made by these projects, research efforts are made in building interoperable framework to orchestrate different devices. Another project, Amigo [Amigo, 2007], though not specifically designed for assisting elderly people, investigated ambient intelligence for the networked home environment to provide attractive services and improve end-user usability. Pervasive devices were largely deployed in the Amigo project, being managed in an adaptive, context-aware and autonomous way. The scenarios of this project are sophisticated and demonstrate that it is possible to provide users with customized services by combing different smart devices together. The applications in the Amigo project are not restricted to home environment but extend to connecting work environment through mobile devices, meanwhile also connecting family members.

The Amigo project is a huge step towards general introduction of the networked home and towards Ambient Intelligence achieved by increasing the usability of a networked home system. The achievements made in the Amigo project could be applied in Ambient Assisted Living for the elderly to provide services through advanced ICT (Information and Communications Technology). However, as the purpose of the Amigo project is to deliver the Ambient Intelligence for the networked home environment, although the project provides a scenario in assisting the elderly people living home, there are still challenges to be met in order to fully express the potential of adopting those technologies to assist the elderly people living independently and safely in their houses. One of such issues is to increase human participation in the process of providing assistance to the elderly people.

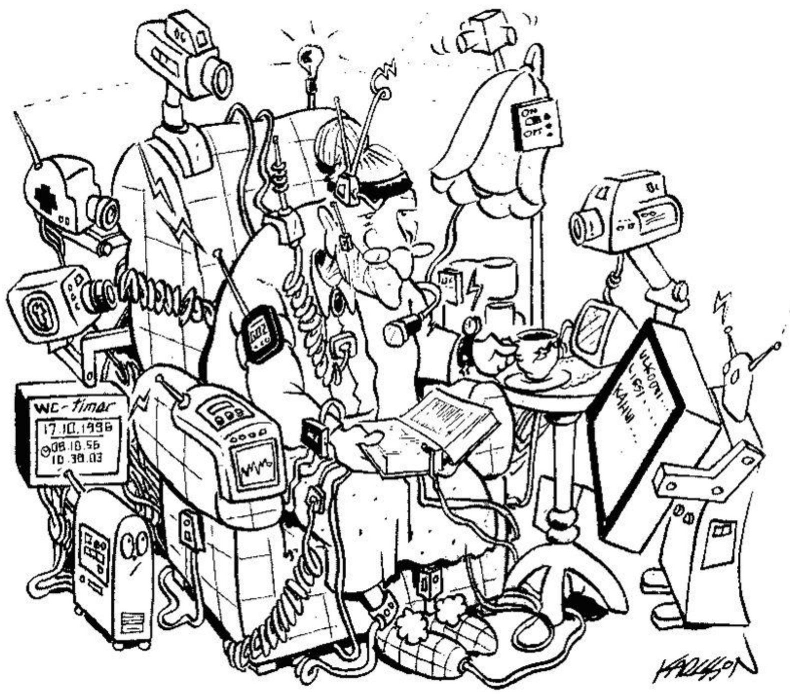


Figure 3.1 Side Effect of Over-Using Assistive Devices [Karlsson, 1996]

Human participation is essential in building effective ambient assisted living systems. The AAL country report of Finland remarked that “the (assistive) devices are not useful if not combined with services and formal or informal support and help” [Fin, 2005]. I share this view and deem informal care from relatives, friends and neighbouring people indispensable when constructing timely and cost-effective services in addition to formal care support arrangements [Sun, 2009a] [Sun, 2010a]. The use of assistive devices helps transfer dependence from the human side to the mechanical side and enhances the degree of independence. However, dependence on assistive devices can potentially reduce social connections and can mean that people are merely surviving safely, but not actively living. Figure 3.1 shows a humorous depiction of the possible side-effect of over-using technology. Although the effect is exaggerated, the picture reminds us that we should be cautious of over-reliance on assistive devices, and that we should never overlook human and social aspects.

Luckily, many researchers are now realizing the importance to include social intelligence and social connection [Aarts, 2009]. There are efforts towards building social links between the elderly people and their families or neighbours to address their needs and increase their social connections. In the scenario of Amigo project, as well as some other projects on Ambient Assisted Living (i.e. I-Living, Aware Home), there are communication channels established between the elderly people and their family (or neighbours in some projects). In case of emergency situation affecting the elderly people, signals will be sent to corresponding person, so that certain aids could be delivered. In some other scenarios, entertainments among different generations of a family (e.g. online chess between a grandson and a grandpa) are carried out online, which helps to increase social connections of the assisted person. The restriction in these projects is that the connections among the assisted people and others are fixed, which lacks of dynamicity.

There are research efforts carried out towards expanding the connections of the assisted person so as to provide more services. One of such projects is COPLINTHO [COPLINTHO 2008], which built an eHomeCare system to link the patient's family, friends and the overall care team. It aimed to make interaction possible (passive or active; images, sound and data) with all actors (home health care personnel, hospitals, the family practitioner, and if necessary an alarm centre) involved in the care process of the patient. This project focused on the recovery progress of a patient, thus on exchanging patient medical data rather than on more generalized application. This thesis further explores the possibility to expand the connections of the assisted person to allow them participate in more generalized activities.

In conclusion, much research has been carried out in building ambient assisted living environment for the elderly people to live independently. These efforts demonstrate many examples of how to use the assistive devices to build up safety environments around the elderly people. Besides using advanced technology to assist the daily lives of the elderly people, research efforts in recent years have been paying more attention in introducing human services in ambient assisted living. Connections between the assisted person and the outside world are established, which help to carry out emergence services and certain social events or entertainment activities.

In my opinion, such social connections in ambient assisted living should be enhanced to allow more flexible and broader social connections. In so doing, more human services could be involved and more social activities could be carried out to the assisted person to help them living more actively. Effective and efficient solutions to meet the challenges in Ambient Assisted Living should combine the forces from both technology and society.

The application of the ICT technology and the involvement of human participation are not contradictory with each other, in fact, they are mutually beneficial:

- On one hand, the involvement of human beings could help to fully express the potential power of smart devices, and maintain the social awareness of the elderly people.
- On the other hand, the usage of advanced ICT technology could better connect the elderly people together, e.g. organizing community activities.

In the following section, I propose a possible approach to construct a system to integrate the ICT technology and human services – the mutual assistance community. Such a system could help to extend the application domain of ambient assisted living from home to a broader community, to orchestrate the resources with community level computation. The overall structure of such a system will be introduced and some simulations will be presented.

3.2 Overall Structure of Mutual Assistance Community

Community level computation, which refers to the orchestration of community resources, has emerged since the modern communication technologies are connecting people more closely than ever before. It aims to model the behavior of our surrounding world, take the perspective of the world at the community level, and make it smart in information sharing. The digital city of Kyoto [Ishida, 2002], the proposed mutual assistance community in this thesis, and the later efforts from IBM's Smarter Planet plan [Smart Planet, 2009] are all efforts to set social information infrastructure for our everyday lives.

The community level computation interconnects people, technologies, organizations, and promotes the resource sharing amongst them. Service orchestration at this level focuses on dynamic reconfiguration of the available resources to meet the requirements from different bodies. It meets the satisfaction of a single entity, meanwhile also takes consideration of the welfare of the overall community.

Current solutions for ambient assisted living are costly. Introducing more human participation and orchestrating the resources at the community level could help to save the cost in assisting the elderly people independently living. In this thesis, I propose to build a mutual assistance community where dwellers may help each other when they are able to, supported by assistive devices which also contribute to a smart environment. Participants are able to make different contributions to the welfare of the community. The technological and social forces are more seamlessly combined, providing better services to people.

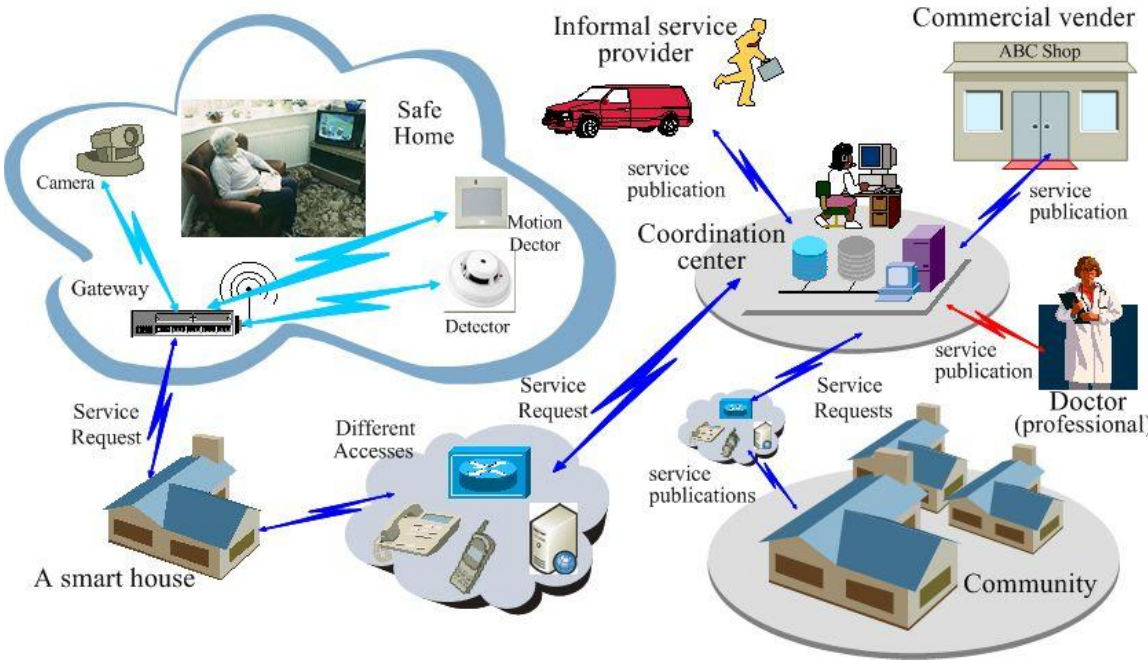


Figure 3.2 Organization of a Mutual Assistance Community

The architecture of the proposed community is shown in Fig 3.2. This combines assistive devices and ICT technologies, and provides social interactions. The main actors playing in the proposed mutual assistance community are as follows:

a) Professional Caregiver

Professional caregivers refer to those who are able to provide professional medical care, such as medical doctors, nurses, etc. Services provided by professional caregivers are more stable and reliable; however, the available professional caregivers inside a community are normally limited, so that they should be provided to the people who need them most.

b) Informal Caregiver

Informal caregivers refer to those who are willing to provide help to other peers through their own competences, normally as volunteers. Informal caregivers are normally acted by the families, friends, or neighbours of the people who need assistance. Services provided by informal caregivers are normally non-medical ones, e.g. helping with family work. In certain emergency situations, informal caregivers could also provide first aid services to the people in need. The main characteristic of informal caregivers is that services from their side are highly dynamic, i.e. their availabilities are constantly changing over time, which requires the proposed mutual assistance community with certain ability to tolerate such dynamicity.

c) Commercial Vender

Commercial venders refer to those who are providing commercial products or services for profit, such as commercial housing cleaning services, or shops selling certain products. The participation of commercial venders helps to diversify the service type and brings convenience to the user, at the same time laying the foundation for economical exploitation and self-sustainability. The services from commercial venders are normally abundant and much more availability compared with those from informal caregivers.

d) Service Requester

Service requester refers to those who are in need of services. The requested services could e.g. be medical services from the elderly people, which require medical treatments from professional caregivers. There are also requests for non-medical services, such as cleaning the garden or some social activities such as chatting, or taking a walk in a park with companions, etc. Service requests are not only restricted to elderly people – the younger generation may also post their requests for help or assistance on problems regarding e.g. their studies, work, or daily lives.

e) Assistive Devices

Assistive devices refer to those devices installed around the elderly people to guarantee their safety, such as cameras, motion detectors around the living environment and body sensors (e.g. ECG (electrocardiogram), etc.) around the body of the elderly people. These devices are to be managed by a local gateway in a flexible way (see point f).

f) Local Gateway

In the proposed mutual assistance community, local gateway is set up in the home domain managing the assistive devices, gathering and taking the preprocessing actions before sending out the information collected from those assistive devices. Abnormal situations, which normally refer to a danger situation of the assisted person, could be detected by the assistive devices, deduced at the local gateway, and sent out to the outside world. Other service requests from the elderly people may also be published through the local gateway in customized way. Details of the adaptation among the loosely coupled devices are introduced in the Chapter 4.

g) Service Coordination Center

The service coordination center is working as the “brain” of the proposed mutual assistance community. All the availabilities and locations of services will be published in the service coordination center and stored in the database. The database will correspondingly update the available services based on the appearance and disappearance of the service providers. The requests for services will also be sent to the service coordination center. The matching process between the available and requested services will be carried out in the service coordination center by a service matching engine in automatic way. Details of such processes will be introduced in the following chapters.

As mentioned already, human participation plays an important role in assisting the elderly people living well, thus the proposed mutual assistance community allows bringing in human participation from all corners of the community. The proposed community also allows disparate technologies and people to work together to help people who are experiencing the inconveniences from aging or disabilities. People located inside the community are connected together via different means of communication media, such as fixed lines, mobile phones, the internet, etc. For the elder people who are not familiar with new technology, they may access the community through advanced human computer interface (HCI) [Kleinberger, 2007]. For the younger generation, they may connect to the system by their smart phone or PDA. People who are able to provide services are encouraged to do so and assist the requesting people as informal caregivers. Professional caregivers and commercial vendors are also included, which helps to diversify the service categories.

In the proposed community, elderly people are also encouraged to participate in group activities, which not only help to maintain their physical and psychological health but also reduce the requests of professional medical resources. When people grow older, their mobility can be reduced, and they may become physically weaker. However, the elder people have accumulated valuable experiences and knowledge during their lives, and they may use such knowledge to assist other people in need. Inter-generational activities could also be carried out in this way. For example, the younger generation could act as informal caregivers by helping the elderly people with physical strength-demanding tasks. In turn, the elderly people with their accumulated life experience and knowledge could use their knowledge to assist the younger generation in solving their problems e.g. at work and during their studies. Through this process, not only would the younger generation get needed information, but the elder generation could also find a means to further contribute to society.

Our society may also benefit from these mutual assistance activities. Less assistance would be required for helping elderly people as many requests would be completed by the informal caregivers. The engagement of the informal caregivers together with those group activities help to increase the social connections of the elderly people, which may benefit their psychological health. In addition, the knowledge and experience of the elder generation may also pass along to the younger ones, which would be not only beneficial for their studies and work, but also provide the elderly people with the chance to live creatively and with greater self-esteem. The result is that the social resources are utilized in a more efficient and effective way.

Figure 3.2 shows the general structure for the implementation of the mutual assistance community. Applications from assistive devices and services from human side are both described as web services; they are both registered and processed in the service coordination center. Details of managing the assistive devices in an adaptive way will be presented in Chapter 4, information of linking the human services and requests will be stated in Chapter 5.

3.3 Simulation of Human Participation

3.3.1 The Simulation Model

In order to evaluate the effectiveness of mutual assistance community with the participation of informal caregivers, I have set up a simple cell grid model to simulate the community behavior. The characteristic of a mutual assistance community is that services from informal caregivers are highly dynamic, i.e. the availability of services is continuously changing. My simulation model reflects such dynamicity by constructing an $n \times n$ cell grid; each cell in this grid represents an individual, and the whole grid represents a community.

The simulation model is a simple model which mimics some basic behaviours from our living world into the mutual assistance community. Participants of the community are divided into three types, and differentiation of service types are not taken into account.

The individuals in this community are divided into three types – client, provider, and neutral. Client cells represent people who request help in the community. Their states will turn to neutral after the service is completed. Provider cells represent people who are providing help in the community. After one service completed, their states will either turn to neutral or stay as provider, depending on different help policies. Neutral cells represent those people who neither ask for help, nor provide help – they will not take any actions. Such an abstraction is in purpose to simplify the simulation model, more types of cells (individuals) are modeled in Section 3.4.

The above mentioned community dwellers are represented as cells with different colours in my simulation model in Fig 3.3. In order to simplify the simulation model, a cell in the community may only look up its four adjacent neighbouring cells, and once it establishes a connection with its neighbour to provide or receive service, such a connection will be established, and it will last certain steps for the service delivery.

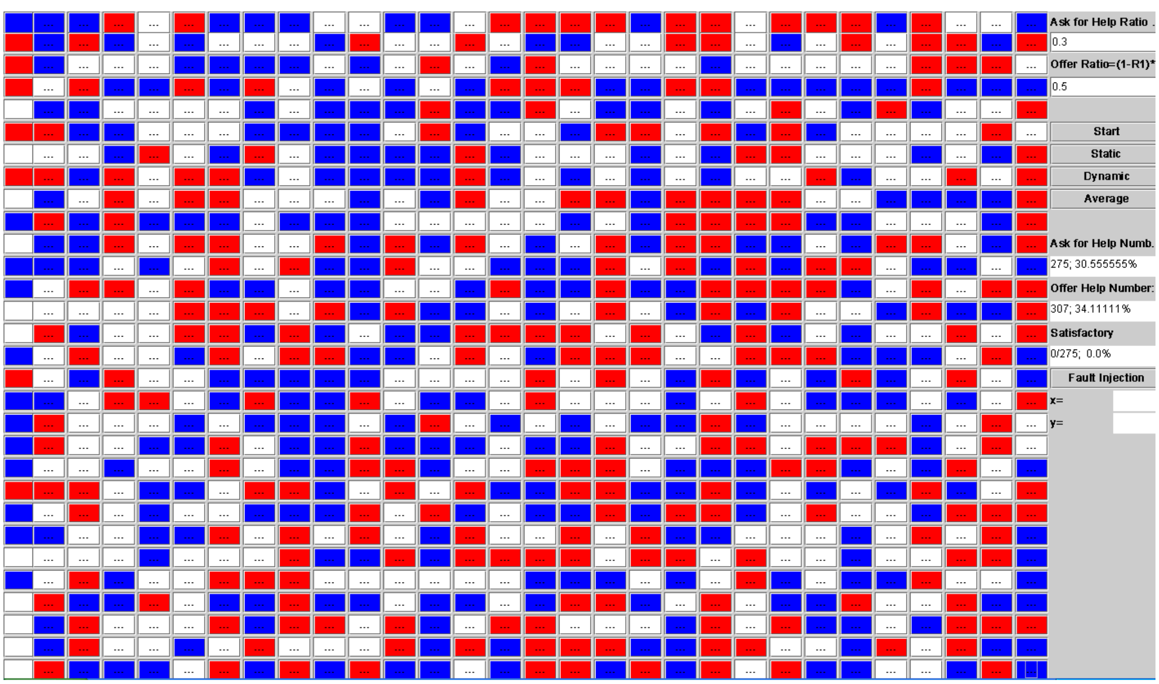


Figure 3.3 Simulation model of mutual assistance community

Fig 3.3 shows the initializing stage of the graphic interface of the simulation model [Sun, 2006]. This figure depicts the initialization of a 30×30 cell grid. The cells in blue are neutral cells; the red cells are client cells and the cells in white are the cells providing help. The rates of these three types of cells can be manually set. In Fig 3.3, the client cells account for 30.56%; the provider cells account for 34.11% and the neutral cells account for 35.33%. The satisfaction is 0 because there is no service provided in the initial stage.

The simulations of this mutual assistance living system are organized as follows: After the initialization, a small portion of the cells will be randomly chosen and change their states according to predefined probabilities. This investigation targets on investigating the performance of the system with these dynamic changes, whether the community could maintain a balance and provide service continuously and stably.

3.3.2Simulation Results

During the simulation, the community shown in Fig 3.3 changes in discrete time steps. Each activity (e.g. receives/provides service) requires a certain number of steps to finish. The needed steps to complete an activity depend on the workload of the requested service, which is a randomly generated number ranging from 5 to 25. On every step, cells who are receiving services will decrease their workload value by 1. Once their workload value becomes 0, it indicates the activity is finished: those who have received their services will be set as neutral cell, and those who have provided their services become available again. Every step, those cells that are inactive (cells not connected with other cells on service activities) will change their role with a certain probability.

The simulation firstly investigates the impact of the initial state. Figure 3.4 simulates a community organized as a 30 by 30 cell grid. The coordinate x shows the rate of the client cells in the initial state, and the coordinate y shows how many cells are still requiring help (did not receive service). After initialization, the community will develop step by step. During every step, there will be one cell randomly chosen to change its state. In this run the chance it changes to a neutral cell is 50%, while the chance it turns to be a client cell or provider cell is both 25%. The system lets the community develop 5000 steps to reach a stable state. After that, the system records periodically the number of cells still asking for help (every 50 steps). This is done for 100 cycles. The average values of the calling for help are shown in the coordinate y. Fig 3.4 shows that the initial state of the community does not take any impact on the community: after the community reaches a stable state, the rate of the cells asking for help is around 5%.

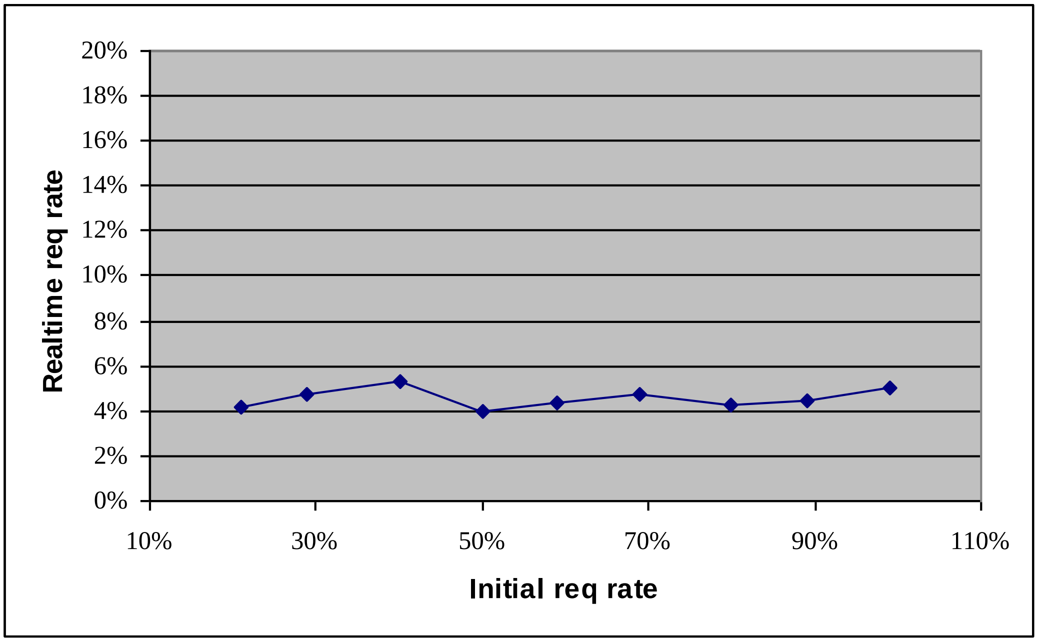


Figure.3.4 Impact of Initial State

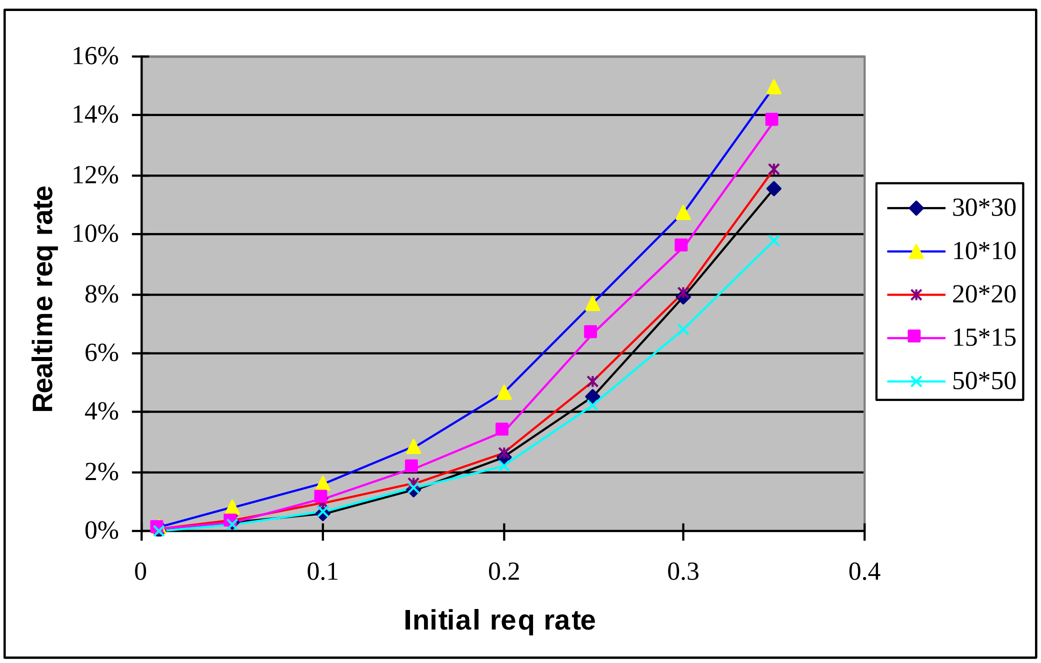


Figure 3.5 Real-time satisfaction

Fig 3.5 shows how the dynamic change of roles within the community does influence the overall satisfaction of the community. Clearly this dynamicity reflects the way a real community behaves, with its members changing their willingness to be neutral or provide help, or their need for help. Curves with different colours represent different sizes of the cell grid. The coordinate x (initial req rate) indicates the probability that a changing cell becomes a requestor of help. The probability to become neutral is both 50% in the two figures, hence that of becoming a provider is 50%-x. Every step, one cell will be randomly chosen to change its state. The coordinate y shows the rate of the cells that are asking for help.

