ESPOO 2004 VTT PUBLICATIONS 534

Päivi Kallio

Emergence of Wireless Services

Business Actors and their Roles in Networked Component-based Development

VTT PUBLICATIONS 534

Emergence of Wireless Services
Business Actors and their Roles in Networked Component-based Development

Päivi Kallio

VTT Electronics

Academic Dissertation to be presented, with the assent of the Faculty of Science, University of Oulu, for public discussion in Raahensali (Auditorium L10), Linnanmaa, on August 5th, 2004, at 12 o'clock noon.

ISBN 951–38–6386–7 (soft back ed.)
ISSN 1235–0621 (soft back ed.)

ISBN 951–38–6387–5 (URL: http://www.inf.vtt.fi/pdf/)
ISSN 1455–0849 (URL: http://www.inf.vtt.fi/pdf/)

JULKAISIJA – UTGIVARE – PUBLISHER

VTT, Vuorimiehentie 5, PL 2000, 02044 VTT
puh. vaihde (09) 4561, faksi (09) 456 4374

VTT, Bergsmansvägen 5, PB 2000, 02044 VTT
tel. växel (09) 4561, fax (09) 456 4374

VTT Technical Research Centre of Finland, Vuorimiehentie 5, P.O.Box 2000, FIN–02044 VTT, Finland
phone internat. + 358 9 4561, fax + 358 9 456 4374

VTT Elektroniikka, Sulautetut ohjelmistot, Kaitoväylä 1, PL 1100, 90571 OULU
puh. vaihde (08) 551 2111, faksi (08) 551 2320

VTT Elektronik, Inbyggd programvara, Kaitoväylä 1, PB 1100, 90571 ULEÅBORG
tel. växel (08) 551 2111, fax (08) 551 2320

VTT Electronics, Embedded Software, Kaitoväylä 1, P.O.Box 1100, FIN–90571 OULU, Finland
phone internat. + 358 8 551 2111, fax + 358 8 551 2320

Technical editing Leena Ukskoski

Otamedia Oy, Espoo 2004

Kallio, Päivi. Emergence of Wireless Services. Business Actors and their Roles in Networked Component-based Development. Espoo 2004. VTT Publications 534. 118 p. + app. 83 p.

Keywords wireless service development, business actors, business roles, wireless business networks, business models, development, software

Abstract

Mobility of wireless services allows their use anywhere and anytime via portable terminals. 3G (Third Generation) technologies with increased transmission speeds are expected to increase the revenue from wireless services significantly for all actors involved in business, especially as they will enable new kinds of services based on versatile business models. A business model is architecture for the product-, service- and information flows, including a description of the various business actors and their roles, the potential benefit for the actors and the sources of revenue.

Success in the wireless business requires combining an appropriate business model with an attractive terminal and content, and offering it to the user as an appealing service package. Companies have to select their role and partners in the business network carefully by taking into account the diversity of terminals, networks and software. A large variety of technologies combined with the need to save costs and a faster time-to-market leads towards open interfaces and standards to allow interoperability between different operators and networks, and a predominately component-based approach to service development.

The aim of this thesis is to analyse how wireless services can be developed more efficiently in the context of 3G networks by focusing on the actors' roles and component-based development. The business models and roles are illustrated in this research with the aid of two types of wireless services that are under development for 3G networks. Development of the two case services – a multi-player game and a trading on-line (TOL) service – in this research proves the suitability of component-based development to the development of wireless services. To achieve the best results one has to take into account the risks of commercial component use and the initial investments component-based development requires. This research also shows that component documentation helps remarkably in the former and is amendable to decrease the latter by providing information about components' use in service development.

Preface

The major part of this research was carried out between 2001 and 2003 at VTT Electronics while I was working in EU/IST-project WISE funded by the European Commission and the participating five European companies and three research institutes including VTT Electronics. In completing this research, I would like to thank my supervisor, Professor Veikko Seppänen, for his great support and advice. Our co-operation started as I was choosing the topic of my Master's thesis and Veikko has effectively led me towards the highest academic degree. The path has not always been easy, but I was lucky to get one of the best supervisors – a person who has always understood my difficulty in finding my way between the economics and information processing sciences.