It is shown in Fig 3.5 that the change probabilities of the cells are crucial to the satisfaction of the community: When the probability to become a client cell equals the one to become a provider, the satisfaction is around 94%, i.e. only 6% cells are not served within the community. In this simulation, ‘satisfaction’ only refers to whether services are delivered; whether they ‘satisfied’ the user is not taken into account in this simulation. When the chance to become client cell is lower than 10%, the satisfaction of the community gets over 99%.

The simulations also demonstrate that satisfactions of the people who are requesting services are largely depending on the dwellers’ willingness to provide help or participate to group activities. There are some other features observed in other simulations carried out in [Sun, 2006], e.g. the size of the community does not bring direct effect on the users’ satisfaction, however expanding the contact that a user may reach with the other users may help to increase users’ satisfaction rate. The results also suggest that a stable community with fewer changes on people’s roles may create higher satisfaction.

In conclusion, the simulations in this thesis showed that it is possible to provide the elderly and disabled with non-medical help by their neighbours promptly. The involvement of informal caregivers could reduce the dependence on professional medical caregivers significantly, thus greatly save the social resources on providing assistance to the elderly people for their independent living.

3.4 Participant Model for Group Activities

3.4.1 The Concept of Participant

The concept of participant carries on the concept of informal caregiver. It is envisaged that some activities which the elderly people want to engage in may need more than one person to participate, such as walking in the park with someone else, playing chess, etc.

The conventional approach to fulfill these requests would be, e.g. to send a nurse to the requester. With the assistance of an informal caregiver, before requesting the nurse service, informal caregivers would be sought and, if successfully identified, would be asked to provide help. The introduction of the informal caregiver could save the professional medical resources. But the elder people are always passively receiving help and losing their independence in both situations.

In addition to bringing in the informal caregivers, the concept of participant provides another solution to meet these requirements of the elderly people [Sun, 2007a]. When elderly people want to initiate or join a group activity, they will send a request to participate in this activity. The request will be parsed by the Service Coordination Center as shown in Fig 3.2. If such a participation event is ongoing, the requester could join this activity directly; otherwise, based on the time constraint of the requester, the system will either initiate a new joint activity or try to find service from informal or professional care-givers to fulfill the user's requirement.

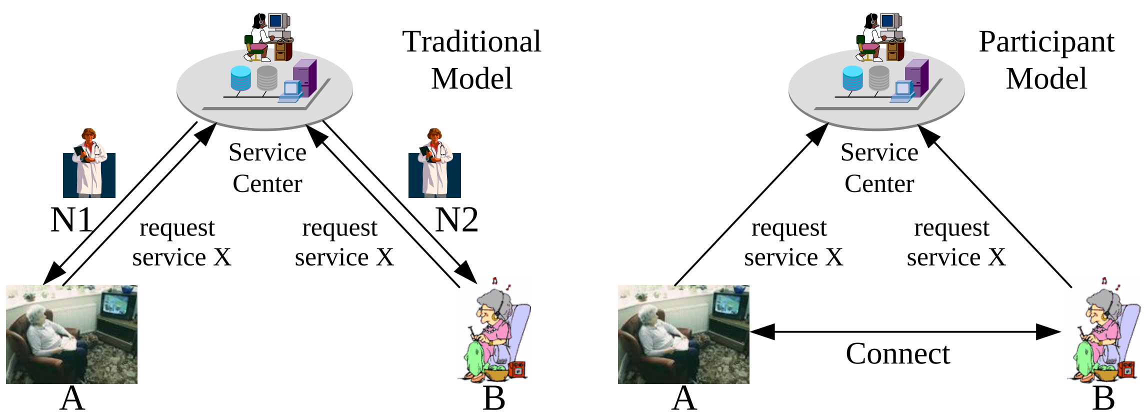


Figure 3.6 Comparison between traditional model and participant model

Figure 3.6 compares the participant model with the traditional one. A and B represent elderly people, N1 and N2 represent care-givers. When A and B want to participate a same event, the service center will try to establish a link between them in the participant model rather than requesting for help in the tradition model. This not only decreases social costs, but also encourages social contacts and produces self-esteem.

3.4.2 A model for Simulation

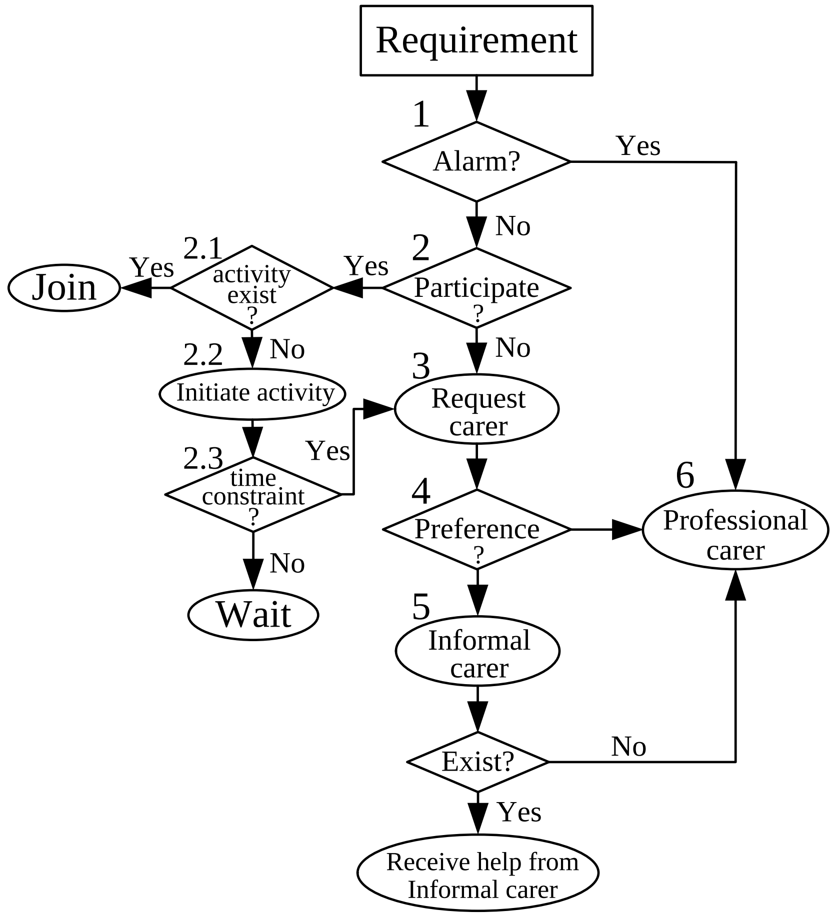


Figure 3.7 Process to parse a request

Figure 3.7 shows the process to parse a requirement in my proposed mutual assistance community. The requirement from a requester is divided into three categories: alarm signal, non-urgent request and activity participation request. The alarm signal may be triggered by the user or assistive devices – it has the highest priority. The participant signal represents the people’s willingness to participate an activity, and has the second highest priority. The non-urgent request for care is left with the lowest priority.

The requirement of an elder person could be fulfilled by the following three players: Professional Caregiver, Informal Caregiver and Participant. The professional caregivers and informal caregivers are as defined in the previous section, and the “participants” refer to those who are willing to join certain activities, such as chatting with someone else together or taking a walk in a park with other companions. In such situation, they either join an established group activity, or initiate one. People who are joining a participant activity are neither givers, nor receivers, but peer-level players. They neither get reward nor need to pay, but interactively "play" with other people in the engaged participant activity.

The process of parsing a user's request with the "participant" approach is as follows:

1. Once a requirement signal is received, check whether it is an alarm, if yes, turn to professional caregivers for help, otherwise, turn to step 2.

2. Check whether the requirement signal is a participant signal, if no, resort to helps from professional or informal caregivers, otherwise, turn to group activities.

    2.1 Check whether there is ongoing participation activity as required; if so, join this activity, otherwise, initiate one.

    2.2 A new participation activity is initialized as the user required.

    2.3 Check whether the time constraint is met; if so, turn to help from caregivers, else, keep "wait".

3. The requirement will be treated as a non-urgent request.

4. The service center looks for a caregiver for the requester based on the requester's preference. This can either be a professional caregiver or an informal caregiver.

5. The requester prefers informal care. If an informal caregiver is found, the requester receives help from that informal caregiver, otherwise, choose professional caregivers.

6. The requester prefers professional care. Professional care is assumed to be available and able to fulfill any kind of request.

3.4.3 Simulation

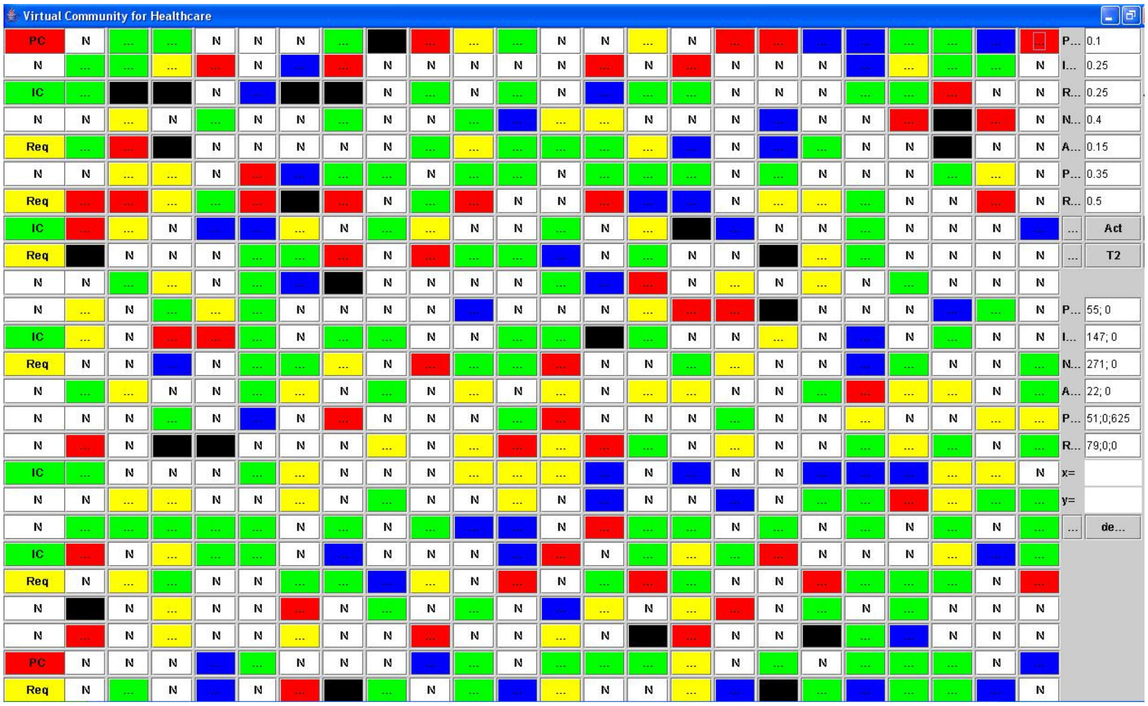


Figure 3.8 Simulation model of mutual assistance community – participant model

The system proposed in this thesis was again modeled by a 25 by 25 cell grid. Each cell in this grid represents an individual, and the whole grid represents a community. All the cells in the simulation are constructed homogenously with a same prototype. According to the predefined rates, every cell is randomly assigned to different states in the initialization stage. The community changes in discrete time steps, and each activity (e.g. receives/provides service) takes a certain number of steps to finish. The steps needed to complete an activity depend on the workload of the requested service, which is a randomly generated number ranging from 5 to 25 (Tickers for those activities requesting for "participant" group activities are all set as the same value, which allows both parties involved to finish at the same time step.). On every step, cells who are receiving services will decrease their workload value by 1. Once their workload value becomes 0, it indicates the activity is finished: those who have received their services will be set as neutral cell, and those who have provided their services become available again. Neutral cells will randomly change their states after initialization according to predefined probabilities. The simulation was to investigate how the system behaves with different rates of different cells.

The simulation model simplifies the categories of the requested services and available services, though the categories are complex compared with those categories shown in Fig 3.3. The community dwellers are divided into the following categories:

a) Professional Caregiver (PC): those who are able to provide professional medical care.
b) Informal Caregiver (IC): those who are willing to provide help on non-medical services.
c) Neutral (Ne): those who neither need service, nor provide it.
d) Service Requester (SR): those who need services, they are further divided into three sub categories:
   1) Alarm (A): requests need to be served by professional caregiver.
   2) Normal Request (R): requests could be served by either informal or professional caregiver.
   3) Participant (P): requests to join group activities as peer participant.

In the simulation, with every specified setting, the system is run for 10000 steps after initialization. The status of the grid is recorded every 10 steps, thus 1000 results are produced for every setting. The average of these 1000 results will be used to reflect the system performance under different settings.

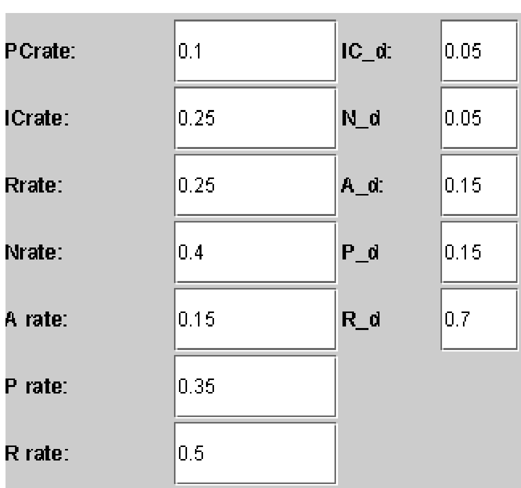


Figure 3.9 Screenshot of one setting

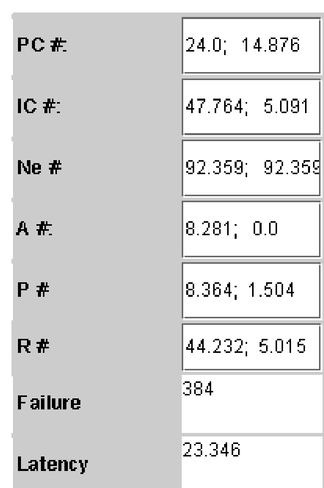


Figure 3.10 Corresponding result

Figure 3.9 shows the screenshot of one simulation setting. The setting shown in Fig 3.9 is a very stringent situation: In the beginning, the rate of Professional Caregiver, Informal Caregiver, Service Requester, and Neutral are 10%, 25%, 25%, and 40% respectively.

Among the Service Requester, alarm requests (A) count for 15%, participant (P) requests count for 35%, and non-urgent requests (R) count for 50%. After initialization, there will be 5 cells randomly picked to randomly change their cell types (the selection excludes the Professional Caregiver and cells who are actively interacting). The selected cell will have 5% probability to be an Informal Caregiver, 5% to be Neutral, and 90% to be a Requester. Clearly this is a challenging situation as the probability to be a requester is rather high. When the cell is selected to be a Requester, the probability to have an alarm request is 15%, that of a participating request is 15%, and that of a non-urgent request is 70%.

Figure 3.10 shows the corresponding status of the community, based on averaged values, at the end of the simulation run. The indexes labeled with "#" show the averaged number of Professional Caregiver, Informal Caregiver, etc. In each field, the first number indicates the average number of people, while the second one means the number of cells which are not active. For example, after a simulation run on the average the community hosts 47.764 informal caregivers, 42.673 of which are providing help while 5.091 are not interacting (that is, they are available). The number of failures indicates the total number of aborted requests accumulated in 1000 results; this is not an averaged value. Field "latency" indicates the averaged steps that all the requester cells had to wait in every step.

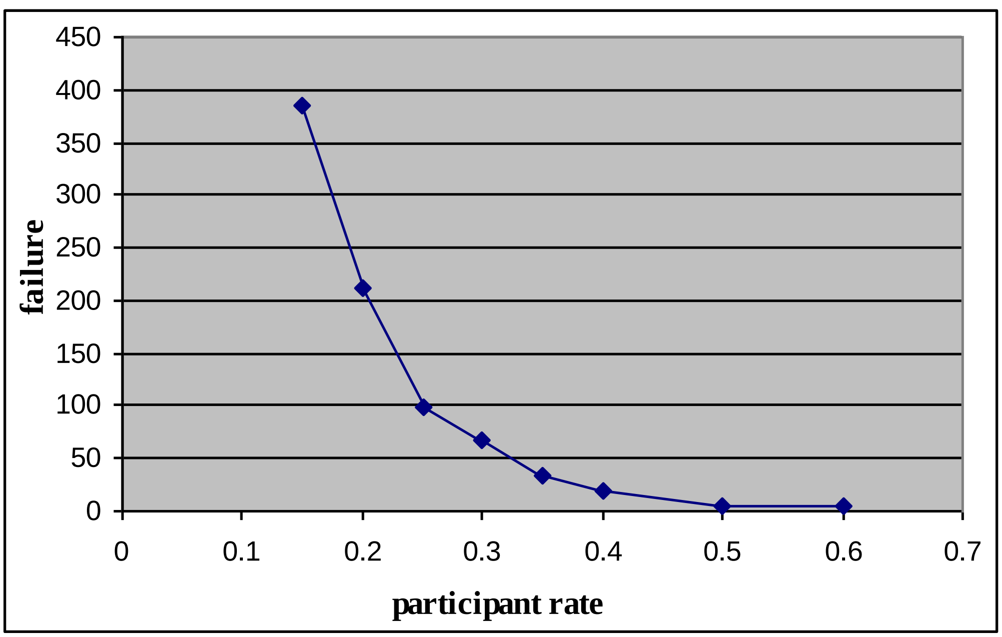


Figure 3.11 Experienced failures

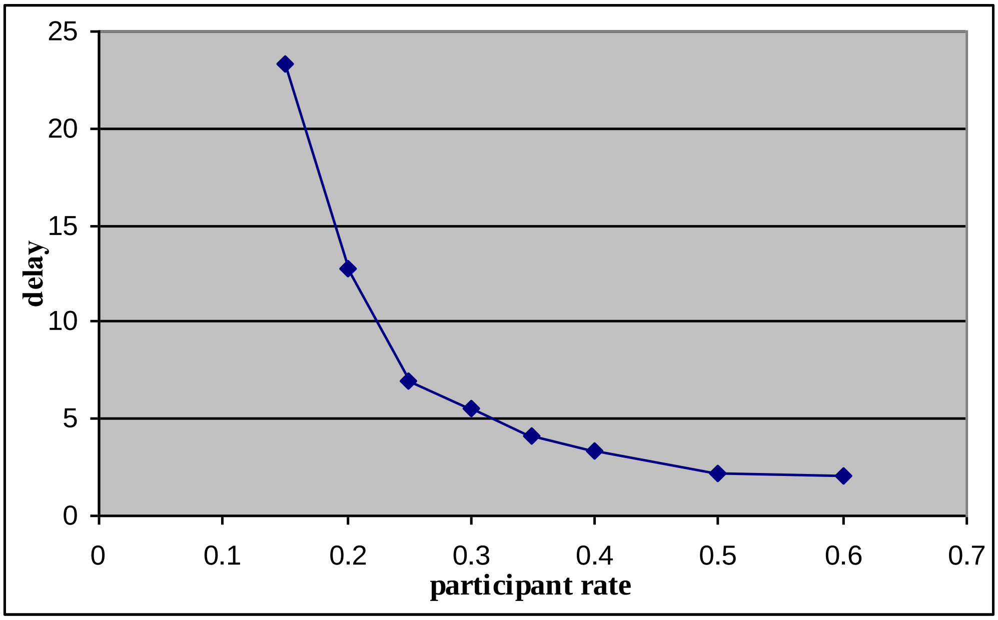


Figure 3.12 Experienced delays

The amount of failures and delays experienced is shown in Fig 3.11 and Fig 3.12 respectively. The setting of the system follows the values in Fig 3.9 except for the participant rate and R rate change. The participant rate here indicates the probability to send a participating request if a cell is selected to change its type into a requester, and the R rate represents the probability to send a non-urgent request.

The main conclusion of the simulations is that both the failure numbers and the latency decrease with the increase of participant rate. When the participant rate is 15%, there are 309 failures, and the average latency is 23.3 time steps. When the participant rate increases to 25%, the failures drop to just 99, and the averaged latency drops to 7.0. When the participant rate reaches 60%, the failures further drop to 5 and the averaged latency drops to 2.0.

These two figures indicate that when the elderly people are actively participating is group activities, the dependence on social resources are further reduced, both the failures and service latency (i.e., the time difference between service requests and their response, if any) go down considerably. When the participant rate is low, its impact is very significant: a tiny increase on the participant rate will bring great improvement. What was not shown in the above figures is that when keeping the other settings but decreasing the participant rate to 10%, failures increase to 987, and the latency will reach 4864.8, which means the community becomes unstable and chaotic.

3.5 Threat to Validity

In this section, I identify factors that may affect the simulation results in section 3.3 and Section 3.4. Some actions to alleviate the risk associated with those factorsare also presented. Following the guidelines from [Runeson, 2009] and [Yin, 2002], I examine the threats to validity as expressed by the following four aspects:

Construct validity – to what extent the experiments represent what the researcher intend to investigate?
The experiments in Section 3.3 and 3.4 intend to mimic the general behavior of a mutual assistance community where informal caregivers and participants are involved in. For this purpose, a cell grid model is set up to imitate the behavior of a mutual assistance community. However, the modeling process largely simplified the dynamicity of the roles that a human could play as well as various types of services involved in typical daily lives. Such simplification may jeopardize the precision of the experiments. However, as the experiment is not intended to model the behaviors of a complex mutual assistance community; instead, it focuses on assessing the impaction of introducing informal caregivers and group activities. Standing on this premise, the simplification of the complexity behaviors in a mutual assistance is acceptable.

Internal validity – are there unknown factors which might affect the outcome of the experiment?
The matching algorithm between service requester and service provider may influence the outcome of the experiment. It is difficult to select one player from the pool of service providers to a service requester. In real application, such a service matching process should be conducted based on evaluations of many factors, e.g. exact service categories, location between service providers and service requesters, .etc. However, it is not the purpose of this simulation to take all these considerations into account. This is because semantically matching the provided and requested services is already a very complex challenging task; it is difficult to introduce such a complex matching in every step for each player in the simulated community. In this simulation, service matching is simplified by assigning a randomly generated *matching value* to each player (e.g. service requester, service provider) when it is initialized. During each step, those free (not bound) service providers are sorted based on their *matching value*. Meanwhile, similar action is carried out for the service requesters. The service provider with highest *matching value* is

matched with a service requester who also with highest *matching value*, and so forth. A more complex service matching algorithm based on semantic matching algorithm is also studied and is presented in Section 5.3.

External validity – to what extent it is possible to generalize the finding?
Users' willingness to offer/look for services through such a system would affect the simulation results. Although the important role that informal caregivers could play in Ambient Assisted Living is now widely recognized [Broek, 2011], the availability of informal caregivers might be still very limited due to several factors – e.g. occupation on other businesses. Moreover, the willingness and ability of elderly people to use this community to look for their needed services might also be low especially in the initial stage of deployment of such a community. In order to cope with this uncertainty, the simulation covered a wide range of possible engagement rates of the players. In addition, Fig. 3.11 and Fig. 3.12 summarized the performance achieved when setting different relative rates for the willingness of requesting a service and participating a group activity. The main lesson learned is that the involvement of group activities may greatly improve the performance.

Reliability – to what extent the data and analysis are dependent on the specific researcher?
The simulation model is built with Eclipse development environment in Java. However, the simulation and analysis is general and not dependent on any specific software. The manual setting of service matching algorithm is highly dependent on the researcher who is carrying out this experiment, so it is likely that other researchers will conduct different observations. In real life application, the diversity of the provided and requested services is tremendous, it is difficult to provide complex matching algorithm in the simulation model. Cases where service requester could not find appropriate service provider would arise in real application, and improving service matching by introducing semantic service matching is discussed in this thesis in Section 5.3.

3.6 Conclusions

This chapter presents the concept of building mutual assistance community to help the elderly people living independently. The importance of human participation is noticed and emphasized in such a community. The general structure of such a mutual assistance community is presented. The simulations testing the function of informal caregivers are carried out, which confirms their effectiveness. This chapter also presented a participant model which aims at promoting the participation of the elderly people to the group activities. Simulations also demonstrate such activities could further reduce the dependence on social resources.

In the following chapters, I will present more technical details on adaptively managing the assistive devices, the publication and searching of available human services, and the proposed approach of integrating the applications from devices and the services from human side.

References

AAL Finland (2005). *Ambient assisted living*. Country report, Finland.

Aarts, E., de Ruyter, B. (2009), New research perspectives on ambient intelligence. Journal of Ambient Intelligence and Smart Environments, 1(1), pp. 5-14.
Karlsson, R. (1996). Cartoon printed in *Helsingin Sanoma，October 18.*

Amigo (2007). Amigo: Ambient intelligence for the networked home environment. Retrieved August 24, 2009, from http://www.hitech-projects.com/euprojects/amigo/.

Aware Home (2008). Aware Home Research Initiative Project, Georgia Institute of Technology. Retrieved August 24, 2009. from http://awarehome.imtc.gatech.edu/.

Broek, et.al (2011). Ambient Assisted Living Roadmap, VDI/VDE-IT, 2011.

COPLINTHO (2008). COPLINTHO Project, IBBT. Retrieved August 24, 2009, from https://projects.ibbt.be/coplintho/.

Ishida, T (2002). Digital City Kyoto, *Communications of the ACM*, Volume 45 , Issue 7, July.

Kleinberger, T., Becker, M., Ras, E., A. Holzinger, and P. Muller (2007). Ambient Intelligence in Assisted Living: Enable Elderly People to Handle Future Interfaces, *Universal Access in Human-Computer Interaction. Ambient Interaction, Part II, HCII 2007,* pp. 103-112.

Runeson, P., Host, M., and Alshayeb, M. (2009). Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering.

Smarter Planet (2009), IBM, http://www.ibm.com/smarterplanet/us/en/, retrieved on April, 2010

Yin, R. K. and Alshayeb, M. (2002). Case Study Research: Design and Methods, 3 edition. Sage Publications.

Chapter 4. Service Adaptation for Mobile Devices

As introduced in the last chapter, digital mobile devices are widely applied nowadays; there is a trend to connect these devices together as a network to provide better services to the end users. However, due to the diversity of devices, connection methods, and the changing surrounding environment, architects are less able to anticipate and design interactions between the components and configure them correctly at design time, leaving such issues to be dealt with at runtime [Kephart, 2003].

Many research efforts are made aiming to improve the adaptability of systems and devices, which help them to take the right configurations that match context changes. Approaches such as autonomic computation are developed to construct self-governing computing systems that can manage themselves given high-level objectives from administrators [Kephart, 2003]. Other similar techniques such as context aware computing and ambient intelligence have been also attracting much research interest in the past decade as they could help to build systems which are aware of the changes of the application environment and either adapt the devices or change the surrounding environments to improve users' quality of experience (QoE).

In this chapter, I propose an adaptation model to flexibly organize the monitoring process of system resources with the combination of Service Oriented Architecture (SOA) and Aspect Oriented Programming (AOP), which provide an easy access for the users to specify their preferences. Adaptation logics are separated from applications and controlled by the system, while the reactions are still carried out on applications. Such a framework could flexibly implement, modify, and execute adaptation policies base on users' preferences, thus helping avoid e.g. performance failures and increase the users' quality of experience.

The rest of this chapter is organized as follows: I first discuss some of the related techniques in developing adaptation schemes for distributed mobile devices. I also present my studies of service adaptation in OSGi (Open Service Gateway initiative) environment, which includes a generic adaptation framework, its application in taking adaptations over mobile communication applications.

4.1 Related Issues

4.1.1 Current Limitations

With the development of technologies, the functions of the mobile terminals are becoming more and more powerful; accesses to pervasive services become available in many areas. Users with powerful mobile terminals are granted with the ability to access information and multimedia services from almost anywhere at anytime.

However, there are still some restrictions degrading the user's quality of experience (QoE) due to the limitation of the available resources: examples include a high CPU usage impacting the system's ability to support burst computation requirements; or an environment that is changing when the user is roaming, thus affecting the available bandwidth, etc. In order to increase the user's QoE and avoid performance failures (also known as late timing failures [Cristian , 1991]), the mobile terminal should intelligently detect the changes of environment (battery capacity, bandwidth usage, CPU usage, etc.), and take proper adaptations.

Many adaptation strategies are already developed in different domains, such as switching transcoding methods, or adjusting the tasks' priorities. However, adaptations are highly application-specific or domain-specific, which may bring conflicts between different application domains. For instance, in the power consumption domain, reducing the power could prolong the battery life, while in the communication domain, increasing the power may guarantee a better communication quality. Another feature of these application-specific adaptations is that they are normally hard bound with the application code and lack of flexibility to make reconfigurations in different environments.

Rather than scattering the adaptation logics in different applications and represent them as low-level binary code, architecture-based adaptation uses external models and mechanisms in a closed-loop control fashion to achieve various goals by monitoring and adapting system behavior across application domains. A well-accepted design principle in architecture-based management consists in using a component-based technology to develop management system and application structure [Kon, 2002] [Sylvian,2008] [Costa, 2007]. However, in most of the above mentioned approaches, the configurations of applications and components are generally carried out before runtime; these systems still need to improve their ability to take runtime reconfigurations according to the change of system status.

The adaptation framework presented in this thesis addresses the above mentioned limitations. By combining the Open Service Gateway Initiative (OSGi) and Aspect Oriented Programming, the framework is able to provide a flexible way to manage the devices and applications, and carry out the adaptations during the runtime. In the following, I will introduce the techniques of OSGi framework and Aspect Oriented Programming (AOP), together with the benefits to integrate these two techniques.

4.1.2 OSGi Framework

OSGi is a light-weight standardized service management platform, it allows for dynamic service provision from multiple providers. The OSGi platform support for Java programming language and implements a complete and dynamic component model. Applications or components can be remotely installed, started, stopped, updated and uninstalled without requiring a reboot.

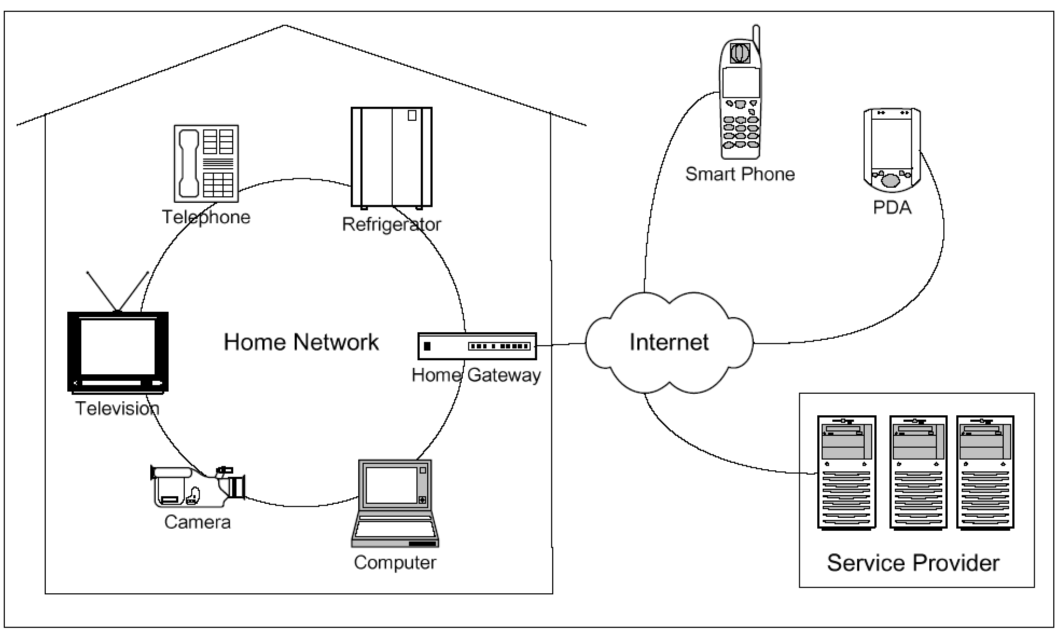


Figure 4.1 Typical Structure of OSGi Gateway for Smart Home

The initial plan of OSGi is to work as home gateways to connect and manage the household devices. Figure 4.1 shows an example of such implementation [Hall, 2004], where the home gateway acts as a mediator between the end user of the devices and the service providers who want to provide the services to the devices. Many applications are developed with OSGi gateways to provide secured living environments [Gu, 2004].

The OSGi framework has moved beyond the original focus of service gateway, it is now expanded to include any networked environment, such as mobile multimedia application [Eikerling, 2002], automotive telematics [Zhang, 2004], etc.

The adaptation framework in this thesis combines the OSGi framework with AOP to further improve its adaptability. There are many OSGi frameworks existing, e.g. Knopflerfish, Equinox, Apache Felix, etc., the framework used in this thesis is Equinox.

In the OSGi framework, modularized units, e.g. services and applications, are defined as services and represented as bundles. A bundle is comprised of Java classes and other resources, which together can provide functions to end users. In the OSGi framework, a bundle is built as a JAR file that contains the resources necessary to provide some functionality together with a manifest file describing the contents of the JAR file and providing information about the bundle [OSGi, 2007].

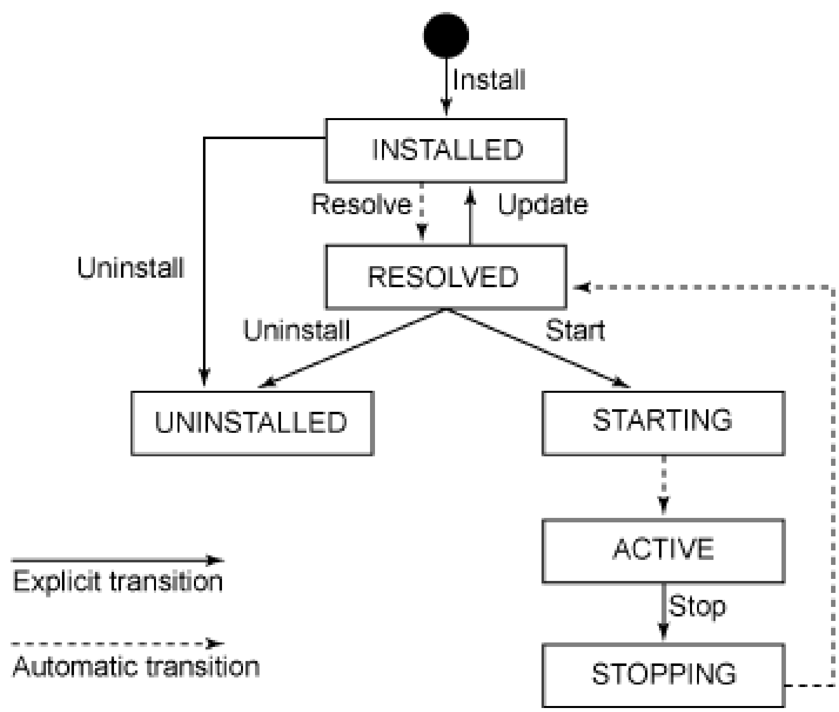


Figure 4.2 Life-cycle of OSGi Bundle

Figure 4.2 shows the states that a bundle might experience in its life-cycle, together with the transitions between these different states. Once a bundle is installed, it must be resolved before it can he used in any way. The framework resolves a bundle by checking its deployment manifest for dependencies on external Java packages. If the bundle has dependency on other bundles or packages, the framework will check whether the required ones exist. Once a bundle is resolved, it can be activated or stopped afterwards.

Adopting the OSGi framework allows the applications to be organized following the service oriented approach (they are built as OSGi bundles); the framework is able to flexibly manage the components/application in loosely coupled way. In the adaptation framework to be presented in the following sections, I will also use the OSGi framework to wrap and organize the applications as OSGi bundles.

4.1.3 Aspect Oriented Programming (AOP)

As well known, Object Oriented Programming (OOP) improved the modularity of programming; it addresses common concerns in the code: those attributes and behaviours that are common on related entities are captured and built as properties of a class or interfaces. However, on the other hand, OOP also has its limitation in localizing concerns (properties of areas of interest) involving global constraints which refers to concerns that might be included by different classes, which may tangled into other structural elements and become a mess [Elrad, 2001].

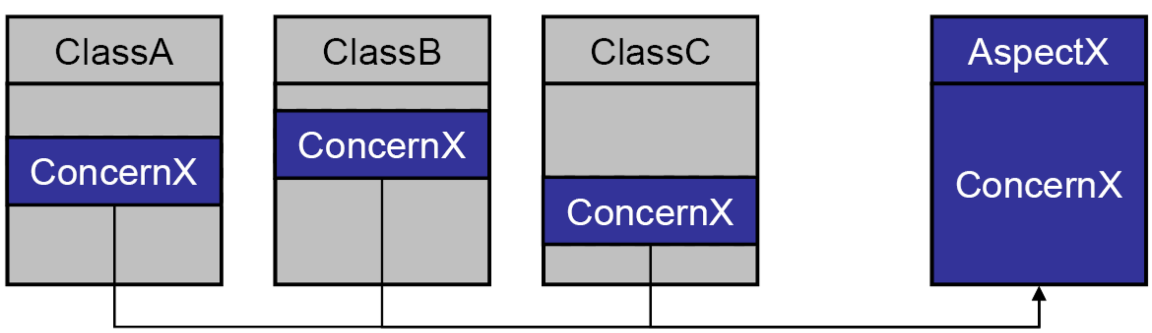


Figure 4.3 AOP – Express Cross-cutting Concerns

Aspect Oriented Programming (AOP) is a technique to design and address the cross-cutting concerns [Kiczales, 2007] [Kiselev, 2002]. AOP attempts to realize scattered concerns as elements of a separated Aspect class, and eject them horizontally from the object structure. Figure 4.3 shows the logic that AOP separates cross-cutting concerns. In the traditional methods of OOP, the concern X would be coded and processed as sub-programs in Class A, B and C respectively. They would be scattered and tangled, and there might be conflicts between each operation. AOP helps to better express the cross-cutting concerns in a horizontal way: the concern X in those three classes is addressed together in the AspectX class, which avoids the tangling concerns.

The AOP technique has been implemented in many programming languages, e.g. C and C++, Java, Delphi, Python and .etc. The programming language used in this thesis is mainly Java; AspectJ is an open-source tool that implemented AOP in Java [Kiczales, 2001] [Kiselev, 2002], there is also a related Eclipse project AspectJ Development Tools (AJDT) project that provides tool support for Aspect Oriented Software Development (AOSD) with AspectJ in the Eclipse platform [http://www.eclipse.org/ajdt/].

The AOP technique could well support the adaptations by monitoring and processing the cross-cutting concerns. Sensitive (or interested) attributes scattered around the system could be gathered together and get processed. Changes of these attributes could be monitored by the AOP constructs and corresponding adaptations could also be carried out by the Aspect class.

Listing 4.1 A Simple Example of Monitor and Adaptation

```
1  pointcut CPUMonitor(float value) :
2      call(float DisplayCPU(float))
3      && args(value);
4
5  after(float value) : CPUMonitor(value) {
6      /*
7      Advice, specify adaptation policy here
8      */
9  }
```

Listing 4.1 shows an example of how to carry out adaptations in AspectJ. Line 1-3 is the pointcut session; it defines the interest of the concern. In line 1, the **pointcut** is defined with the name as *CPUMonitor*. In line 2, it indicates the action of calling function *DisplayCPU* will trigger the corresponding reactions, which are indicated in the advice region. In line 3, it indicates that the variable value will be retrieved and passed to the advice session. The segment from line 5 is the advice session. In line 5, it indicates this **advice** implements the reactions corresponding to the **pointcut** *CPUMonitor*. In this example, detailed adaptation policies could be written in this session to guide the adaptations according to the value of CPU, whenever the function of *DisplayCPU* is called. By periodically calling the *DisplayCPU* function, the system could monitor the CPU value and take necessary adaptation – if necessary.

4.1.4 Binding AOP with OSGi

The systems around us, both the digital systems and the service systems, are becoming more and more powerful. Meanwhile, the complexity of the systems also keeps increasing, which may result in a tangled system. The above mentioned problem could be addressed by improving the separation of system concerns and the modularity of the system [Parnas, 1972]. Both AOP and SOA aim to improve the separation of concern and modularity, however, these two approaches seem to be orthogonal to each other: SOA provides means to develop applications in terms of loosely coupled services, thus improving the modularity of the system; while AOP is designed to modularize crosscutting concerns across the whole system.

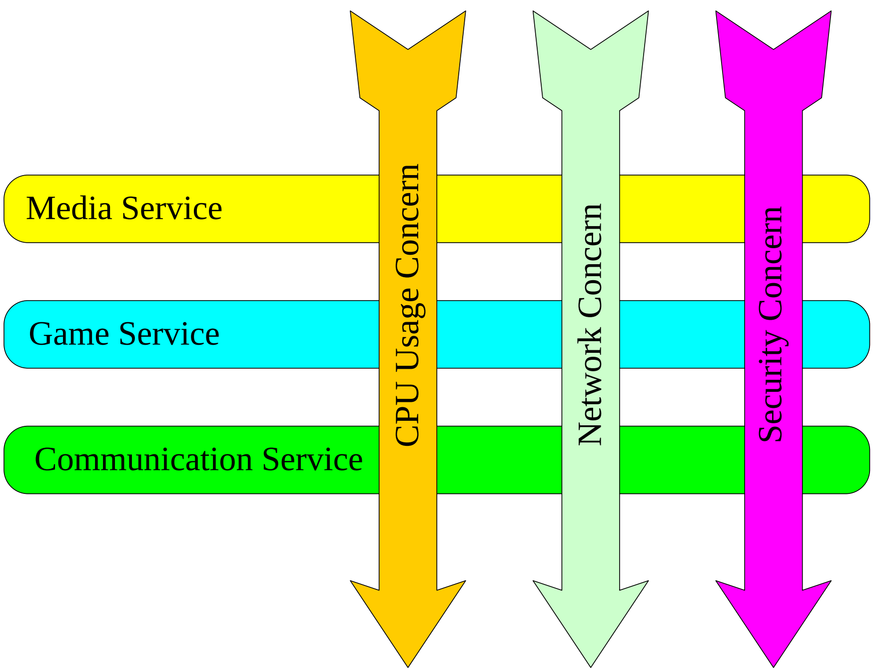


Figure 4.4 Improving Separation of Concerns and Modularity with AOP and SOA

Figure 4.4 shows that SOA helps to separate the different applications into services in the horizontal way, e.g. media service, communication service, .etc.; while AOP helps to separate different system concerns in the vertical way, e.g. CPU usage, Network Condition, etc. The former one modularizes the functional concerns while the latter one separates the non-functional concerns. Their combination could provide a powerful solution to build systems which clearly separate both functional and non-functional concerns thus well addressing the tangling issues.

In the adaptation framework presented in this thesis, the OSGi framework is used to modularize the applications as services, thus the integration with OSGi should be solved before the AOP could be used to guide the adaptation. In Fig 4.3, the classes having the concern X (classes A, B, and C), and the aspect class all reside in the same classpath, however, in the OSGi platform, those classes could be wrapped into different bundles, separated in different bundle-classpaths. In this case, techniques to weave the aspect between different bundles are required.

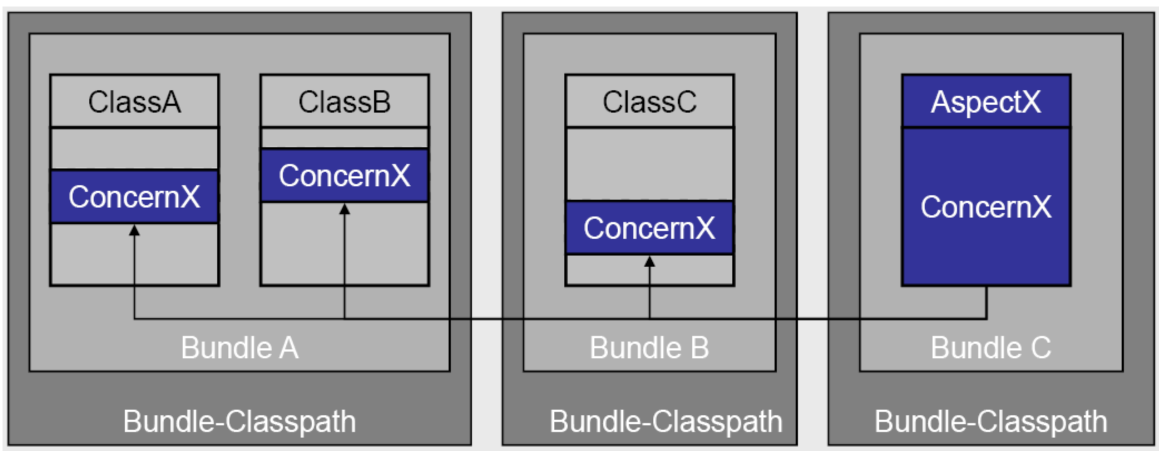


Figure 4.5 Aspect Weaving for OSGi [Lippert, 2008]

There are many research efforts carried out in integrating the AOP with OSGi, such as the AJEER [Lippert, 2004] and AOSGi, which were merged as Equinox Aspect project [Equinox Aspect, 2008]. They integrated the AspectJ into the OSGi platform (Equinox); using a load-time weaving extension it is able to add AspectJ aspects to the bundle-based system just by putting them into general OSGi bundles. Figure 4.5 shows the aspect weaving for OSGi. Applications shown in Fig 4.3 are now encapsulated in different OSGi bundles; the aspects also reside in a separated aspect bundle (Bundle C). Although located in different bundles (Bundle A and B), the concern X could still be woven by the Bundle C. There are also other approaches to realize the integration, such as Jadabs [Frei, 2005], which uses dynamic proxies to implement the aspects into the OSGi platform; and some other effort is carried out in integrating the JBoss AOP into the OSGi platform [Irmert, 2007]. This thesis uses Equinox Aspect in constructing the adaptation framework.

4.2 A Generic Service Adaptation Framework

This section will introduce the construction of a generic service adaptation framework.

4.2.1 Scales of Adaptation Systems

The scales of the adaptation systems evolved from simple devices to complex systems. In Fig 4.6, I have summarized the hierarchy of the adaptation systems, which may also be considered as the trend of the development of adaptation systems, as the following three levels:

1) **Device Level**

The origin focus of adaptations is aimed to improve the performance on single devices. By changing the settings of certain parameters during the runtime, the device could greatly improve its performance. Adaptations on this level normally focus on certain goals or features such as energy saving adaptation for battery [Flinn, 2004], wireless network rate adaptation [Lacage, 2004], etc.

2) **Network Level**

With the wide application of network technology, especially the massive application of wireless communication techniques, devices are more and more connected together as a network and communicate with each other, such as Ad Hoc networks [Wu, 2004]. Applications are always focused on certain domains such as smart home [I-living, 2007] or intelligent office [Hall, 2001], etc. Devices with adaptation abilities are important components of the networked system. Based on the changes, corresponding adaptations could be carried out on the right devices. Adaptation concerns are not only restricted on a single device, but also take care of the overall performance of the connected devices.

3) **Community Level**

The next stop of system evolvement, after the networked level, is the evolution towards the scale of community. Systems on this level are focused on characterizing the behaviours of the related communities; they work as social information infrastructure for our everyday lives. Networked devices are acting in different specific domains, together with different participants, to consist the community as whole. Instances of the communities in this domain could be e.g. digital cities [Ishida, 2002], mutual assistance community (as shown in this thesis) [Sun, 2010], etc. Adaptations on the community level are the interactions between different entities inside the community, especially satisfying the requests from different participants by providing appropriate services in prompt way. Systems on the community level are aiming to improve the welfare of the whole community.

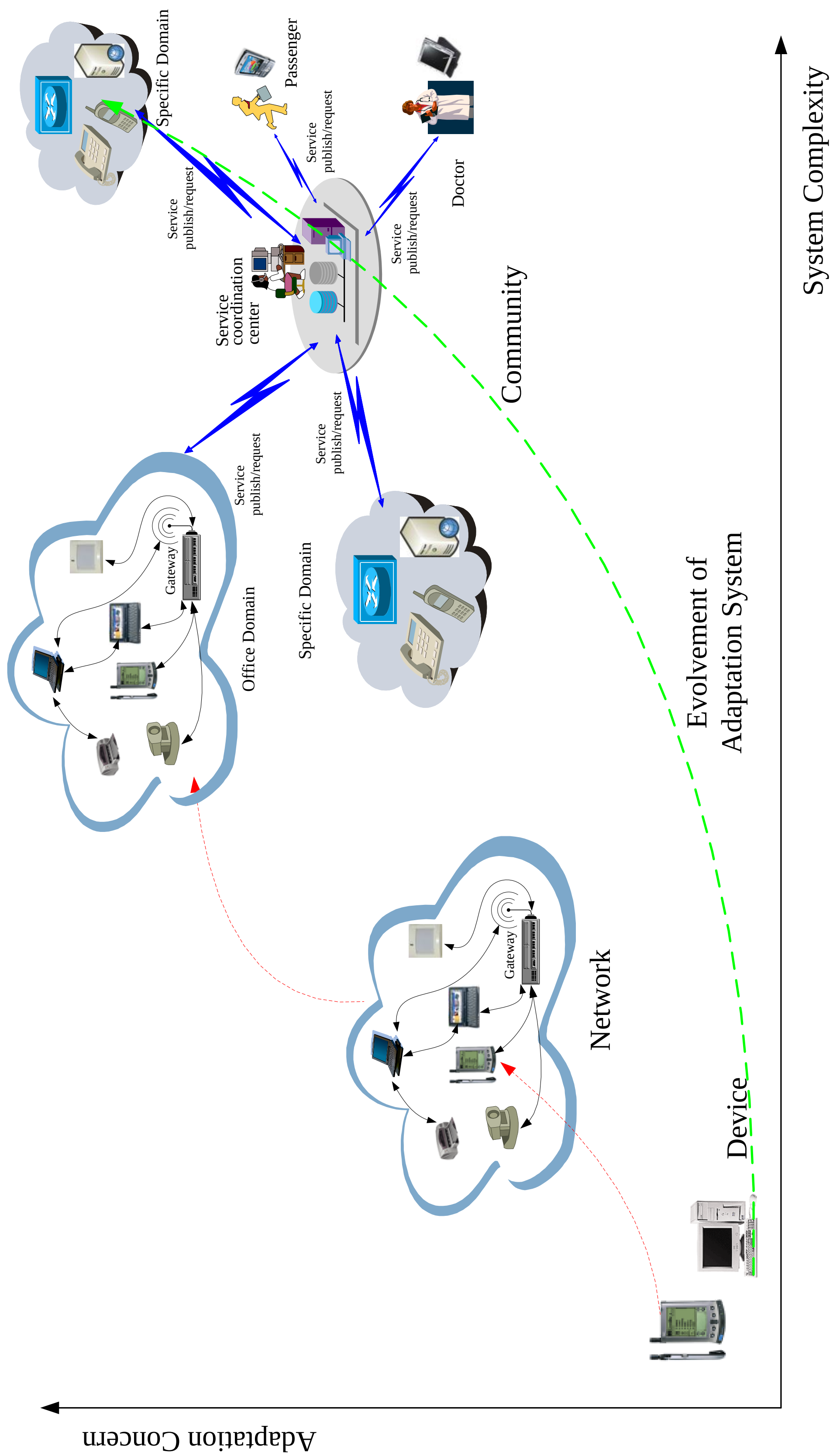


Figure 4.6 The Evolvement of Adaptation Systems

4.2.2 Design Issues

Many design issues need to be considered in developing adaptation framework for mobile systems, among which, we deem the following aspects to be the most important ones:

When to take the adaptations:

Choices here include deploying the adaptation policies in the installation phase, or at execution time [Houssos, 2003]. Both of these two solutions have their advantages and disadvantages.

If the adaptation schemes are solely deployed in the setup time, they would lack flexibility, while if there are too many operations in the runtime, this will also introduce extra costs such as extra time to switch the policies, and make the system too complex to design. In the design of the adaptation framework in this thesis, I set the adaptation policies in the deployment time, however, still providing the users with access to switch adaptation policies and modify adaptation parameters during the runtime. Such runtime modifications can be implemented in a simple way and without introducing an excessive amount of design complexity.

Where to carry out the adaptation:

Three models are widely used to take adaptations: centralized, application-transparent adaptation; decentralized, application-specific adaptation; and an integrated model [Edmonds, 2001]. The centralized model performs adaptation at operating system level, which avoids the competition between applications, makes decisions concerning the best adaptation strategy on system level and aids efficiency in resource usage. However, the drawback is that lacking the knowledge from specific applications, adaptation actions are not efficient in such a model. The decentralized model performs adaptation solely within the application. Monitoring and adaptations are glued together in certain applications without any support from the system. Such a solution is effective to solve a particular adaptation facet, however, it is scarcely efficient system wide, and there might be conflicts between different decentralized adaptations. The integrated adaptation model takes the advantage of the two above mentioned approaches: it combines the system management of resources with the application specific knowledge of exactly what is required and how to execute the adaptations. The integrated approach is much more difficult to implement compared with the other two approaches: it requires especially tight cooperation between the applications and the operating system, and existing applications need to be modified to run effectively under such model. This thesis uses the integrated approach to design the adaptation framework. Monitoring and applications are scattered in the system, however, the monitoring results, adaptation analysis and adaptation instructions are processed in a centralized way.

How to carry out the adaptation:

There are many ways to take adaptations; however, here I generally categorize them as three levels based on granularity: the highest level is switching adaptation policies according to different locations, targets, etc.; the second one is adaptation of services, e.g. switching the coding method of a video encoder, etc; the last one is setting key parameters, e.g. resetting the frame dropping rate of a mobile video player, etc. In my previous research [Sun, 2007], I have investigated on building an adaptation framework combining adaptations on both services and parameters. However, the drawback here is that these two adaptations are carried out separately through SOA and Reflective and Refractive variables [De Florio, 2007] – a low-level application-layer reflective structure. In this chapter, I will combine the Aspect Oriented Programming and SOA to integrate the above mentioned approaches together.

4.2.3 Global Adaptation Framework

In my previous work [Sun, 2007], I have proposed to use a so-called global adaptation framework to realize the adaptation. In order to make the correct adaptation in the mobile application, there is a stringent requirement for the system to be aware of the environmental changes. The proposed adaptation strategies are thus constructed as an event-condition-action model [Etter, 2006] [De Florio, 2007], which is similar to what proposed in autonomic computing [Kephart, 2003]. The system is context aware, and adaptations are taken when the context changes. The event module detects the changes of the surrounding environment. The detected events trigger the reasoning of the rule engine; depending on the conditions, specific adaptation rules will be fired and the adaptations will be held in the action module.

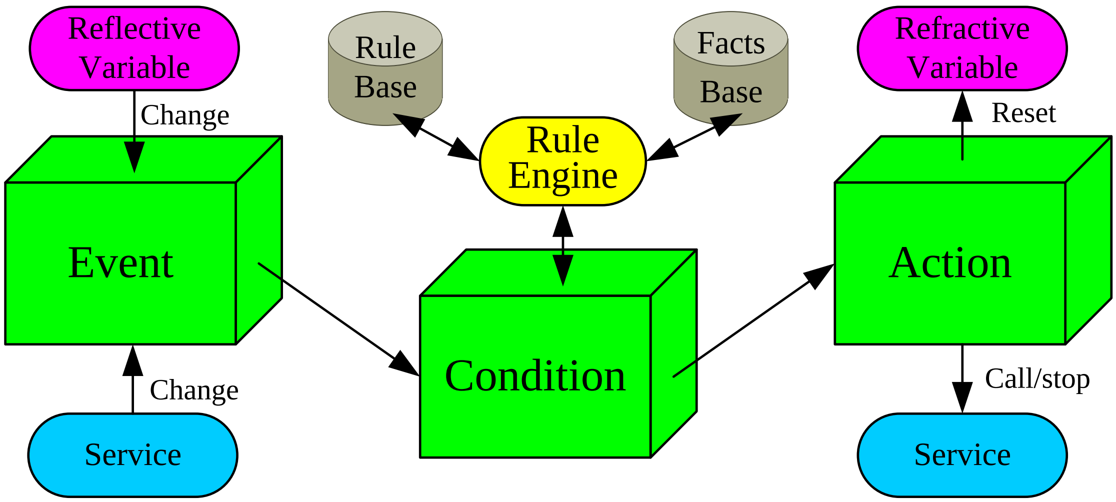


Figure 4.7 Architecture of Global Adaptation Framework [Sun, 2007]

In Fig 4.7, the event module is designed to monitor the environment around the mobile terminal, availability of applications and the system status of the mobile terminal. Services and reflective variables are used to separate different kinds of information. Services are developed as OSGi bundles, and changes of the services are detected by the service listener. In order to better indicate the system status, so-called "reflective variables" [De Florio, 2007] are used to represent some of the key system state parameters. Adaptation decisions are made in the condition module, where certain rule engines can be implemented, and adaptation decisions are to be deduced based on predefined rules. The action module in the framework executes the adaptations when certain adaptation rules are fired in the condition module. Adaptations in the action module are executed as adjustments on service, such as start/stop/update OSGi bundles; or by setting some system parameters, which is done through so-called "refractive variables" [De Florio, 2007].

The above mentioned solution provides an agile way to detect the change of context by the combination of service change detection and the reflective/refractive variables. It also provides fine grained adaptation by the combination of changing services and resetting the refractive variables. The drawback is that firstly, the monitoring and adaptations of services and variables are processed separately by SOA and Reflective and Refractive variables, which increases system complexity and reduces efficiency; and secondly, the adaptation policies are all stored as predefined rules in deployment, which lacks of flexibility to carry out runtime changes.

In the next section, I will introduce a generic context aware adaptation framework, which addresses the above mentioned drawbacks. Experiments based on such a framework will also be given.

4.2.4 Generic Context Aware Global Adaptation Framework

The last section discussed the architecture of an adaptation framework, the purpose is to manipulate monitoring and adaptations in a different granularity, however, separating the monitoring and adaptations in different approaches, i.e. services and key variables, also increased the system complexity. This section will demonstrate how to process the changes on different levels homogeneously in a generic adaptation framework.

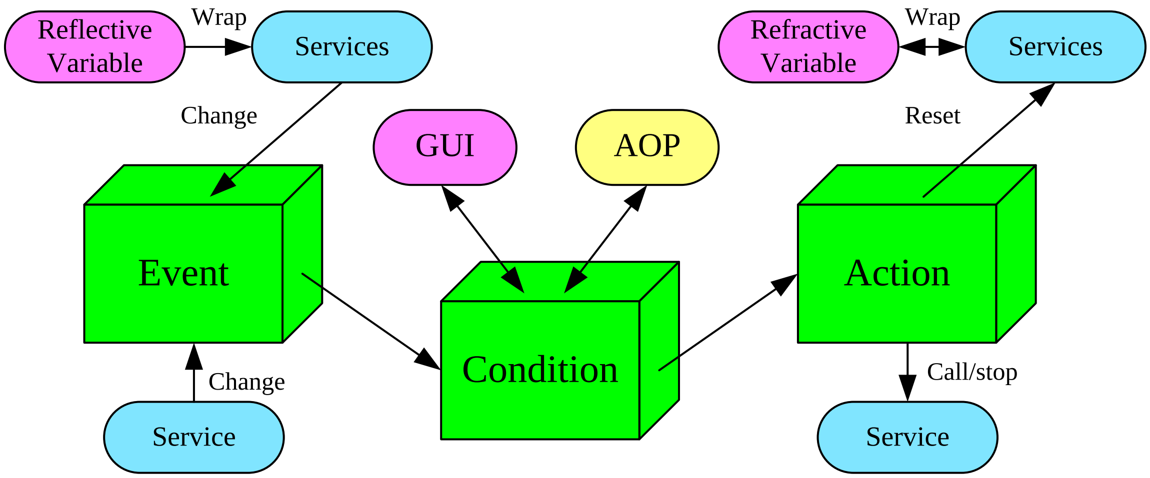


Figure 4.8 Unifying the Adaptation Models

Figure 4.8 shows that the changes of variables or services could be universally addressed by wrapping the variables as services. In Fig 4.7, the events of monitoring and acting are based on either services or some important variables; while in the adaptation framework shown in Fig 4.8, all the actions are carried out as services. Those changes carried out as services in Fig 4.7 remain, while those changes carried out as parameters are wrapped in read/write functions, and bundled in services. The actions of those read/write functions could be manipulated by adaptation policies specified with AOP, and adaptations are thus carried out.

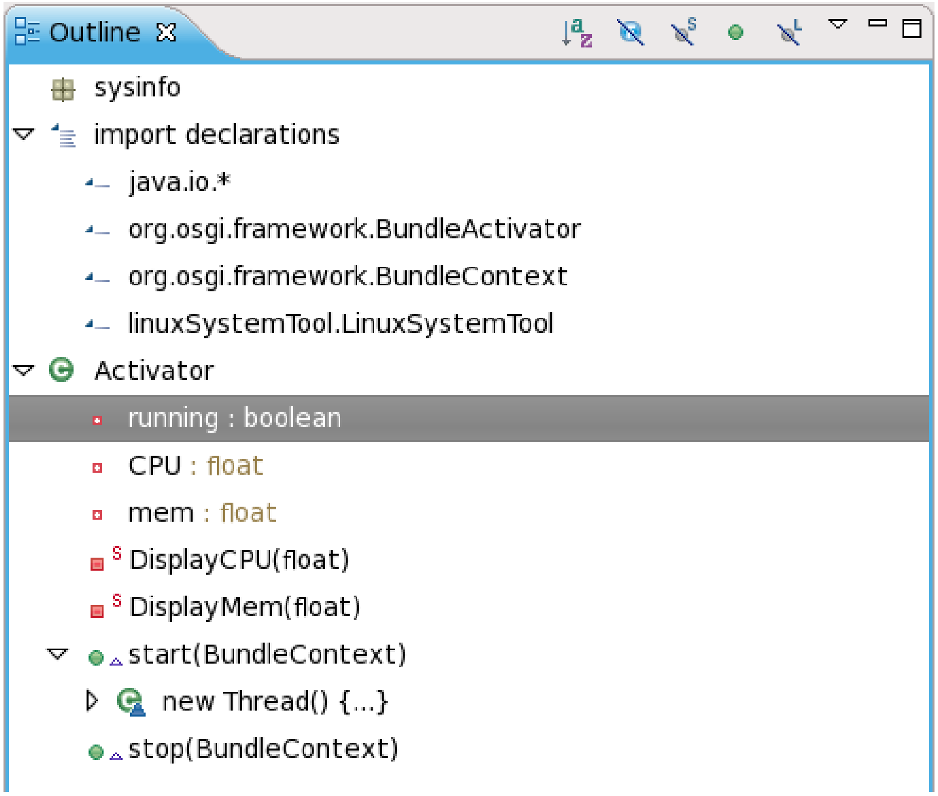


Figure 4.9 Building Resource Monitor in Bundle

Figure 4.9 shows the outline of the bundle *sysinfo*, which monitors the value of CPU usage and Memory usage. Functions to retrieve these two parameters are implemented in the *linuxSystemTool* package, and this package is imported in the *sysinfo* package. The value of CPU and Memory usages are retrieved in the activator body by calling the functions in the *linuxSystemTool* package. The functions *DisplayCPU()* and *DisplayMem()* return respectively the values of CPU and Memory usage. These two functions would be monitored by an aspect bundle once they are called, the return value (which refers to CPU usage and memory usage respectively) would be retrieved by the aspect bundle and corresponding reactions could be attached. A thread is created in the *sysinfo* bundle, which periodically calls the *DisplayCPU()* and *DisplayMem()* functions. By doing so, the adaptation framework periodically retrieves the values reflecting the system status, and it could take the reactions correspondingly.

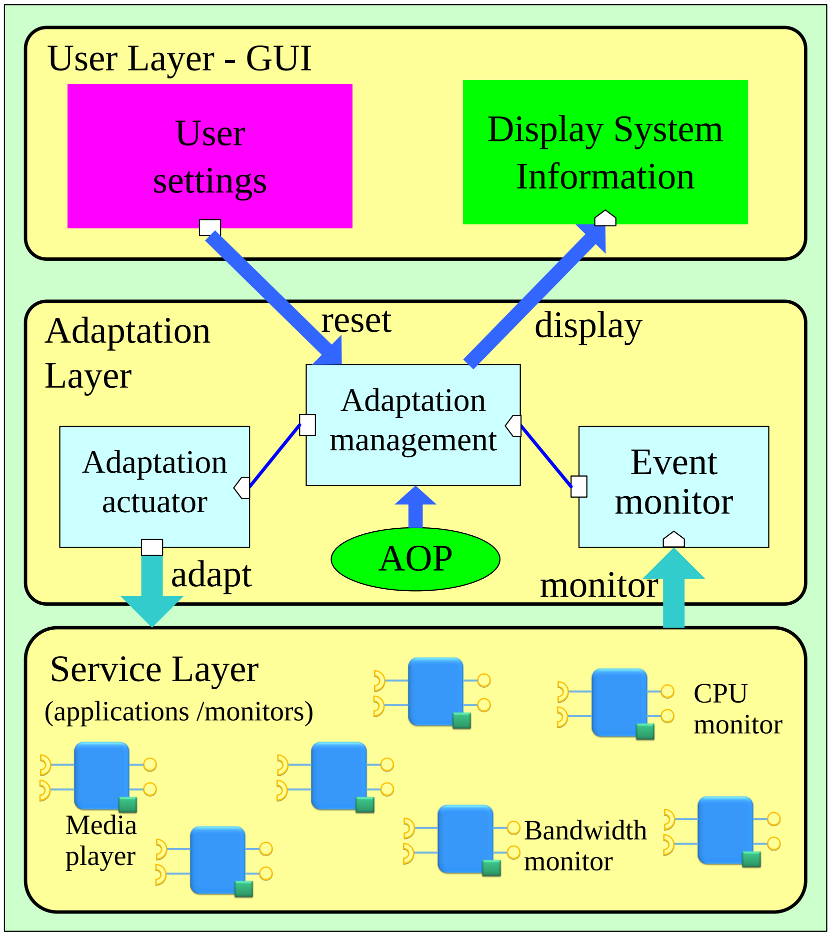


Figure 4.10 Generic Context Aware Global Adaptation Framework

Figure 4.10 shows the generic context aware adaptation framework, which is evolved from the adaptation framework shown in Fig 4.8. The framework in Fig 4.8 [Sun, 2009] is divided into three layers. In the service layer, applications are wrapped as services to provide approaches to manage the system in a modularized way. Applications could be of various categories, ranging from applications such as media players, or different types of monitors (e.g. CPU monitor, bandwidth monitor, etc.).

In the adaptation layer, the event monitor module collects the detected changes from different monitor bundles. Information of the interested resources are collected system-widely: for instance, the value of CPU usage and bandwidth usage are reported by the CPU monitor and bandwidth monitor separately; however, contrarily to the application-specific approach, these monitored values will be collected together by the event monitor module to analyze and trigger the adaptations.

The relationship between the monitor modules and adaptation modules are generally expressed as aspect weavings in Equinox Aspect [Equinox Aspect], which is shown in

Fig 4.5 [Lippert, 2008]. Concerns in different bundles, e.g. CPU usage, are monitored by a separate aspect bundle. Adaptation policies are also resident in the same aspect bundle written with AspectJ language in the advice area. An independent GUI is also attached to this adaptation bundle.

Adaptation decisions are deduced in the adaptation management module, based on the adaptation policies and the system resources. The relationship between monitoring and adaptation in AOP is generally expressed as shown in List 5.1. In real-life applications, the expression of adaptation logics could be very complex, and such adaptations will be executed by the adaptation actuator module. Those actions are actually carried out on the applications in the service layer, such as the applications of media-player, etc.

In order to address the user's quality of experience (QoE), a user layer is added to the adaptation framework, which hosts a graphical user interface. Such GUI is built in a separated *QoEGUI* bundle, and is connected to the adaptation management module in the adaptation layer.

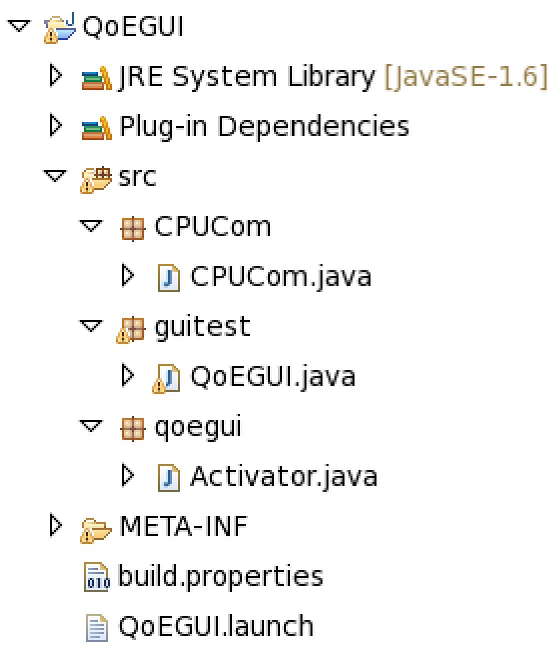


Figure 4.11 Structure of GUI Bundle

Figure 4.11 shows the structure of the *QoEGUI* bundle. Three packages are contained in this bundle. The *guitest* package implements the graphic interface; the *CPUCom* package contains function to artificially increase the CPU usage, for testing purpose; the *qoegui* package contains the activator for the bundle.

```java
package qoegui;

import guitest.QoEGUI;

public class Activator implements BundleActivator {

     * @Hong Sun

    public QoEGUI GUIObj(QoEGUI GUI)throws IOException{

        GUI.init();
        return GUI;
    }

    public void start(BundleContext context) throws Exception {
    System.out.println("QoE Graphical User Interface is starting now");
        QoEGUI GUI1=new QoEGUI();

        GUIObj(GUI1);
    }
    public void stop(BundleContext context) throws Exception {

    }

}
```

Figure 4.12 Activator of the GUI Bundle

Figure 4.12 shows codes of the activator for the *QoEGUI* bundle. The activator implements a function *GUIObj()*, which is called when the bundle is started. The function initializes the GUI interface, and returns the GUI object. Such an object is then retrieved by the aspect bundle, and the changes in the GUI is then received in the aspect bundle and guide the adaptations in the runtime.

```java
pointcut GUIMonitor(QoEGUI GUI):
    call(public QoEGUI GUIObj(QoEGUI))
    && args(GUI);


after(QoEGUI GUI):GUIMonitor(GUI){
    GUIObject=GUI;
    GUIVar=true;
    System.out.println("Graphic User Interface is now starting");
}
```

Figure 4.13 Retrieving GUI Object in Aspect Bundle

Figure 4.13 shows the code of retrieving the GUI object in the aspect bundle. By manipulating the GUI object, the system information received by the monitor module is then displayed on the GUI. Most importantly, the GUI provides the users with an access to switching between the predefined adaptation policies, or resetting some key parameters, such as applications priorities, during the runtime. An instance of the GUI will be shown in the following experiments sections.

4.3 Experiments

The following application was developed as an experiment to validate how the proposed adaptation framework could monitor the system information and guide the adaptations on video playing.

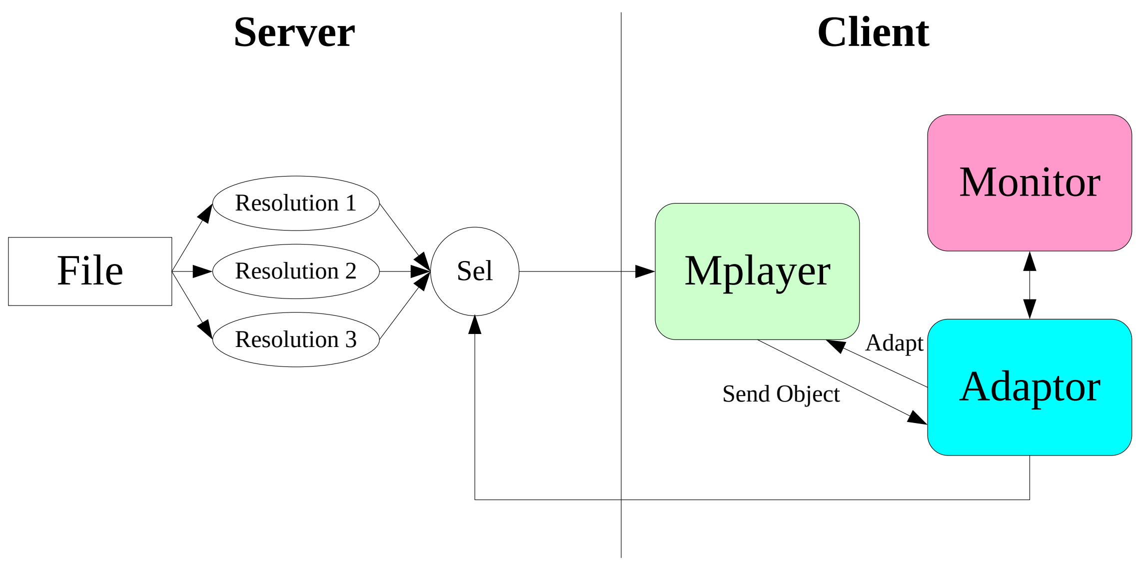


Figure 4.14 Experiment Structure

The structure of the experiment is shown in Fig 4.14. A videoclip is played by the Mplayer (an open source media player) in the client side as an application. The sources of the displayed videos are stored in the server side, and the server provides the videos with three different resolutions upon decisions from the client side. The monitor module in the client side monitors the information of CPU usage, bandwidth, and memory exchange rate; such information is displayed in the GUI in Fig 4.15. The adaptor bundle retrieves the system information from the monitor bundle; meanwhile, it also retrieves MPlayer objects, which describe and control the video that is being played by the MPlayer. According to the system status, the adaptor bundle takes corresponding adaptations on the Mplayer objects, and selects the appropriate resolutions. When the system is meeting restrictions on resources, such as low CPU, or limited bandwidth, it is still possible to avoid performance failures by trading off quality of service that least affects QoE. In this case performance failures are avoided by reducing the resolution of the videos.

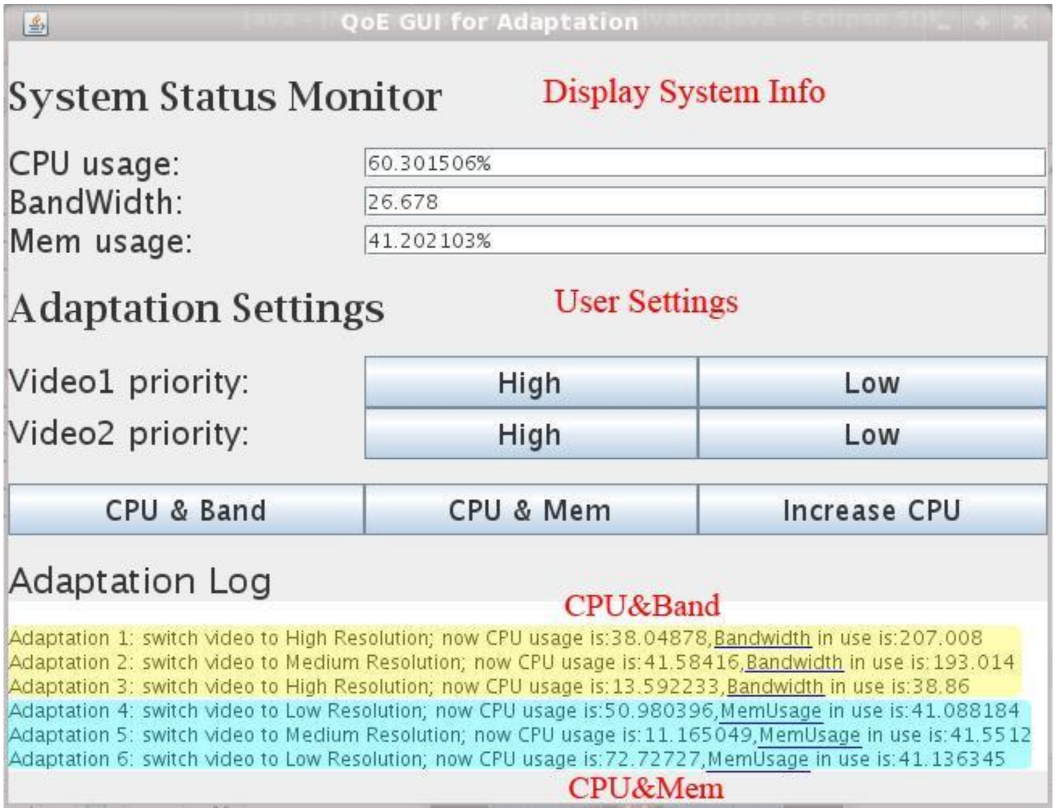


Figure 4.15 Graphic User Interface

The thresholds defined here to take the adaptations are shown in Fig 4.16. There is no particular reason to choose these numbers, which are to be considered here merely a proof of concept for our adaptation framework. There are two adaptation policies in the experiment: either considering the combined value of CPU and Bandwidth usage, or considering the combined value of CPU and Memory usage. Users have to choose one of these policies; it is also possible for user to change the adaptation policies during the runtime through the GUI (see the User settings buttons in Fig 4.15.). In the adaptation log in Fig 4.15, it is shown that the adaptations are initialy triggered by the change of the values on CPU and bandwidth, while later on we switched to adaptations based on the combination of CPU and memory usage. It is also possible to switch the priorities between different videos when more than one video is being displayed.

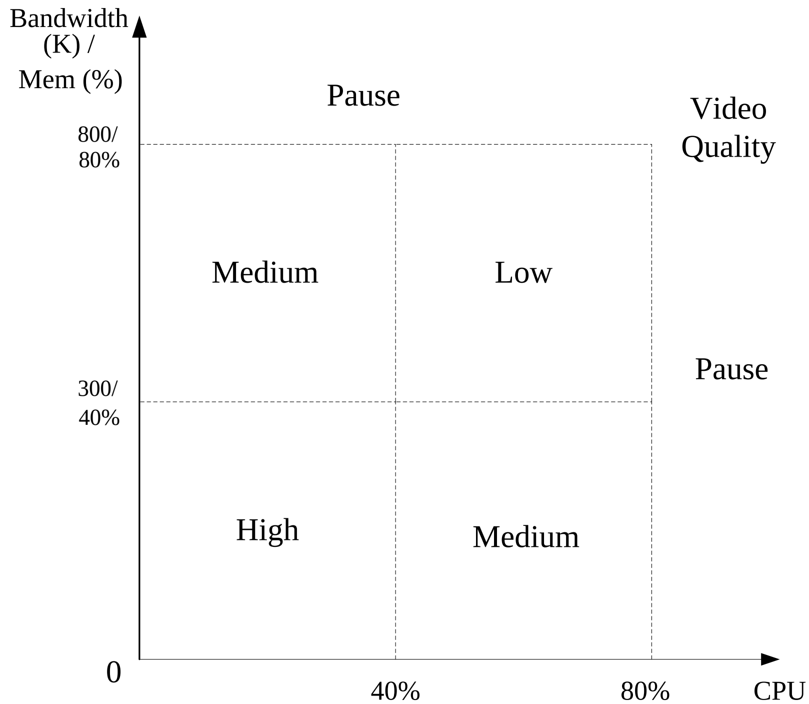


Figure 4.16 Adaptation Thresholds

The results of the experiments proved that our proposed adaptation framework could flexibly organize the adaptations, while also being able to meet the user's requirements, and even change adaptation policies at runtime. The above experiment was presented as a demo at the final project event of IBBT project End-to-end Quality of Experience [QoE, 2006]. In our demonstration, the video could be broadcast with minimal impact on quality of experience by reducing resolutions in stringent computation environments.

The demo here is worked out as a proof of concept to show that the adaptation framework is able to switch adaptation policies at runtime and it is also capable to orchestrate adaptations on different applications. Thus the adaptation policies that we listed above only works as a simple scenario – finding efficient adaptation policies is out of the research scope of this thesis.

4.4 Managing the Assistive Devices

Assistive devices are important in building smart environments. Those devices are providing valuable information about the health status of the elderly people (e.g. ECG (Electrocardiogram), glucose monitors), as well as the situations of the surrounding environment (e.g. motion detectors, gas leakage detectors). However, mostly, such information is provided by separated assistive devices in their own ways. Data provided by the assistive devices are hard to process for further analysis, and integrating information from different assistive devices also appears to be difficult.

Advanced, smart assistive devices are developed recently to provide easier expression of information and allow better post processing of the information [Braecklein, 2005]. For instance, Avetana is a Personal Health Record system which provides Health Telematic Services [Avetana, 2010]. The system supports the automatic uploading of relevant vital signals (weight, blood pressure, glucose level and ECG) from medical devices via wireless channels. In the application, a chest strap is worn by high risk patients, which permanently monitors the ECG signal and analyzes the signal with an intelligent ECG sensor. If the analysis identifies an acute, life-threatening event or other heart problems, the system will trigger an alarm and ECG data around the event will be sent to the data centre. In addition, information about a patient’s weight, blood pressure etc. is stored in the data centre as an Electronic Health Record (as shown in Fig. 4.17) which is able to reflect the patient’s health status in the long term.

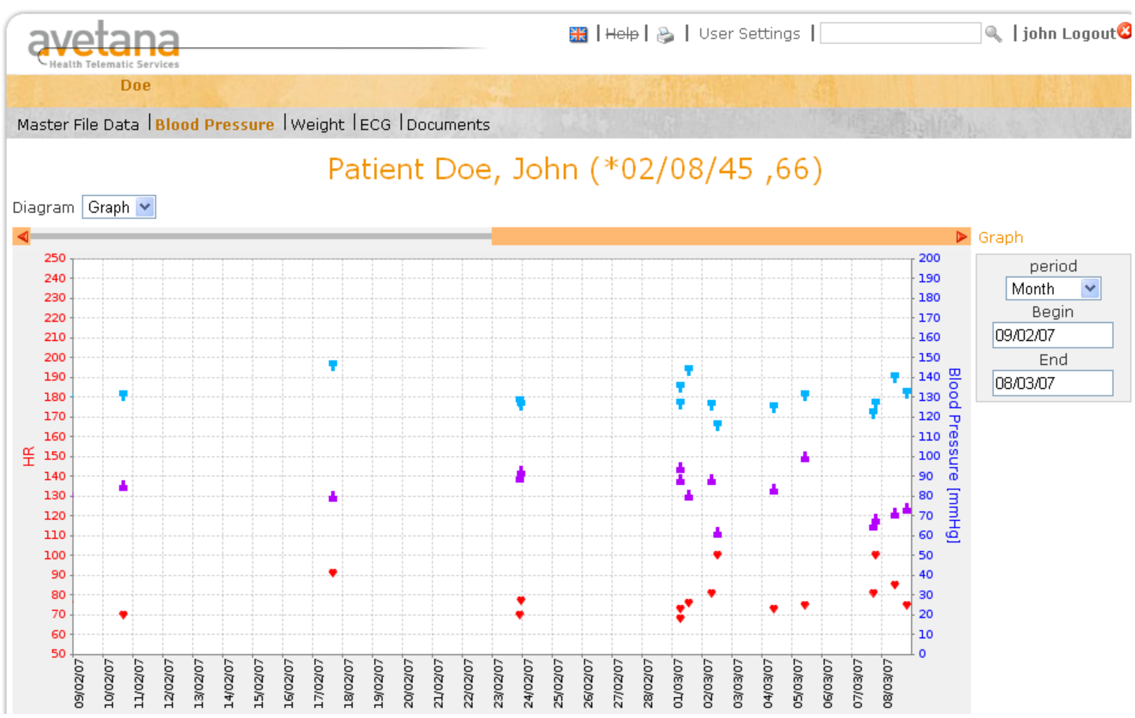


Figure 4.17. Sample Electronic Health Record of a Patient’s health status

Applications such as Avetana greatly improved the usability of the data from the assistive devices. However, in constructing ambient assisted living environments in the real world, there are many assistive devices to be installed, which are provided by various manufactures. Integrating data from these different assistive devices could best explore the potential of these devices to construct a safe and intelligent environment. However, as these devices are using different techniques and protocols to detect, process and send data, communication between these assistive devices is difficult and exchanging data among them is a big challenge.

The service adaptation system based on SOA could be a good candidate to meet the above mentioned challenge. By wrapping the assistive devices into service components, those devices could be loosely coupled together, and they may easily communicate with the system and exchange their data.

Nevertheless, wrapping various assistive devices into services in the SOA architecture is no easy task. Several standards exist in processing sensor data in a formal way, e.g. Sensor Model Language [SensorML, 2010], IEEE 1451. There are also research efforts with this specific target of developing different sensor devices into SOA architecture: An example is SODA (Service Oriented Device Architecture), an Alliance carried out between IBM and University of Florida. SODA [Chen, 2009] [SODA, 2010] aims to let programmers deal with devices such as sensors and actuators as services in Service Oriented Architectures. With its own Device Description Language, SODA converts hardware devices to software services with well defined interfaces. Such process is independent of the programming language and the computing platforms to which they are connected.

After wrapping the assistive devices into SOA services, the context aware adaptation framework introduced earlier in this chapter could be extended to integrate the assistive devices together to set up smart environments around the elderly people. Figure 4.18 demonstrates how to build context aware environments to link assistive devices. In the Assistive Device layer, different assistive devices, such as ECG, motion detectors, etc., are wrapped as services. In the Local Coordinator layer, data from different assistive devices are monitored. In case abnormal situations are detected, which could be a combination of data from more than one device, corresponding reactions would be triggered. Reactions could be either triggering the actuators in the assistive devices layer (e.g. turn off the heater when the temperature is high) or sending service requests to the Service Coordination layer (e.g. looking for emergence service when the ECG signal is abnormal). With such a framework, different assistive devices could be integrated within one ambient assisted living system. Data from different devices could be collected and the adaptation policy could use data from all the devices. By combing data from different devices and analyzing with adaptation rules, the system is able to detect and analyze and deduce more situations which were impossible with single assistive device. The system also allows more reactions when events are detected.

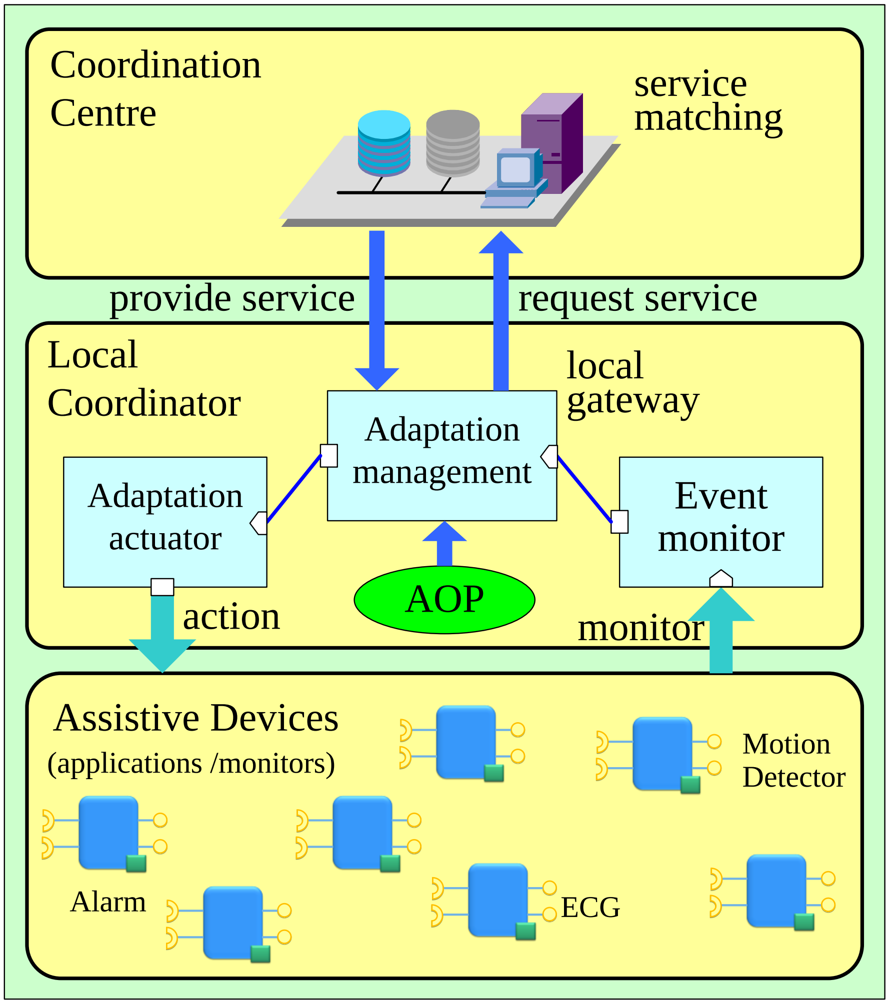


Figure 4.18 Managing Assistive devices

4.5 Conclusions

This chapter discusses the issues of carrying out service adaptations on mobile devices. The complexity of mobile devices and the mobile network ever increases, which makes it difficult to manage such systems, and carry out appropriate adaptations to improve their performance. Modularization could be considered as an effective approach to cope with the increasing complexity of the mobile devices. This chapter discusses two approaches for modularization: the OSGi and AOP. Though they seem to be orthogonal to each other, the combination of these two techniques could provide a full fledged solution to build up a modularized and adaptive system.

A generic context aware adaptation framework is presented in this chapter, which combines OSGi and AOP and allows to process adaptations on different granularities in a generic way. The adaptation framework presented in this chapter also includes a GUI to allow the users to change their preferences during the runtime.

Experiments based on such an adaptation framework are also presented. A demo is built as a proof of concept. More complex applications could be built up upon the above mentioned adaptation framework. In such situations, more bundles would be involved, the adaptation policies would be more complex and rule bases might be required to switch between different adaptation policies.

The integration of different assistive devices in ambient assisted living is also discussed. Through wrapping the assistive devices as services in the SOA architecture, the assistive devices could be managed with context aware adaptation frameworks. Data from different assistive devices could be aggregated together for decision making, which helps to better utilize the assistive devices to build up smarter ambient assisted living environments.

References

Aventana (2010), Retrieved on December, 2010 from http://www.avetana.de/

Braecklein, M., Tchoudovski, I., Moor, C., Egorouchkina, K., Pang, L.P. and Bolz, A (2005). Wireless Telecardiological Monitoring System for the Homecare Area, in the proceedings of the 27th Annual Conference of IEEE Engineering in Medicine and Biology.

Chen, C. and Helal, A. (2009). Device Integration in SODA Using the Device Description Language in proceedings of 9th Annual International Symposium on Applications and the Internet.

Costa, P. , et al (2007). The RUNES middleware for networked embedded systems and its application in a disaster management scenario, 5th Annual IEEE Int. Conf. on Pervasive Computing and Communication.

Cristian, F. (1991). Understanding Fault-Tolerant Distributed Systems, Communications of the ACM vol.34 no.2, pp.56-78.

De Florio, V. and Blondia, C. (2007), Reflective and Refractive Variables: A Model for Effective and Maintainable Adaptive-and-Dependable Software, in the Proceedings of the 33rd Euromicro Conference on Software Engineering and Advanced Applications (SEEA 2007), Luebeck, Germany.

De Florio, V. and Blondia, C. (2005), A System Structure for Adaptive Mobile Applications, In Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2005 (WOWMOM 2005), Giardini Naxos, Italy.

Edmonds, T., Hopper, A. and Hodges, S. (2001), Pervasive Adaptation for Mobile Computing, in proceedings of 15th International Conference on Information Networking (ICOIN'01),.

Eikerling, H. and Berger, F. (2002), Design of OSGi Compatible Middleware Components for Mobile Multimedia Applications. In Protocols and Systems for Interactive Distributed Multimedia Systems.

Elrad, T., Filman, R., and Bader, A. (2001), Aspect-Oriented Programming, Communications of the ACM, Volume 44, No.10.

Equinox Aspects Project (2008). Retrieved on April, 2010 from http://www.eclipse.org/equinox/incubator/aspects

Etter, E., Costa, P. and Broens, T. (2006), A Rule-Based Approach Towards Context-Aware User Notification Services. In: Proceedings of the IEEE International Conference on Pervasive Services 2006, Lyon, France. pp. 281-284.

Flinn, J., Satyanarayanan, M (2004). Managing battery lifetime with energy-aware adaptation, ACM Transactions on Computer Systems (TOCS), Volume 22 , Issue 2.

Frei, A., Alonso, G. (2005), A Dynamic Lightweight Platform for Ad-Hoc Infrastructures. In: PERCOM ’05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications, Washington, DC, USA, IEEE Computer Society.

Gu, T., Pung, H.K. and Zhang, D.Q. (2004), Toward an OSGi-based infrastructure for context-aware applications, IEEE Pervasive Computing., vol. 3, no. 4, pp. 66–74.

Hall, D., Gal, C., Martin, J., Chomat, O., Crowley, J.(2001), MagicBoard: A contribution to an intelligent office environment, Robotics and Autonomous Systems, Volume 35, Issues 3-4.

Hall, R. and Cervantes, H. (2004), Challenges in building service-oriented applications for OSGi. IEEE Communications Magazine, 42(5):144–149.

Houssos, N., Gazis, V. and Alonistioti, N. (2003), A Novel Mechanism for Mobile Service Adaptation, In 57th IEEE Vehicular Technology Conference (VTC 2003 Fall).

I-living (2007). I Assisted Living Project, Retrieved on April, 2010 from http://lion.cs.uiuc.edu/assistedliving/

Irmert, F., Meyerhofer, M., and Weiten, M. (2007), Towards Runtime Adaptation in a SOA Environment. RAM-SE'07 - 4th ECOOP Workshop on Reflection, AOP and Meta-Data for Software Evolution, Berlin, Germany.

Kephart, J., Chess, D. (2003), The Vision of Autonomic Computing, Computer, Volume 36, Issue 1.

Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Videira Lopes, C., Loingtier, J.-M., and Irwin, J. (1997), Aspect-Oriented Programming, in the proceedings of the European Conference on Object-Oriented Programming (ECOOP), Finland. Springer-Verlag LNCS 1241.

Kiczales, G., Hilsdale, E., Hugunin, J., Kersten, M., Palm, J. and Griswold, W. (2001), An overview of AspectJ. Proceedings of the European Conference on Object-Oriented Programming (ECOOP'01) (Lecture Notes in Computer Science, vol. 2072). Springer: Berlin.

Kiselev, I (2002). Aspect-Oriented Programming with AspectJ. Sams, Indianapolis, IN, USA.

Kon, F., Costa, F., Blair, G. and Campbell, R. (2002), The case for reflective middleware: building middleware that is flexible, reconfigurable, and yet simple to use. Communications of the ACM (CACM), Volume 45, Issue 6, 33-38.

Lacage, M., Manshaei, M., Turletti, T. (2004), IEEE 802.11 rate adaptation: a practical approach, in Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems, Venice, Italy.

Lippert, M. (2004), AJEER: An AspectJ-Enabled Eclipse Runtime. In: OOPSLA '04: Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications, New York, NY, USA, ACM Press.

Lippert, M. (2008), Aspect weaving for OSGi, in proceedings of Conference on Object Oriented Programming Systems Languages and Applications, Companion to the 23rd ACM SIGPLAN conference on Object-oriented programming systems languages and applications.

OSGi (2007). OSGi Service Platform Release 4, Retrieved on April, 2010 from http://www.osgi.org/Release4/

Parnas, D. (1972), On the criteria to be used in decomposing systems into modules, Communications of the ACM, Volume 15.

QoE (2006), End-to-End Quality of Experience Project, IBBT, Retrieved on December, 2010 from https://projects.ibbt.be/qoe/

Sensor Model Language (SensorML 2010), Retrieved on December, 2010 from http://www.opengeospatial.org/standards/sensorml

SODA (2010), Service Oriented Device Architecture, Retrieved on December, 2010 from http://www.sensorplatform.org/soda/

Sun, H., De Florio,V., Gui, N. and Blondia, C. (2007), Global Adaptation Framework for Quality of Experience of Mobile Services, in proceedings of the 2007 IEEE Three-Rivers Workshop on Soft Computing in Industrial Applications.

Sun, H., De Florio,V., Gui, N. and Blondia, C. (2009), Adaptation Strategies for Performance Failure Avoidance, in proceedings of the 3rd IEEE International Conference on Secure Software Integration and Reliability Improvement (SSIRI 2009), IEEE.

Sun, H., De Florio,V., Gui, N. and Blondia, C. (2010), Building Mutual Assistance Living Community for Elderly People, in Jeffrey Soar, Ed., Intelligent Technologies for the Aged: The Grey Digital Divide, IGI Global.

Sylvain, S., Fabienne, B., and Noel De, P. (2008), Using components for architecture-based management: the self-repair case, in Proceedings of the 30th international conference on Software engineering, Germany.

Wu, J., Stojmenovic, I. (2004), Guest Editors' Introduction: Ad Hoc Networks, Computer, vol. 37, no. 2, pp. 29-31.

Zhang, D., Wang, X., et al. (2004), OSGi Based Service Infrastructure for Context Aware Automotive Telematics. IEEE Vehicular Technology Conference, Milan, Italy.

Chapter 5. Service Adaptation in Healthcare Community

In Chapter 4, I presented a general view of the adaptation system, and showed how the adaptation concerns are evolving from the performance of a single device to the communications of the networked devices and finally moving towards the social scale level e.g. the welfare of a community. The last chapter introduced the service adaptations for connected mobile devices. In this chapter, I introduce the service adaptation in the community level, specifically, for the healthcare community. The mutual assistance community introduced in Chapter 3 is used as the target community to carry out the service orchestration. Service orchestration discussed in this chapter mainly focuses on organizing human services. A service discovery architecture for such a community and the semantic web service e matching techniques are discussed. Experiments together with a scenario are also given to show that the service activities discussed in Chapter 3, e.g. group activities as “participant model”, could be carried out in a real-life application.

5.1 The Evolvement of Healthcare System

As well known, the proportion of elderly people keeps increasing since the end of last century. Much effort is made which aims to build up efficient and effective healthcare systems to assist the elderly people living independently. As discussed in the previous chapters, an adaptation system for autonomous computing could be simplified as a system which monitors the events, analyzes the condition, and takes the corresponding reactions. In this view, a healthcare system could also be considered as an application of the adaptation system, as the function of a healthcare system is to detect changes or requests (i.e. dangerous situations, etc.), and take corresponding reactions to meet the requirements of the users.

Figure 5.1 shows the evolvement of the healthcare system, which quite follows the track of the evolvement of adaptation systems. Similar with the three stages of the adaptation systems introduced in chapter 3, the development of the healthcare system could be summarized as three levels/stages:

1) The first stage is assistive device level: There is much work carried in developing assistive devices to assist the people in need on certain target, e.g. glucose monitoring, reminder services, etc. Services in this level are generally quite simple, reside in separated devices and are provided by different assistive devices separately.

2) The second stage is the smart network level, where much effort is made in combining the assistive devices together to set up a smart environment. Services provided at this level are e.g. localization services, accident detection, etc.; they are provided by the assistive devices and mostly based on the communications between these assistive devices. Healthcare system at this level is more sophisticated than the previous one, and it is possible to provide complex services.

3) The third stage is the community level, where different forces scattered around our society are integrated together to provide services to the people in need. The healthcare system in this level is much more complex compared with the previous two, and the services in the community are provided from different bodies, ranging from devices to human beings. The healthcare system at community level is still at its infancy stage, current research is mostly on connecting the assisted people with limited social connections, e.g. relatives, neighbours [Amigo, 2007] [Aware Home, 2008] [COPLINTHO, 2008]. The mutual assistance community presented in Chapter 3 is a blueprint in extending the healthcare system to the community level.

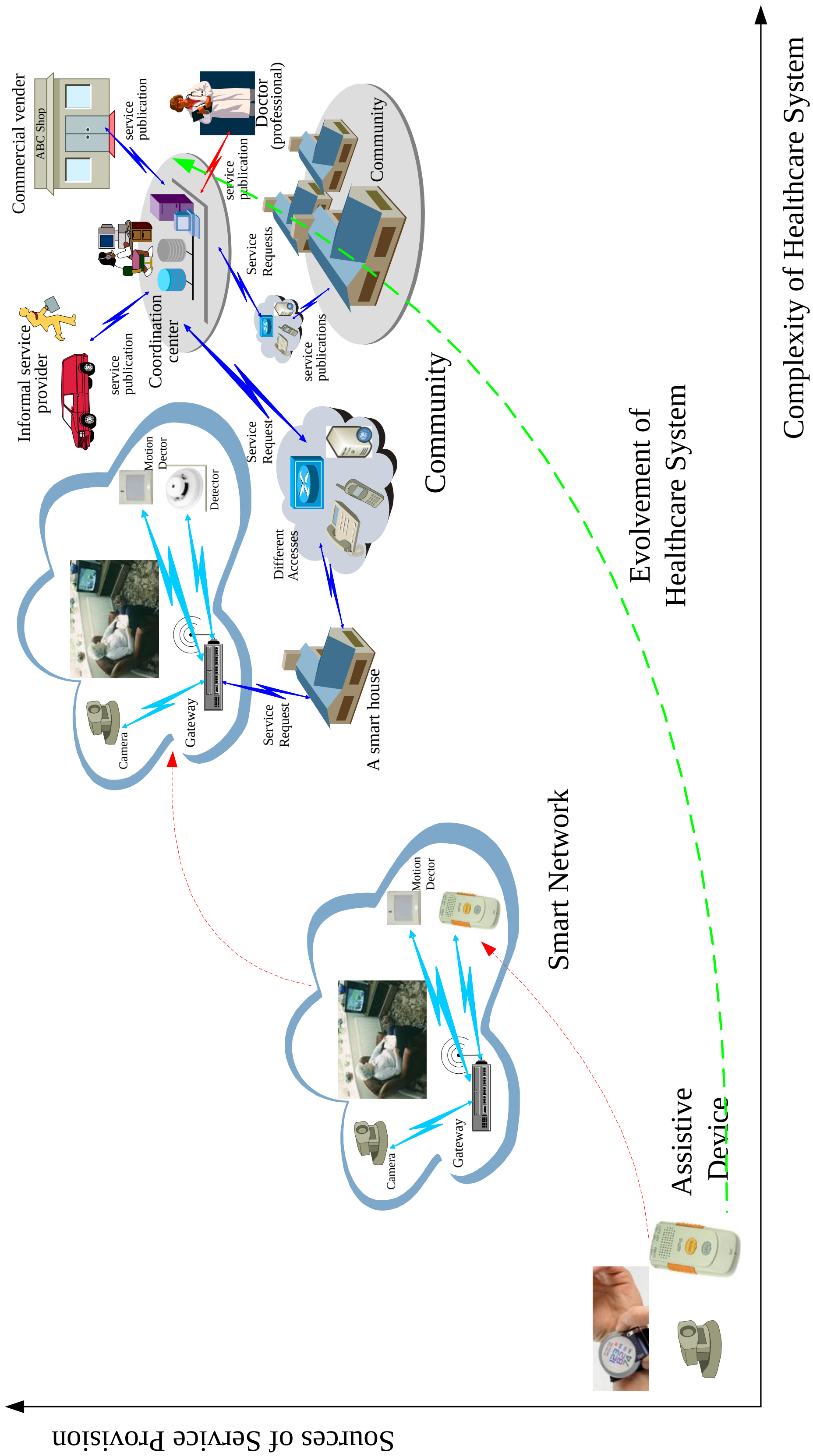


Figure 5.1 The Evolvement of Healthcare System

The adaptations on the first two levels could be implemented with the adaptation approaches proposed in the last chapter by wrapping those assistive devices into service bundles, and coupling with the Aspect Oriented Programming, it could well take the corresponding reactions regarding the changes detected, thus organizing a smart adaptive environment.

Adaptations of the healthcare systems at the community level exhibits a broader view, which is more focused on orchestrating the available resources to meet the requirements of the people who need assistance. Such a system aims to deliver services to the people in need in a prompt way, meanwhile also taking the perspective of the welfare of the whole community. In the following sections of this chapter, I will introduce the service orchestration of the mutual assistance community, which could be considered as a healthcare system at community level.

5.2 Challenges of bringing human participation in Ambient Assisted Living systems

The service orchestration at the community level aims at orchestrating all the available resources across the community, and provide the required service to the people in need. As stated in Chapter 3, the human participation is indispensible in providing efficient and cost effective services to the assisted people. However, there are still many challenges towards the implementation of such an environment. In this section, I first discuss those I deem to be the main challenges, the necessary steps, and then present some preliminary experiments in effectively delivering services which exploit human participation in Ambient Assisted Living.

5.2.1 Dynamicity of service availability

Although informal caregivers may help to reduce the needed social resources, and increase the social connections, they are also very difficult to be utilized, due to their inherently dynamic nature: the availabilities of these services are continuously changing. How to manage this dynamicity becomes a big challenge.

Service Oriented Architecture (SOA) could be an effective approach to cope with the above mentioned dynamicity. SOA is a flexible, standardized architecture that supports the connection of various services, and as such it appears to be an ideal tool to tackle the dynamicity problem. The application of SOA, such as the OSGi platform [OSGi], can also help to establish a framework such that various smart devices could be integrated together and could be automatically discovered, called, started and stopped. The unified service format will also help the process of service matching. Research efforts of using OSGi to build safety home environment are also reported in [Aiello, 2008].

Another attractive feature of SOA can be found in recent research towards integrating people activities into service frameworks, which culminated in two specifications launched in the summer of 2007: Web Services Human Task (WS-HumanTask) [WS-HumanTask] and WS-BPEL Extension for People (BPEL4People) [BPEL4People]. WS-HumanTask targets on the integration of human beings in service oriented applications. It provides a notation, state diagram and API for human tasks, as well as a coordination protocol that allows interaction with human tasks in a service-oriented fashion and at the same time controls tasks' autonomy. A people activity could be described as human tasks in the WS-HumanTask specification. The BPEL4People specification supports a broad range of scenarios that involve people within business processes, using human tasks defined in the WS-HumanTask specification. These two specifications could help to meet the challenges of integrating human services in AAL systems.

5.2.2 Service Mapping

How to let the computer automatically map the available/requested services is a big challenge towards constructing effective AAL systems.

The foundation for service mapping is a service description. A Semantic Knowledge Base is required to precisely describe the advertised services: certain ontology libraries describing the domain knowledge of the home-care environment should be developed with the interdisciplinary collaboration of the researchers in this domain. With such domain knowledge, conceptual models for semantic service matching could be applied. OWL-S [OWL-S] is currently the most used technology in this domain; it is able to provide a framework for semantically describing web services from several perspectives, e.g., service inquiry, invocation, and composition.

There are several service matching tools developed for matching OWL-S services, such as OWL-S Matcher [Tang, 2006] OWL-S UDDI/Matchmaker [Srinivasan, 2004], and OWLS-MX Matchmaker [Klusch, 2005]. These tools serve as good starting points to investigate AAL web service matching, while more elegant and efficient matching engines should be developed. Some preliminary tests of service matching in AAl domain will be presented later this chapter.

5.2.3 Raising User Acceptance

In [Aarts, 2007] [Aarts, 2009], the persuasiveness of ambient intelligence is discussed: in order to achieve human participation, AAL system should provide both trust persuasion and enough motivation to get people's acceptance. As people enter old age, they need to contend with the likelihood of physical and cognitive decline. In addition there may be psychological issues. They may become passive consumers of societal services rather than active creators. In so doing, their self-esteem may suffer. There is a risk that some AAL systems for elderly people may consider their users as people who are weak and need to be passively assisted by others. The technology needs to be designed and implemented to recognize that elderly can still make their contributions to society and have a wealth of valuable experience to share. The proposed mutual assistance community could help encourage elderly people to participate actively in group activities as peer participants, and possibly even to use their experiences to help the younger generations solve issues, for example, work and school problems [Sun et al., 2008b]. Such activities provide elderly people with the chance to live creatively and with greater self-esteem.

Overcoming technical barriers

Elderly people may have more difficulties getting accustomed to the application of new technology than younger people. In order to help them get used to the ambient assistive devices and "bridge the grey digital divide," we must construct user friendly interfaces, and also provide appropriate training for users. Developing adaptive, straightforward, and multimodal human computer interfaces is a challenge in terms of the design of user interfaces to assisted living technologies [Kleinberger, 2007]. The issue of improving the ICT knowledge of the elderly people so as to include them in the information society is receiving more and more attention from governments, and appropriate courses are provided to the elderly people in many countries. It is proposed that people should be involved in training on how to use assistive devices before they really need it [Floeck, 2007].

Technical barriers can be bridged when people have a strong drive to do so. Social connectivity is one such drive. Andreessen, former CEO of Netscape, recalled how people broke the technical barrier to surf online when Netscape first came to the public:

> There was no shortage of skeptics at the time, who said that none of this would work because it was all too complicated. ‘You had to go out and get a PC and a dial-up modem. The skeptics all said, 'It takes people a long time to change their habits and learn a new technology.' [But] people did it very quickly, and ten years later there were eight hundred million people on the Internet. People will change their habits quickly when they have a strong reason to do so, and people have an innate urge to connect with other people. [Friedman, 2005, pp. 62-63]

In a mutual assistance community, elderly people not only benefit from keeping connected with other people, but also are provided with chances to make contributions to society, to feel that they are still useful, and to live in an active way with increased self-esteem. Such benefits could provide our elderly people with further stimuli to break the grey digital divide.

5.3 Service Orchestration in Mutual Assistance Community

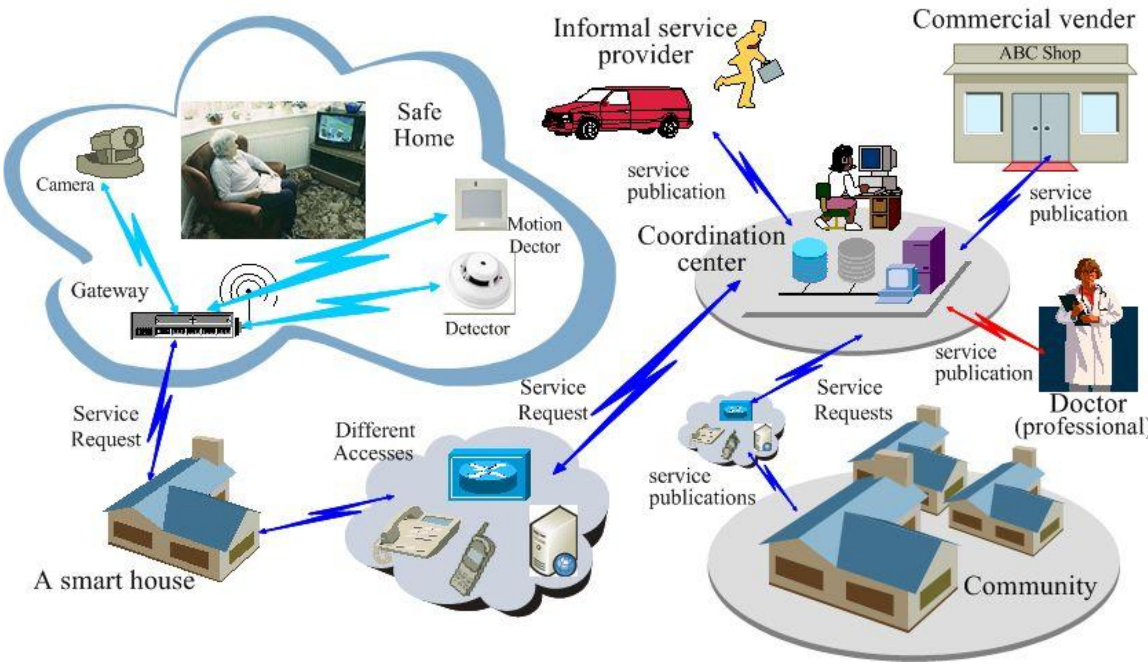


Figure 5.2 Organization of a Mutual Assistance Community

In Chapter 3, I have presented the structure of a mutual assistance community, which aims to orchestrate the medical resources at the community level. In this chapter, I discuss the issues of connecting the different dwellers in this community together; the techniques of publishing their request or availability of services; and the approaches of matching service provider and service requester.

5.3.1 Integrating different services

The ambition of the service orchestration of a mutual assistance community is to set up the software backbone, where services from human side and machine applications could be seamlessly integrated in the community framework. Both services from the assistive devices and the human services are published in a common format and automatically processed.

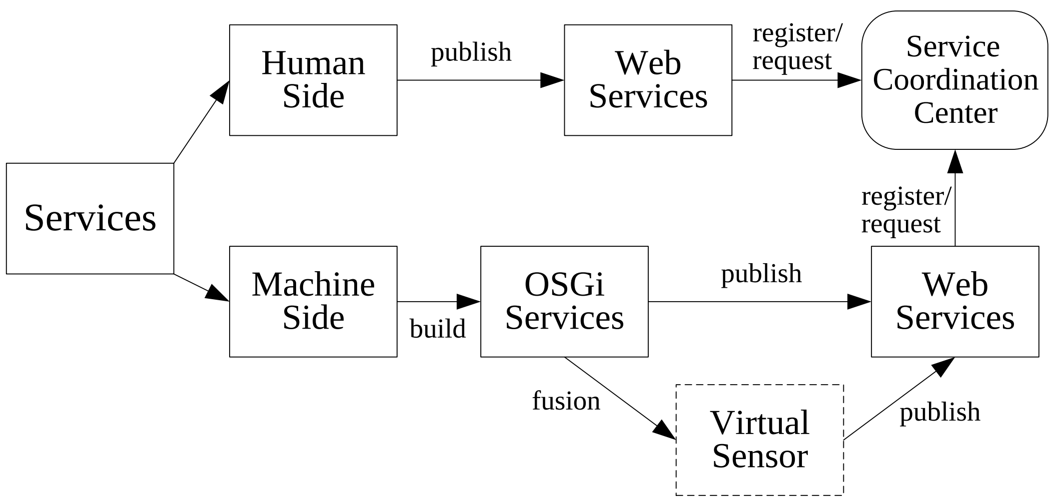


Figure 5.3 Structure of service integration

Figure 5.3 shows one possible approach to realize such ambition. In the framework shown in Fig 5.3, services from assistive devices and human tasks are processed separately and later integrated together. Services from assistive devices are built as OSGi bundles (in OSGi, a bundle could be treated as an application or component), and coordinated by the OSGi gateway. Part of these services will be published directly as web services in the gateway, while the others will be firstly merged together into composite indicators to abstract higher level information and then published as web services. Such services are named as “virtual sensors.”

On the human side, service processing is more straightforward. Human services are published as web services. Specifications such as WS-HumanTask [WS-HumanTask] and BPEL4People [BPEL4People] are efforts towards the integration of human activities into the service oriented application; they could help to facilitate process of publishing human services. The availability to provide services, e.g. as informal caregivers, is registered in the service coordination center while the request from the elderly people is also posted as service requests. A user could be registered as a service consumer and a service provider at the same time, based on the type of services. For instance, an elderly person could request physical challenging services, while providing knowledge or experience-based advisory services.

In the real-life implementation, there are many restrictions hindering the implementation of the above mentioned targets. These limitations will be presented in the following sections. In this thesis, I present my implementation of a simple example of service matching, which is based on the matching the parameter of service type and service provider in OWL-S [OWL-S]. The binding process after the service matching is not discussed in this thesis; this is because the invocation of human services normally requires manual confirmation, which is different with the automatical invocation the remote web services from computer servers.

5.3.2 Architectures of Service Discovery

In order to best utilize the available resources in the mutual assistance community, it is important to have a common way to present the available resources together with effective ways to discover the resources. In the Web Service Architecture technical report by the W3C, three leading viewpoints for web service discovery are discussed: as a centralized registry, as an index, and as a peer-to-peer system.

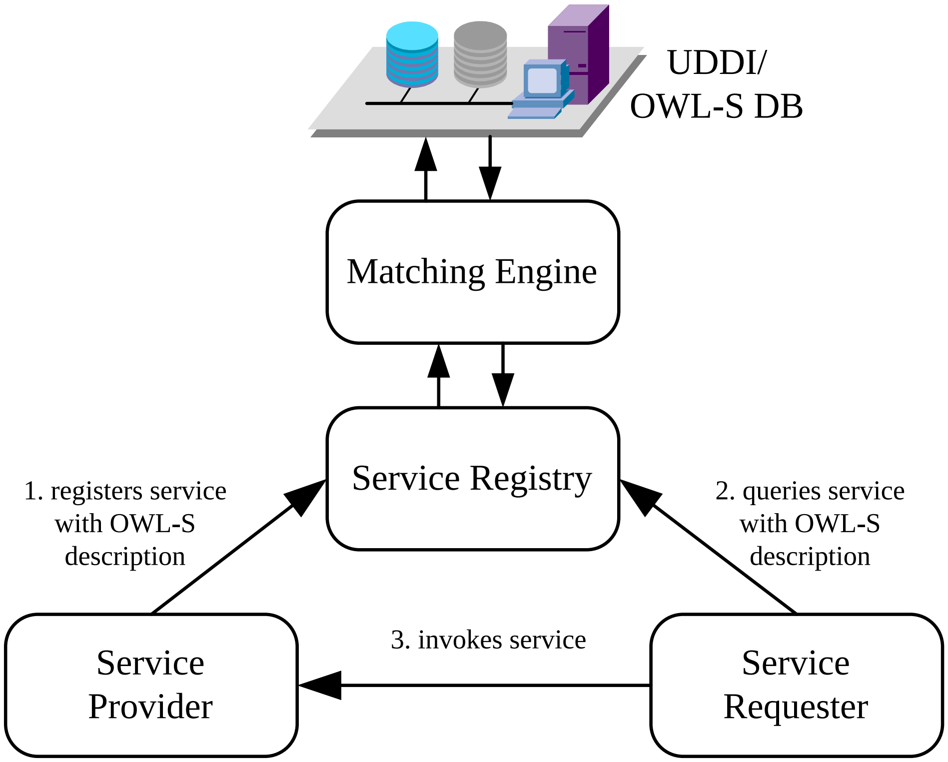


Figure 5.4 Centralized Architecture for Service Discovery

In the registry architecture, a central authoritative registry stores and controls service descriptions, which are explicitly submitted by the service providers. UDDI [UDDI] is often seen as an example of such a service repository, though it may also be applied in the index architecture. Figure 5.4 shows the structure of the centralized architecture for service discovery. When a service requester needs services, query of services is submitted to the service registry. The server processes this request by matching the requirements with the description of the advertised service. If the server finds matched service(s), the matched result(s) will be returned to the service requester, so that it could invoke the service from the service provider.

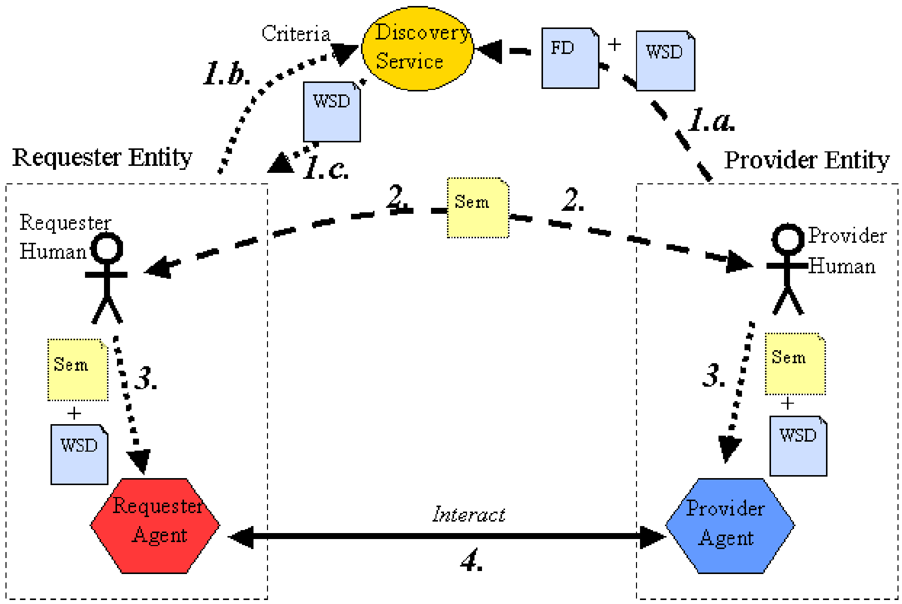


Figure 5.5 The process of engaging a web service [http://www.w3.org/TR/ws-arch/]

To better illustrate the process of service discovery, and interconnection, Fig 5.5 shows an example of engaging a web service:

1. The service requester and service provider "become known to each other":
    a. The provider publishes both the Web service description ("WSD") and an associated functional description ("FD") to discovery service (service registry).
    b. The requester entity supplies criteria to the discovery service to select a Web service description based on its associated functional description, capabilities and potentially other characteristics.
    c. The discovery service returns one or more Web service descriptions (or references to them) that meet the specified criteria.
2. The requester and provider entities agree on the semantics of the desired interaction.
3. The service description and semantics are input to, or embodied in, both the requester agent and the provider agent, as appropriate.
4. The requester agent and provider agent exchange SOAP messages on behalf of their owners.

The index approach is quite similar to the registry approach; the major difference is that in the index approach, the service publication is carried out in a passive way: unlike the service providers advertising their services in the service registry, in the index approach, service providers do not submit their services to the service registry, instead, a server will automatically collect all the available services from the web to create an index of available services. Google is often cited as an example of the index approach.

The peer-to-peer approach is an alternative that does not rely on centralized registry. It allows web services to discover each other dynamically. In such an architecture, a web service is a node in a network of peers. When a service requester needs services, the query of services is submitted to the service discovery. The service discovery will contact their neighbouring peers to find proper service: if any one of the other peers could match the service and replies to the service discovery, the service address of the matched peer will be returned to the service requester, and the searching process will be stopped. Otherwise, each node queries its own neighbouring peers and the query propagates through the network until the requested service is found or other termination criteria are met. In addition, in the P2P architecture, each node may contain its own indexing of the existing web services which are known to it.

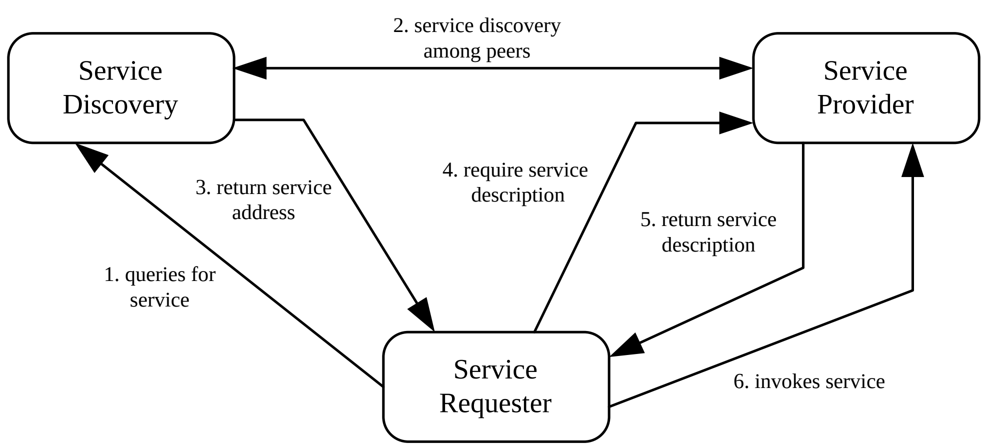


Figure 5.6 Peer-to-peer Architecture for Service Discovery

In the past research in ambient assisted living, most of the approaches connect the neighbours or relatives to constitute a small, statically defined network around the assisted people [AwareHome, 2008] [Amigo, 2007] [Persona, 2009]. Architectures for service discovery for a large community are not discussed in those projects. Healthcare community, e.g. the proposed mutual assistance community, has its own characteristics – the community size could be large and many services are provided by human beings, which are hard to program and manipulate in automatic way. None of the architecture mentioned above could well fit such a community alone for the following reasons:

In the application of the proposed mutual assistance community, the index approach is not appropriate as the service availability of the mutual assistance community is highly dynamic, especially when the services from the human side are included. Passively collecting the available service could not guarantee the service provider could actually provide the desired service when it is invoked.

The other two approaches, the centralized and Peer-to-Peer (P2P) architecture, both have their advantages and disadvantages. The advantage of the centralized approach is that such structure is simple and efficient in execution; however, the disadvantage is the threat of single point of failure at the service registry. On the other hand, for the P2P architecture, the advantage is that there is no single point of failure, meanwhile, it also guarantees the available service is up to date; however, the disadvantage of the P2P architecture comes with performance costs and lack of guarantees of predicting the path of propagation. The size of the proposed mutual assistance community could be rather large, querying services in such a large community in P2P may generate too many enquires, and it is also time consuming in discovering a service in a large community. Especially in the healthcare community such as the mutual assistance community, many services are provided by human beings, enquiring service availability to a peer (in this circumstance, a person) also indicates interrupting that person to check for availability, which means unnecessary interruption.

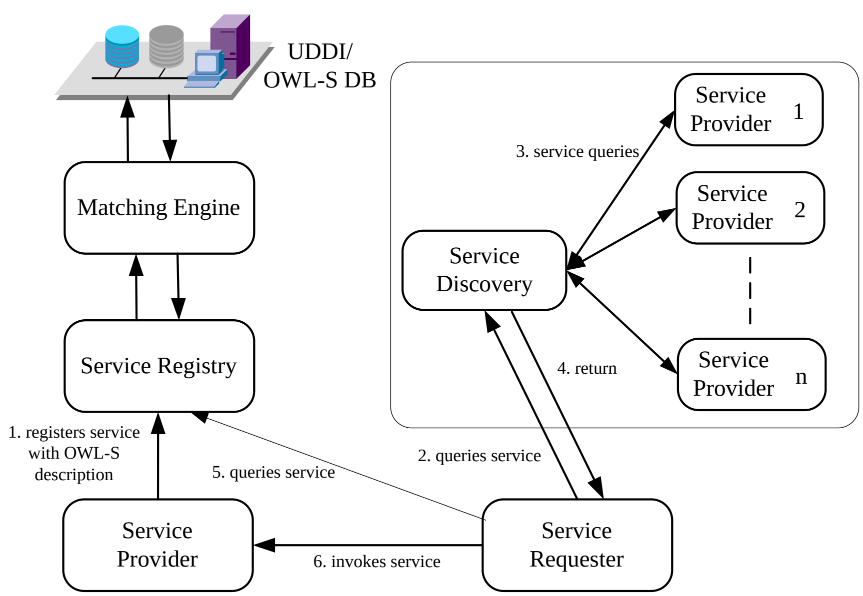


Figure 5.7 Hybrid Architecture for Service Discovery

With respect to the characteristics of the mutual assistance community, a hybrid architecture shown in Figure 5.7 could be adopted to combine the advantage of both P2P architecture and the centralized registry. In the hybrid architecture, a small size network would be built around each dweller of the community, consisting of the frequent contacts of that person, e.g. relatives, neighbours, family doctors, etc. Up to date availabilities inside such a network could be guaranteed by querying the small network, while querying availabilities throughout the whole community would be carried out in the centralized service registry. The organization of service discovery in the small network is similar with those carried out in the peer-to-peer architecture, except for the queries to the service providers are directly sent to the service providers from the service discovery service sequentially.

Once a dweller in the mutual assistance community submits a request, a local service discovery service tries to discover the services from the small network. Once a node in the network is committed to provide the needed service, the service discovery process is stopped, and such a service is subsequently carried out. In case no response is found in the small network, the request is then sent to the centralized service registry to check for availabilities from other community dwellers.

For the implementation of the service discovery architecture in the mutual assistance community, though not carried out yet, the service registry will be implemented in the service coordination center while the service discovery will reside in the local gateway. In the real application, the sequence of checking the small network and the centralized service registry could be configured based on users' preference.

5.3.3 Semantic Service Matching

The previous section discusses the architectures for service publication and discovery; however, getting access to the service description is not enough in finding out which advertised service could provide the requested service. Semantic descriptions are required in describing the functions of the published and requested services; and semantic matching tools are required to compare the advertised and requested services.

Industry efforts to standardize web service description, discovery and invocation have come to standards such as WSDL [WSDL] and UDDI [UDDI]. However, these standards, in their current form, suffer from the lack of semantic representation. The notion of Semantic Web services [Berners-Lee, 2001] [Li, 2003] takes us one step closer to interoperability of autonomously developed and deployed Web services, where a software agent or application can dynamically find and bind services without having a priori hard-wired knowledge about how to discover and invoke them.

The Ontology Web Language for Services (OWL-S) is one of the most widely used approaches in bringing rich semantics to web services [OWL-S] [Martin, 2004], which provides a framework for semantically describing Web services from several perspectives, for instance, service inquiry, invocation, composition. OWL-S is built upon OWL [OWL], which is a language for defining and instantiating Web ontologies. The OWL defines the classes, properties, and their instances in a hierarchical view, thus the data represented with OWL could be processed with semantic reasoning. The semantic reasoning of OWL-S is based on the assumption that the definitions of the semantic concepts/data in the OWL-S service, which contained in a common ontology library, are already presented by the OWL and is available for the reasoning tools.

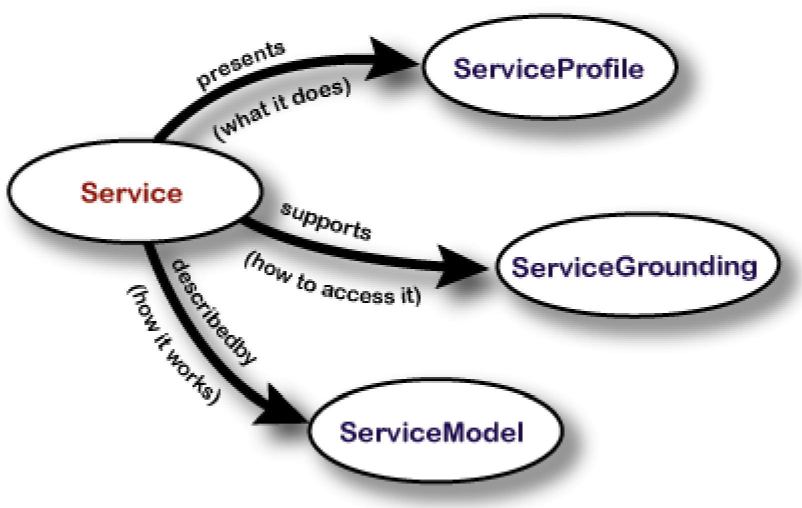


Figure 5.8 The Top Level Organization of OWL-S Service

The properties contained in the description of OWL-S service are shown in Fig 5.8 [OWL-S]. An OWL-S service is described by three main parts, service profile, service grounding and service model. The service profile describes the function of a service, i.e. what does the service provide for perspective clients, or what service is needed from the service requester. The other two properties define how to use the advertised service: the service model tells a client how to use the service. It describes how to ask for the service and what happens when the service is carried out. The service grounding answers how to interact with the service; it specifies the details of how an agent can access a service.

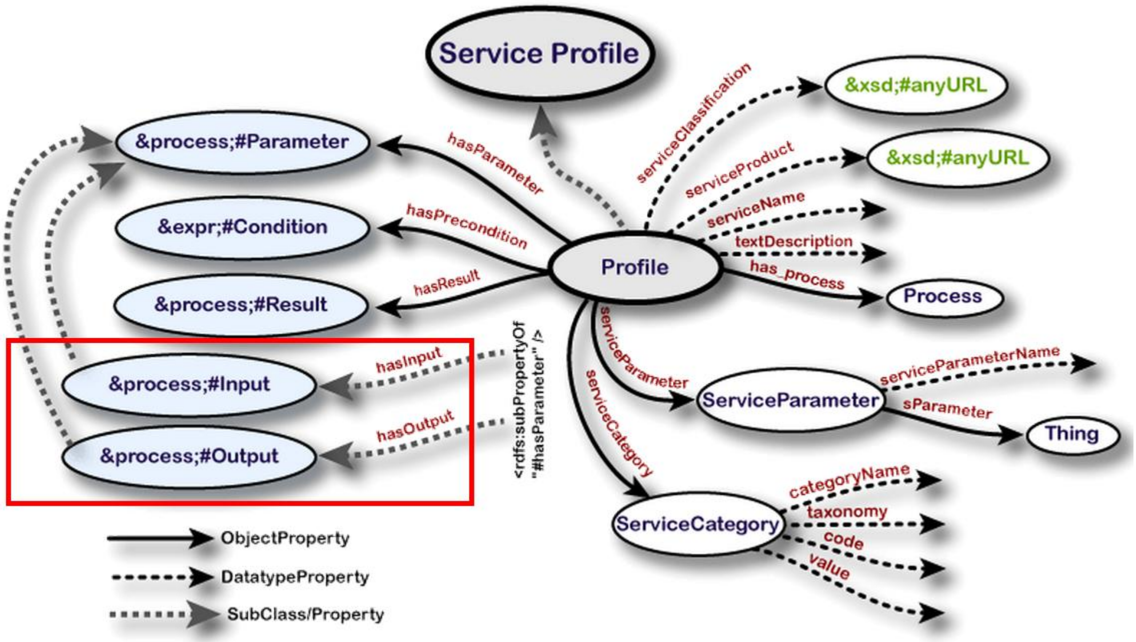


Figure 5.9 A Sample of Service Profile Attributes

The service profile contains much information of a service ranging from the service categories, service parameters, etc. Fig 5.9 [OWL-S] shows an example of the attributes that are included in the service. However, in the algorithms for semantic service matching, the matching result between the advertised and requested services are based on the comparison of the inputs and outputs (as marked in the red box in Fig 5.9) in each service [Paolucci, 2002].

```
outputMatch(outputsRequest, outputsAdvertisement) {
   globalDegreeMatch= Exact
   forall outR in outputsRequest do {
      find outA in outputsAdvertisement such that
         degreeMatch= maxDegreeMatch(outR,outA)
         if (degreeMatch=fail) return fail
         if (degreeMatch<globalDegreeMatch)
            globalDegreeMatch= degreeMatch
   return sort(recordMatch);}
```

Figure 5.10 Algorithm for output matching

Figure 5.10 shows the algorithm for output matching, and the input matching are carried out in the similar approach [Paolucci, 2002]. Both output and input are represented by the ontology predefined in the OWL library, so that semantic reasoning could be carried out.

The comparison of the inputs and outputs in the advertised and requested services are carried out by the semantic reasoning tools, which may obtain a match degree between the compared concepts. In such a process, Description Logic (DL) reasoners are used to match the relationships of two concepts. Various DL reasoners are developed, e.g. RACER [RACER], Pellet [Pellet], OWLJessKB [OWLJessKB]. Those reasoner tools firstly parse the ontology library to create a knowledge base and classify the knowledge base to compute the inferred hierarchy of the ontologies listed in the ontology library, and then reason over the knowledge base to extract the relationships between the two concepts being compared.

The Description Logic reasoners are the core of semantic web service matching tools. The general process for semantic service matching is as follows:

1) Extract the service profile from the web service description of advertised and requested services. Extract the inputs and outputs used in the profile.
2) Match the inputs and outputs between the advertised and requested services using Description Logic reasoner.
3) Get the matching relation between the advertised and requested services in inputs and outputs respectively. Compare the received results with the requirement, and return the final result of the matching process.

There are some service matching tools developed for matching OWL-S services, such as OWL-S Matcher [Tang, 2006], OWL-S UDDI/Matchmaker [Srinivasan, 2004], and OWLS-MX Matchmaker [Klusch, 2005]. A review of these semantic service matching tools is summarized in [Georgantas, 2006]; the main drawback of these tools is that the matching process takes a large amount of time, which restricts the application of these tools in matching requested service from a large number of advertised services.

Although more elegant and efficient matching engines should be developed in solving the above mentioned challenges, in the current stage, these tools serve as good starting points to investigate web service matching. In the remainders of this chapter, OWL-S Matcher is used to build up a proof-of-concept application of the semantic service matching in the proposed mutual assistance community.

5.3.4 OWL-S Matcher

OWL-S Matcher [Tang, 2006] is a JAVA implementation of a matching algorithm for matching requested and advertised OWL-S services by comparing service descriptions. The OWLJessKB reasoner is embedded in the OWL-S Matcher to execute the matching process. After manually loading the advertised and requested services, the user sets their minimum requirements for the matching degree on the inputs and outputs respectively. The inputs and outputs from these two are reasoned by the OWLJessKB reasoner, and the matching results are compared with the user's requirement. A final result is then produced after such comparison.

The matching degree is an essential part in the OWL-S Matcher [Jaeger, 2004]; it represents a similarity degree between requested and advertised services. By providing the matching degree of the assigned ontology, the relationship between the requested and advertised services can be obtained. Let A and B denote two concepts of ontology in the requested and advertised services. The matching degree, in other words, the relationship between A and B could be defined as in Table 6.1:

Table 6.1. Matching Degree

Rank	Degree of match	Explanation
0	Fail	The requested and advertised class do not match.
1	Unclassified	At least one of the classes is unclassified.
2	Type_invert	The class in the advertised service is a sub-class of the one in the requested service.
3	Type_subsumes	The class in the requested service is the sub-class of the one in the advertised service.
4	Match	The requested and advertised class match.

The OWL-S Matcher allows users to set their own requirements of the matching degree between the advertised and requested services. Even the advertised and requested concepts are not exact match, if the user accepts the difference degree, the final results of the two services can still be considered as match. Such options can help to remove the expression difference on service description and increase the chance of finding appropriate services for the service requester. For example, a service request for a "laptop" may be fulfilled by a "Macbook". In the following Experiments Section, I will give a more detailed example of how to carry out semantic service matching in the mutual assistance community with the OWL-S Matcher tool.

5.4 Experiments

The experiment is carried out as an example to show the semantic service matching among the proposed mutual assistance community [Sun, 2007c]. One major characteristic of the services in the proposed mutual assistance community is that all the dwellers of the community are encouraged to provide their services when they are able to, which brings various categories of service providers. The service matching process is aware of such characteristic, and focuses on the comparison of the *Service Type* (indicating the function of a service) and Service Provider (indicating the service organization).

In the experiments, some primitive hierarchical ontology is constructed to represent different service types and different service providers. Web services are also constructed representing requested and advertised services respectively. The mapping process between the requested and advertised services is executed to test how the different degrees of mapping could be obtained.

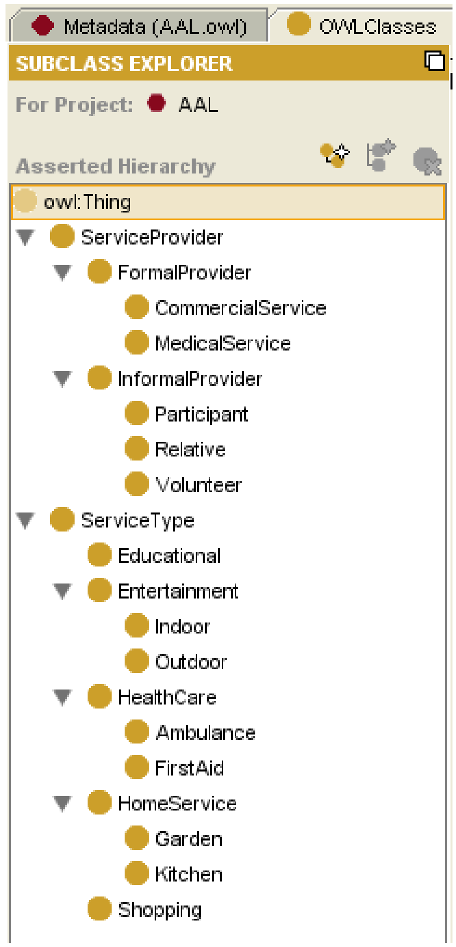


Figure 5.11 Hierarchical Classes in Protégé

Figure 5.11 illustrates some primitive hierarchical concepts that were built in Protégé [Horridge, 2004], which work as the ontology library for the service matching experiment. The concepts described in the ontology library are made up by two main classes: *Service Type* and *Service Provider*. The *Service Type* concepts define a series of services in people's daily lives, such as educational, healthcare, etc. The services displayed in Fig 5.11. are defined in a brief way which are only used for the demonstration purpose; a much larger and more detailed ontology library should be developed when carrying out the application into a real community. Services are categorized as different type of services in the *Service Type* concepts, those concepts are built with hierarchical view, which could help to find alternative services when exact match is not available. For example, *Garden Service* is a subclass of the *Home Service*, if a requester for *Garden Service* can not find an exact match, the *Service Provider* who provides *Home Service* may also be considered as match, if the user so allows.

The *Service Provider* concepts indicate the provider of service, e.g. formal provider, informal provider, etc. The concepts of *Service Provider* also implicitly reflects the format of how a service is organized. As introduced in Chapter 3, dwellers in the mutual assistance community are encouraged to make contributions as informal service providers, e.g. volunteers; a "participant model" is also introduced in encouraging people to participate group activities as peer participants. In the semantic service matching process, service providers or requesters may specify how the service activities are organized by specifying appropriate service providers. E.g. choosing a *Service Provider* as *Informal Provider* indicates the service will be provided by informal service provider, while specifying it as *Participant* means that the service will be organized as *group activity*.

```
</profile:textDescription>
<profile:hasInput  rdf:resource="#_InformalProvider"/>
<profile:hasOutput  rdf:resource="#_Entertainment"/>

<profile:has_subclass rdf:resource="AAL_SERVICE_PROCESS" />

<profile:has_process rdf:resource="AAL_SERVICE_PROCESS" />
</profile:Profile>

<process:ProcessModel rdf:ID="AAL_SERVICE_PROCESS_MODEL">
<service:describes rdf:resource="#AAL_SERVICE_SERVICE"/>
<process:hasProcess rdf:resource="#AAL_SERVICE_PROCESS"/>
</process:ProcessModel>

<process:AtomicProcess rdf:ID="AAL_SERVICE_PROCESS">
<process:hasInput  rdf:resource="#_InformalProvider"/>
<process:hasOutput rdf:resource="#_Entertainment"/>
</process:AtomicProcess>

<process:ServiceType rdf:ID="_Entertainment">
<process:parameterType rdf:resource=
"http://127.0.0.1/ontology/my_ontology.owl#Entertainment" />
  <rdfs:label></rdfs:label>
</process:ServiceType>

<process:ServiceProvider  rdf:ID="_InformalProvider">
<process:parameterType rdf:resource=
"http://127.0.0.1/ontology/my_ontology.owl#InformalProvider" />
  <rdfs:label></rdfs:label>
</process:ServiceProvider>
```

Figure 5.12 Excerpts from the Advertised Service in OWL-S

After the definition of the hierarchical classes, the advertised and requested services are written in OWL-S using the classes in Fig 5.11. Both services are very simple and only specify the Service Provider and Service Type. In the requested service, the requester is asking the informal provider to provide some *indoor entertainment* service. In the advertised service, the service provider is indicated as *informal provider*, and the available service is indicated as *entertainment service*.

Figure 5.12 shows a fragment of the advertised service in OWL-S. The #Informal Provider and #Entertainment ontologies are declared in the my_ontology.owl file, which defines the ontology in Fig 5.11. The *service provider* and *service type* are specified as inputs and outputs of the service. The service matching process will be carried out by comparing the *service provider* and *service type* of the compared services.

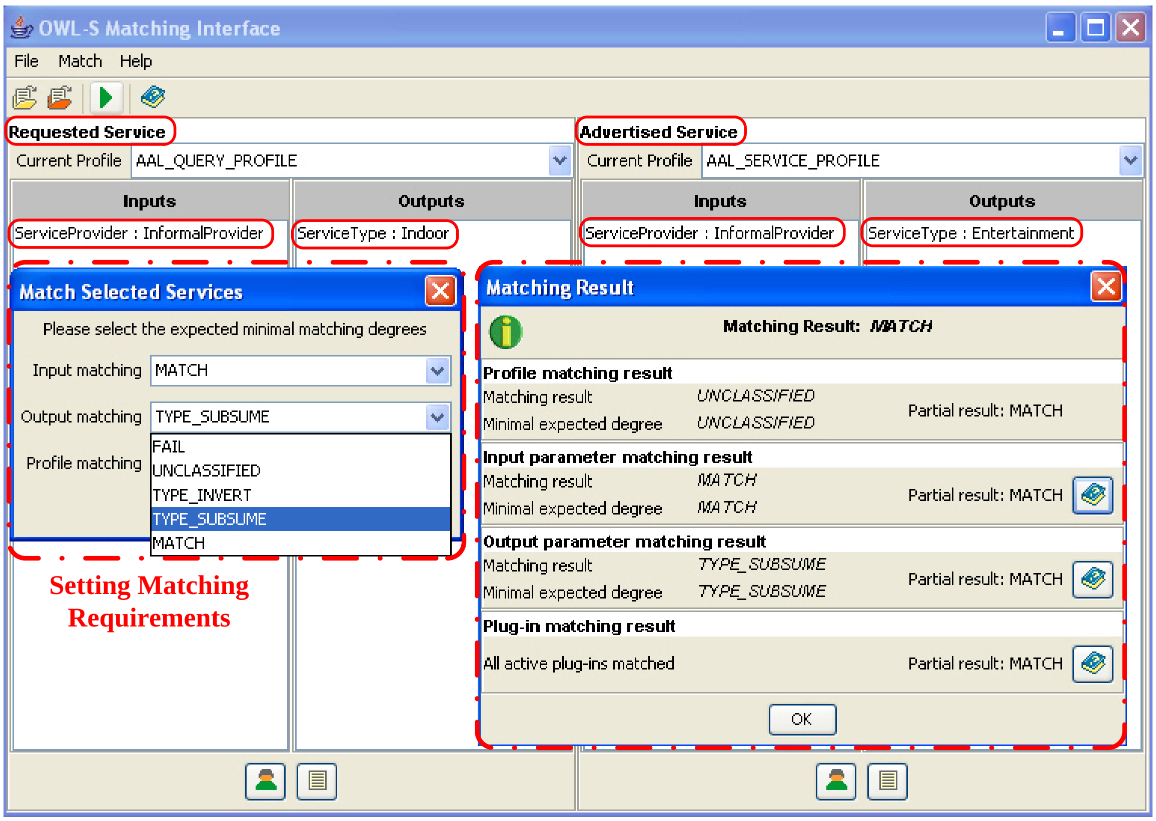


Figure 5.13 OWL-S Matcher Interface and Matching Result

Figure 5.13 shows the experiment running with the OWL-S Matcher. Firstly, the advertised and requested services are loaded respectively, and then the user's requirements on the matching results are specified. In the case shown in Fig 5.13., the acceptance degree for the input (*Service Provider*) is selected as 'Match'; while for the output (*Service Type*), the acceptance degree is 'Type_subsumes'.

The matching result is shown in the central right window in Fig 5.13. It indicates that for the input (*Service Provider*), the matching result is "Match". That is because in both advertised and requested services, the input is exactly the same (in this case, they are set as *Informal Provider*). This result meets the minimal expected degree. Meanwhile, the matching result of the output (Service Type) also meets the expectation, thus the final result for this matching process is “Match”.

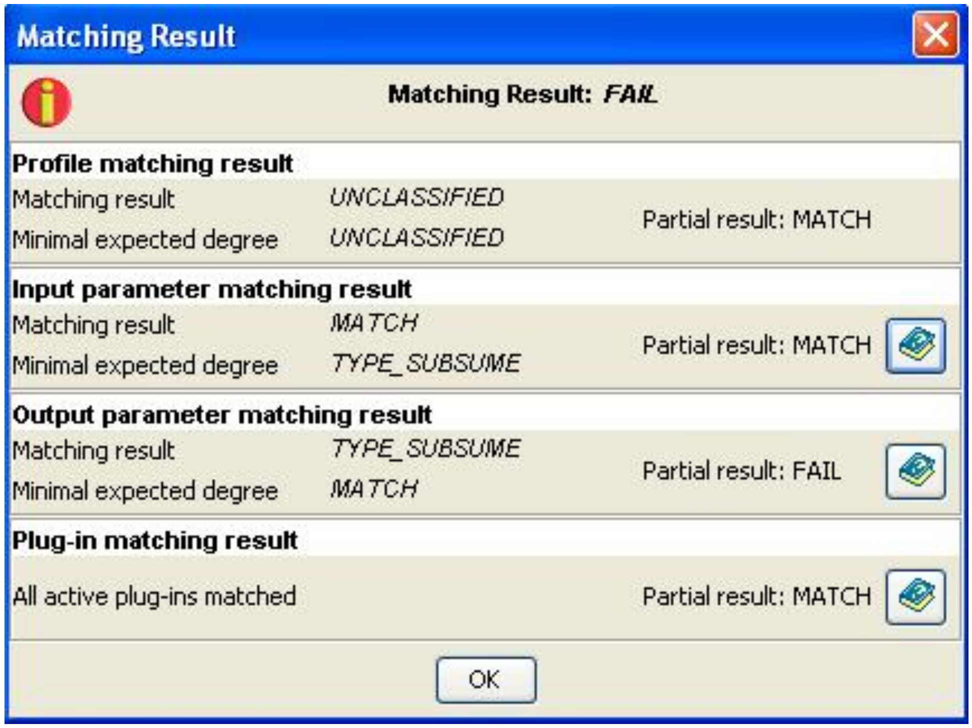


Figure 5.14 Matching Result II

Figure 5.14 shows the matching results after the selection of the minimal expected degrees. It can be seen that the input matching result “Match” still meets the expect degree “Type_subsumes”; while for the output, the result “Type_subsumes” can not meet the expected degree “Match”, so that the final result for the output matching is “Fail”.

The experiments shows that through the OWL-S Matcher, the matching process between the advertised and requested services could be effectively executed. Relationships between the requested service and those advertised could be achieved in the matching process. Service requesters could set different expected matching degrees for the required service. By separating the service function (*Service Type*) and service organization (*Service Provider*), the requester could specify different acceptance degrees on function and organization, which not only increases the flexibility in the matching process, but also help to find alternative services when exact match is not able to achieve.

By listing the *Service Provider* as a key target of the service matching process, service activities could be carried out with various forms proposed in the mutual assistance community, e.g. providing the service as informal caregivers as well as organizing group activities where the elderly people could join as peer participants.

5.5 Scenario

To better illustrate the functions of our proposed mutual assistance community, I will use a scenario to explain how the community organizes group activities [Sun, 2007].

Scenario: Participation in Group Activities

Mary is 70 years old and lives alone in Antwerp. In the afternoon of a sunny day, Mary wants to take a walk in Middelheim Park, which is close to her home. She wants to have somebody accompanying her during her walk. Mary decides to use the mutual assistance community to find someone who also wants to walk in the park. She switches on the TV, which is the graphical interface of the mutual assistance community. She navigates the service menu, which is built as an ontology tree, and selects the "Group Activity". A few photos will be presented to her, representing group activities such as chatting, exercising together, etc. Mary chooses the symbol for walking and types in the location where she wants to hold this event as "Middelheim Park". For the service organization, Mary chooses to receive service from peer "participants", specifies the deadline as "today, 8pm".

After these inputs, the service matching engine starts searching for the appropriate services. The service matching engine is located in the service matching center. Available services are advertised in the service matching center which then produces a list of available services, and ranks them by their relevance to the users' requirements. The settings for the search, translated from the requirements of Mary are as follows:

Required Service: Walk with someone.

Location: Middelheim Park, Antwerp.

Deadline: 8pm, 1st, August, 2007.

Service Form: Participant of Group Activities.

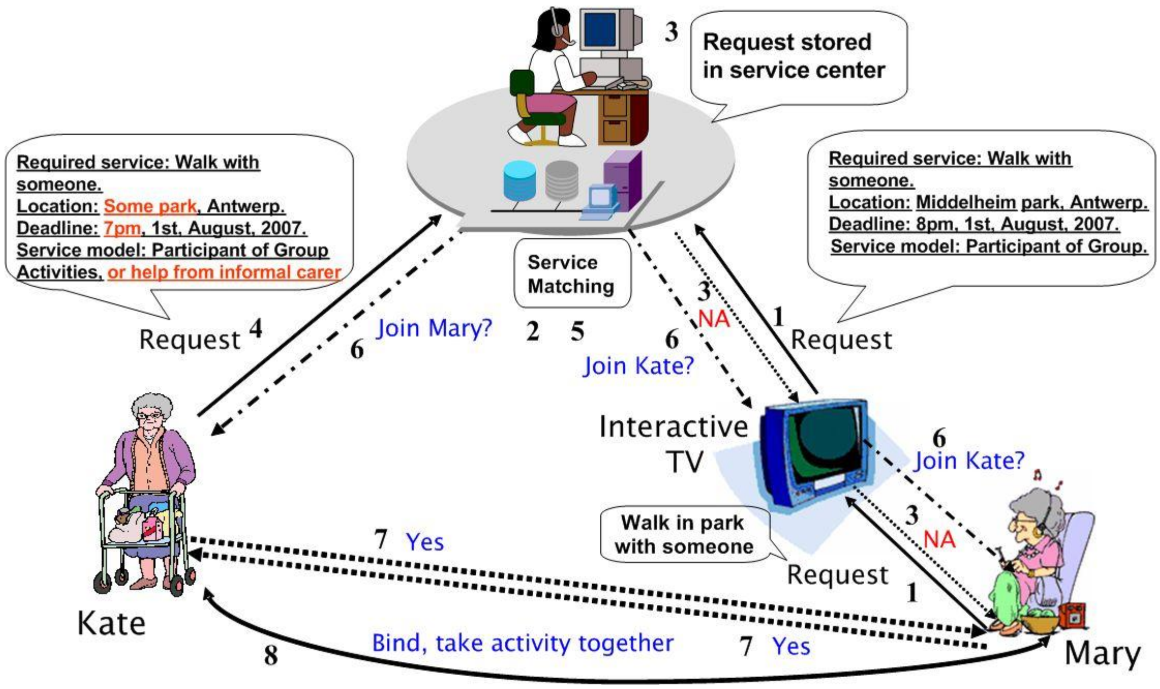


Figure 5.15 A Scenario of Organizing a “Participant Activity”

Unfortunately, the service matching engine could not find a matched or similar service for Mary, so it will notify Mary that no matched services/requests are found. Nevertheless, the request from Mary is still kept in the service center until the deadline arrives.

Luckily, some minutes later, Kate submits a similar request, which is also displayed in Fig 5.15. This time the system finds that Kate’s request is similar to the one from Mary. The degrees of match on different criteria are shown in Table 6.2.

Table 6.2 Service Matching

Criteria	Mary	Match Degree	Kate
Required Service:	Walk with someone.	Match	Walk with someone.
Location:	Middelheim Park, Antwerp.	Sub_class of	Some Park, Antwerp.
Deadline:	8pm, 1st, August, 2007.	Parent_class of	7pm, 1st, August, 2007.
Service Form:	Participant of Group Activities.	Sub_class of	Participant of Group Activities, or Informal Caregiver

The screen in Mary's interactive TV will notify her that it is now possible to meet her request: "*Kate* initiated a *walking* activity in *some park (includes Middelheim Park)* by *7pm today*; if you agree to take this *group activity* with *Kate*, press the **confirm** button and we will forward *Kate* your contact information."

Mary agrees to join this activity and presses "confirm", while Kate also confirms the participation of Mary, agreeing on Middelheim Park as an acceptable venue. Their contact information is then displayed on their TV screens respectively, they call each other to confirm the time and place to meet, and later have an enjoyable time walking in the park together.

In this scenario, Mary and Kate take the group activity – walking in the park – together. No additional help is required to meet their requests, so that social resources are effectively saved, and they also avoid the possible frustration of having to be taken care of by others. Figure 5.15 illustrates the actions taken in this scenario; the numbers displayed indicate the steps of the indicated action, the indicator "NA" means "not available".

The above mentioned scenario is not yet implemented in the real-world application: in order to implement the mutual assistance community and bring such a scenario into real-life, advanced human computer interfaces need to be developed to facilitate the publish and request of web services. The elderly people should also be trained to use the ICT devices and reduce the so-called "gray digital divide" [Soar, 2011].

5.6 Conclusion

This chapter introduces the service adaptations in a broad view – the perspective of community level. The healthcare system is evolving from the separated assistive devices to the networked environment, and finally moving towards the organized social community. The proposed mutual assistance community envisaged such evolvement, and this chapter discusses the service adaptation in the mutual assistance community. Service adaptation in such a community is mainly about orchestration the community resources, discovering the available services and providing them to the requested people in prompt way.

Semantic service description, discovery and matching techniques are introduced in this chapter, which help to discover and utilize services inside a community. A hybrid service discovery architecture, which combines the advantage of both service registry architecture and the peer-to-peer architecture for service discovery, is proposed for discovering services in the mutual assistance community. The general process of using OWL-S service for semantic service description and matching are also introduced in this chapter.

Experiments on semantic service matching in the mutual assistance community are carried out using the OWL-S Matcher. The results demonstrate that the various service models in the mutual assistance community, e.g. informal caregiver services, “participant model”, could be carried out through semantic service matching. The experiments of service matching shown in this chapter are carried out by comparing the service provider and service type, other metrics e.g. the location of the service provider/requester, etc., should also be taken into account in the real-life implementation. A scenario is also presented which shows how complex service matching to organize group activities carried out by peer participants is possible in real-world application.

References

Aarts, E., Markopoulos, P. and de Ruyter, B. (2007), The persuasiveness of Ambient Intelligence, in: M. Petkovic and W. Jonker (eds.), Privacy and Trust in Modern Data Management, Springer, Berlin, Germany, pp. 367-381.

Aarts, E., de Ruyter, B. (2009), New research perspectives on ambient intelligence. Journal of Ambient Intelligence and Smart Environments, 1(1), pp. 5-14.

Aiello, M. and Dustdar, S. (2008), Are our homes ready for services? A domotic infrastructure based on the Web service stack. Pervasive and Mobile Computing 4(4), pp. 506-525.

Amigo (2007). Amigo: Ambient intelligence for the networked home environment. Retrieved August 24, 2009, from http://www.hitech-projects.com/euprojects/amigo/.

Aware Home (2008). Aware Home Research Initiative Project, Georgia Institute of Technology. Retrieved August 24, 2009. from http://awarehome.imtc.gatech.edu/.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001), The Semantic Web, Scientific American.

BPEL4People (2007) Specification: WS-BPEL Extension for People, (BPEL4People). Retrieved August 24, 2009, from https://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/30c6f5b5-ef02-2a10-c8b5-cc1147f4d58c.

COPLINTHO (2008). COPLINTHO Project, IBBT. Retrieved August 24, 2009, from https://projects.ibbt.be/coplintho/.

Floeck, M., Litz, L. (2007), Aging in Place: Supporting Senior Citizens' Independence with Ambient Assistive Living Technology. mst / news, December 2007,pp. 34-35.

Friedman, T. L. (2005). The world is flat – A brief history of the twenty-first century. Farrar, Straus and Giroux.: United States.

Georgantas, N. (2006). Amigo middleware core: Prototype implementation & documentation. IST Amigo Project Deliverable D3.2.

Horridge, M., Knublauch, H., Rector, A., Stevens, R. and Wroe, C. (2004). A Practical Guide To Building OWL Ontologies Using The Prot ́eg ́e-OWL Plugin and CO-ODE Tools Edition 1.0, University of Manchester, Stanford University.

Kleinberger, T., Becker, M., Ras, E., Holzinger, A. and Muller, P. (2007). Ambient Intelligence in Assisted Living: Enable Elderly People to Handle Future Interfaces, Universal Access in Human-Computer Interaction. Ambient Interaction, Part II, HCII 2007, pp. 103-112.

Klusch, M. (2005). Retrieved August 24, 2009, from OWL-MX Matcher. http://projects.semwebcentral.org/frs/?group_id=90.

Li, L. and Horrocks, I. (2003), A Software Framework For Matchmaking Based on Semantic Web Technology, 12th Intl. Conference on World Wide Web.

Martin, D. et al. (2004), Bringing Semantics to Web Services: The OWL-S Approach, Proc. 1st Int’l Workshop Semantic Web Services and Web Process Composition (SWSWPC 04).

Michael C. Jaeger and Stefan Tang, (2004), Ranked Matching for Service Descriptions using DAML-S. CAiSE’04 Workshops, Riga, Latvia.

OSGi (2007). OSGi Service Platform Release 4, Retrieved on April, 2010 from http://www.osgi.org/Release4/Download

OWL, OWL Web Ontology Language Overview, Retrieved on August, 2010 from http://www.w3.org/TR/owl-features/

OWLJessKB, OWLJessKB: A Semantic Web Reasoning Tool, Retrieved on August, 2010 from http://edge.cs.drexel.edu/assemblies/software/owljesskb/

OWL-S, OWL-S: Semantic Markup for Web Services, Retrieved on August, 2010 from http://www.w3.org/Submission/OWL-S/

Paolucci, M., Kawamura, T., Payne, T., and Sycara, K. (2002), Semantic Matching of Web Service Capabilities. In Proceedings of the First International Semantic Web Conference on The Semantic Web.

Pellet, Pellet: OWL 2 Reasoner for Java, Retrieved on August, 2010 from http://clarkparsia.com/pellet/

Pokraev, S., Koolwaaij, J., Wibbels, W. (2003), Extending UDDI with Context Aware Features based on Semantic Service Descriptions. In Proceedings of the 2003 International Conference on Web Services (ICWS'03), Las Vegas, USA.

RACER, Renamed Abox and Concept Expression Reasoner, Retrieved on August, 2010 from http://www.sts.tu-harburg.de/~r.f.moeller/racer/

Soar, J., Swindell, R., Tsang,P. (2011), Intelligent Technologies for Bridging the Grey Digital Divide, IGI Global.

Srinivasan, N. OWL-S UDDI Matchmaker (2004). Retrieved August 24, 2009, from http://projects.semwebcentral.org/projects/owl-s-uddi-mm.

Srinivasan, N., Paolucci, M., Sycara, K. (2005), Adding OWL-S to UDDI, Implementation and Throughput. In: Cardoso, J., Sheth, A.P. (eds.) SWSWPC 2004. LNCS, vol. 3387, Springer, Heidelberg.

Sun, H., De Florio, V., Gui, N. and Blondia, C. (2007). Participant: A new concept for optimally assisting the elder people. In proceedings of the 20th IEEE International Symposium on Computer-Based Medical Systems (CBMS-2007), Maribor, Slovenia.

Sun, H., De Florio, V., Gui, N., & Blondia, C. (2007c). Service matching in online community for mutual assisted living. In Proceedings of The Third International Conference on Signal-Image Technology & Internet Based Systems (SITIS' 2007). IEEE Computer Society. Shanghai, China.

Sun, H., De Florio, V., Gui, N., & Blondia, C. (2008.b). Towards longer, better, and more active lives : Building mutual assisted living community for elder people. In the Proceedings of the 47th European FITCE Congress, FITCE, London.

Tang, S., and Liebetruth, C. (2006). The TUB OWL-S Matcher. Retrieved August 24, 2009 from http://owlsm.projects.semwebcentral.org.

UDDI Spec TC. UDDI Version 3. Technical report, OASIS, http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm, retrieved on April, 2010.

WS-HumanTask (2007). Specification: Web Services for Human Task (WS-HumanTask), version 1.0. Retrieved August 24, 2009 from https://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/a0c9ce4c-ee02-2a10-4b96-cb205464aa02.

CHAPTER 6. Conclusion and Future Work

This thesis discussed issues of service adaptation for networked devices and its application in Ambient Assisted Living environment. Digital mobile devices are widely applied nowadays, which are often connected together as a digital network. Meanwhile, the users of these mobile devices are also interconnected, constituting different social networks. Studies on service systems, which are about dynamic configuration of people, technology and shared information, exist everywhere in our daily lives. Such studies aim to deliver better services to the end users by developing new structures for service delivery. This thesis introduced one of such efforts in the Ambient Assisted Living Domain. It introduces the emerging service science and service system engineering in Chapter 2; the mutual assistance community – a service system for ambient assisted living to help the elderly people independently living is introduced in Chapter 3. Chapter 4 and Chapter 5 introduce respectively service adaptations for the mobile application and the service orchestration in the mutual assistance community.

This thesis presented much preliminary research towards building up a mutual assistance community for the ever increasing elderly population. The demographic change is happening within Europe and elsewhere, which poses great challenges to the governments in economical and social concerns. Much effort is carried out to tackle with these challenges. As an example, in Europe, the Ambient Assisted Living (AAL) Joint Program was launched to provide innovative ICT solutions to help the elderly people living independently.

Much research is being carried out using assistive devices to build safety environment for assisting the elderly people. However, the importance of social interactions is not attracting much attention until very lately. The concept of mutual assistance community presented in this thesis uses the concept of service system to orchestrate the resources and services inside a community. In such a community, social interactions between the assisted people and the outside world are much emphasized and play a central role. The elderly people are not simply treated as persons who are passively receiving help, rather than that, they are also able to provide help to the younger generation, e.g. with their knowledge. Such possibilities could help the elderly people living in active way. Meanwhile, their valuable knowledge is still an asset of the society.

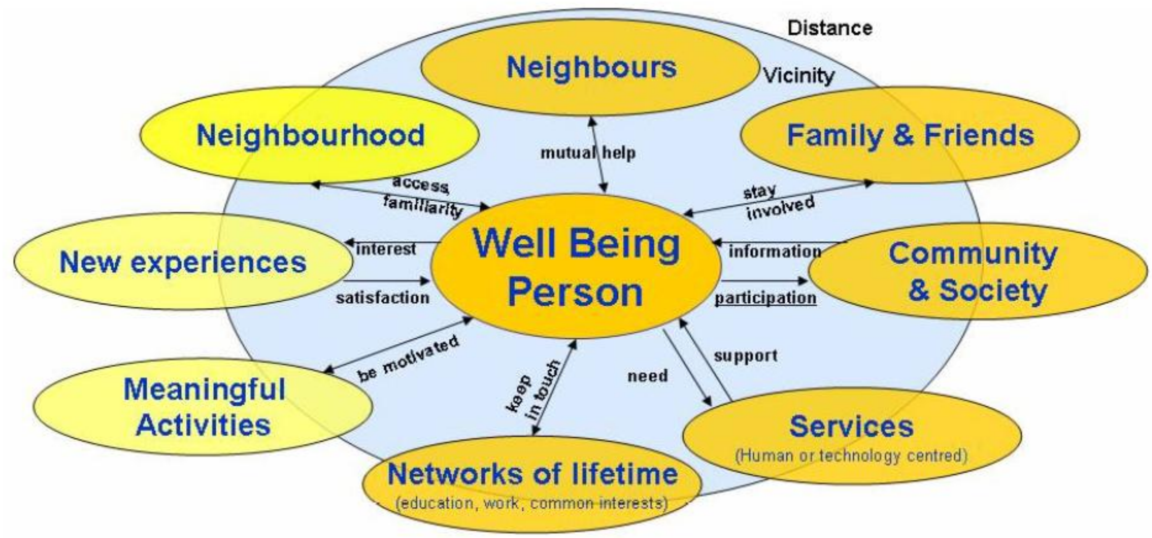


Figure 6.1 The Social Interaction Context of the elderly [AAL, 2009]

The concept of improving social interaction of the elderly people presented in this thesis, which brought out since 2006 [Sun, 2006], was coincidently highlighted in the Ambient Assisted Living Joint Programme Call for Proposals 2009 [AAL, 2009]. The proposal focused with the topic of “ICT based solutions for Advancement of Social Interaction of Elderly People”. It considered that the elderly people want to live independently and meaningfully, meanwhile also respected them for their valuable experience. It was stated that “Elderly people want to perform meaningful activities, and many older people have valuable resources, and can significantly contribute to society. It was also suggested “the networks of a lifetime can be maintained or reestablshed through involvement in common interest groups, in conjunction with the use of technical services”.

The experiments carried out in this thesis are my efforts towards the implementation of the mutual assistance community. Implementing the proposed mutual assistance community in real-life is a huge task which requires contributions from many related parties, e.g. user community, hospital, technical providers. The mutual assistance community proposed in this thesis is not yet implemented in real-life due to the restrictions of resources and certain constraints in technological aspects. Nevertheless, the preliminary simulation results validate its main concept and demonstrate the efficiency of such a community. Preliminary scenario based on web service matching also shows that such a community is able to organize group activities and effectively save the social resources. According to the experiments carried out in this thesis, a context aware adaptation framework is mature to orchestrate the assistive devices to construct a smart environment, while the semantic matching service currently demonstrates a possibility, but still requires more work before its application in a real-life community.

The contribution of this thesis is not focused on implementing an AAL system, rather, it is to raise the awareness of our society on the demographical challenges we are facing, propose innovative solutions to tackle this challenge, take preliminary validation of the proposed solution, and call for related parties to work together to meet the challenges.

Future Work

Implementing the proposed mutual assistance community in real-life requires close collaboration between all involved parties. Figure 6.2 portrays the technologies I envisaged to be necessary to build mutual assistance community or other effective Ambient Assisted Living systems.

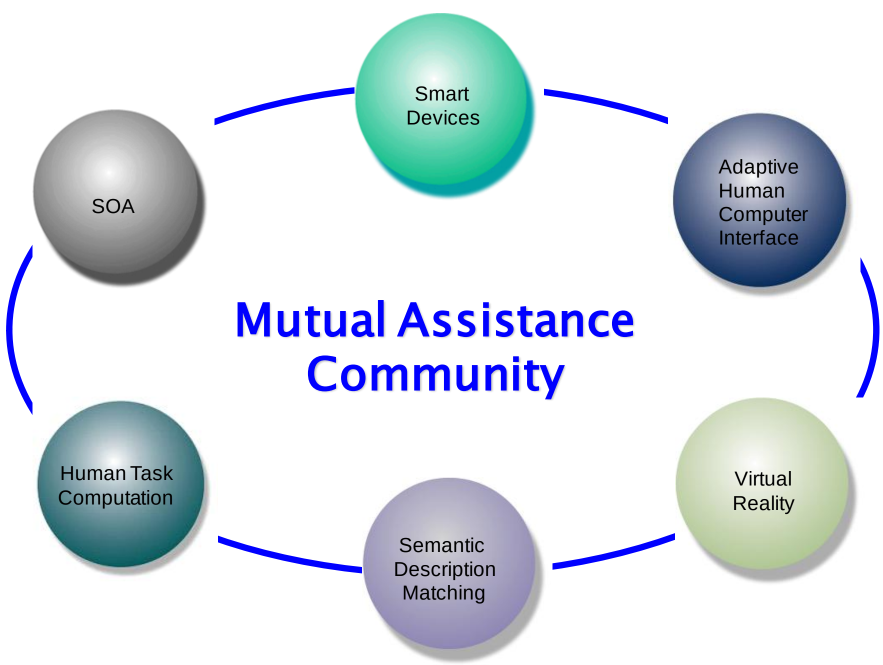


Figure 6.2 Technologies to Construct Mutual Assistance Community

Smart assistive devices are still important and are required to work as the essential blocks to build up safety environment around the assisted people. In order to best play their functions, at least some of those devices should be connected to the web allowing the easy transmission of data. Such requirements could be met by manipulating the devices with service oriented computation. As presented in Chapter 4, coupling with the Aspect Oriented Programming (AOP), the Service Oriented Architecture (SOA) system can flexibly organize the devices, monitor the changes and take corresponding adaptations.

Service Oriented Architecture is the approach to orchestrate the services inside the mutual assistance community; it is not only used to organize the assistive devices, but also for representing services from the human side. Specifications of human task computation (e.g. BPEL4People) could help to represent human services in SOA, and finally integrate them with services from device side.

Semantic description and matching technologies are crucial in the service orchestration of the mutual assistance community. In Chapter 5 of this thesis, an example is presented in matching between a web service requester and a web service provider. However, such implementation still has a long way to go before it could effectively play its role in a living mutual assistance community. Constraints of service description and matching mainly come from two aspects:

Firstly, certain constraints exist in representing services or requests in the mutual assistance community. Common ontology libraries for the AAL domain should be developed by the related parties, so that different parties could define their services/requests using same ontology. Ambient assisted living is a big domain; it is difficult to define a single common ontology library including all the needed ontology. This is because such a library would be extremely huge, thus it would be difficult to get it standardized and be used by different projects. Meanwhile, developing a huge monolithic ontology library will also be hard to maintain and extend. A feasible solution is to break down the AAL domain into a set of sub-domains, which allows users to flexibly deploy their requested libraries and reuse the existing ontology.

Secondly, the efficiency of the semantic service matching tools require further improvement to enable such tools to reason upon large amount of advertised and requested services in a short time. The existing matching tools take rather long time for the process of service matching. In the application of community level, the service matching center needs to process a large quantity of service matching: not only need to process large quantity of services, but also need to compare each request to a large amount of available services. In this thesis, experiments have been carried out using OWL-S to represent human services and use OWL-S Matcher to match the required and advertised service. Such a method is not scalable in execution speed: a real life mutual assistance community would be comprised of tens of thousands of services, loading these services one by one and compare them to the required service would take extremely long time. An option would be building up specific ontologies to describe the required and advertised services, store the related information (availability or request) in triple store or even relational database. SPARQL [SPARQL, 2008] query could be executed on the triple store to select a limited set of candidates. Complex semantic rules could thereafter be applied on the selected candidates to generate the final results. With such an approach, the time consumption could be significantly reduced.

Some other technologies such as virtual reality and adaptive human computer interface are necessary in building up a user friendly interface for the elderly to access such a system. In [Sun, 2008a], possibilities of implementing the mutual assistance community in the form of virtual community and create inter-reality communication are discussed. Other studies of investigating and increasing the user's acceptance of such a community should also be carried out.

In conclusion, the scale of adaptation systems are expanding, orchestrating system resources as services could help to efficiently utilize these resources. The mutual assistance community presented in this thesis shows a vision of service orchestration at the community level for the elderly people. Constructing the Ambient Assisted Living System with the perspective of community level could help to greatly save the social resources, at the same time, also helping the elderly people live in an active way. Although the proposed mutual assistance community in this thesis is not implemented as real life application, some original concepts proposed in this thesis are now appearing in other AAL projects, and more concepts in the proposed mutual assistance community would be implemented in near future.

References

AAL (2009), Ambient Assisted Living (AAL) Joint Programme, Call for Proposals AAL-2009-2, ICT based solutions for Advancement of Social Interaction of Elderly People, retrieved in December, 2010 from http://www.aal-europe.eu/aal-2009-2

Prud'hommeaux, E. and Seaborne, A. (SPARQL 2008), A. SPARQL Query Language for RDF. W3C Recommendation. 15 January 2008. http://www.w3.org/TR/rdf-sparql-query/.

Sun, H., De Florio, V. and Blondia, C. (2006). A Design Tool to Reason about Ambient Assisted Living Systems, in Proceedings of the International Conference on Intelligent Systems Design and Applications (ISDA 06), IEEE Computer Society.

Sun, H., De Florio, V., Gui, N. and Blondia, C. (2008.a). Towards Building Virtual Community for Ambient Assisted Living, in the Proceedings of 16th Euromicro International Conference on Parallel, Distributed and network-based Processing (PDP2008). IEEE Computer Society.