I also would like to thank Professors Tuija Kuusisto and Jari Veijalainen for their useful comments and proposals for improving this thesis. I also would like to thank my collegues, Professor Eila Niemelä and Dr. Pekka Abrahamsson, for leading me into scientific thinking, and all the co-authors of my papers for being involved in this.

I also would like to thank TietoEnator's fund of Suomen Kulttuurirahasto for the financial support for my dissertation work.

I would also like to thank my parents for their patience and love throughout my life, and my brother for his refreshing ideas. And Jaana, Piia and Susanna – thanks for dragging me away from the computer into the real world. Warm thanks also to Seri for your patient waiting at weekends when I was writing this thesis.

And last, but not least – Michel, warm thanks for your love and support and giving meaning to my life beyond the academic arena.

In Oulu, 22.5.2004

Päivi Kallio

Contents

List of original publications

This thesis includes the following eight original publications:

I. Kallio, P. 2001. Application Service Provisioning – Current State and Partnership strategies. GI-Edition Lecture Notes in Informatics: Information Systems Technology and its Applications: International Conference ISTA 2001, Kharkiv, Ukraine, June 2001. Bonn: Gesellschaft für Informatik. Pp. 155–172. ISBN 3-88579-331-8 ISSN 1617-5468.

II. Kallio, P. 2002. Business Models in Wireless Internet Service Engineering. Proceedings of the 3rd ICSE Workshop on Web Engineering, Orlando, May 2002. Pp. 57–64.

III. Kallio, P., Niemelä, E., 2001. Documented Quality of COTS and OCM Components. Proceedings of the 4th ICSE Workshop on Component-Based Software Engineering: Component Certification and System Prediction. Toronto, May 2001. Pp. 111–114.

IV. Taulavuori, A., Niemelä, E., Kallio, P. 2004. Component documentation – a key issue in software product lines. Information and Software Technology. Volume 46, Issue 8, pp. 535–546.

V. Kallio, P., Ihme, T. 2002. Evolution of the Use and Risks of the Commercial Software Components. Proceedings of the 28th Euromicro Conference, Dortmund, September 2002. Los Alamitos: IEEE Computer Society. Pp. 55–61. ISBN 0-7695-1787-0 ISSN 1089-6503.

VI. Kallio, P., Matilainen, A., Boggio, D., De Matteis, G. 2003. Wireless Internet Service development. Proceedings of the 2nd International Workshop on Wireless Information Systems, WIS 2003 in conjunction with ICEIS 2003, Angers, France, April 2003. Setubal: ICEIS Press 2003. Pp. 31–42. ISBN 972-98816-5-0.

VII. Kallio, P., Cortese, G., Tiella, R., Zorer, A. 2002. Accounting and Billing of Wireless Internet Services in the Third Generation Networks. In: Advanced Conceptual Modeling Techniques: ER 2002 Workshops, ECDM, MobIMod, IWCMQ, and eCOMO, Tampere, Finland, October 7–11, 2002, Revised Papers. Lecture Notes in Computer Science Volume 2784 / 2003. Heidelberg: Springer-Verlag. Pp. 229–240. ISBN 3-540-20255-2 ISSN 0302-9743.

VIII. Kallio, P., Holappa, J., Ivanchev, P. 2002. Integrating VPN and OSGi to create a secured Virtual Home Environment. Proceedings of the International ITEA Workshop on Virtual Home Environments, Paderborn, February 2002. Aachen: Shaker Verlag. Pp. 121–130. ISBN 3-8265-9884-9 ISSN 1438-3527.

The author of this thesis is the principal author of the Papers I–III and V–VIII. The research ideas of papers III and V–VIII have been presented by the author of this thesis and the author has carried out most of the practical work of these papers. For Paper III, Prof. Eila Niemelä has provided her expertise in component-based development and for Paper V, Mr. Tuomas Ihme his expertise in use of commercial components. For Paper V the author of this thesis has evaluated the three case-example projects. In Paper VI, the other authors have given comments and input from the viewpoint of the companies they present and the author of this thesis has combined these ideas using her business expertise. For Paper VII, the other authors have provided input and comments about architecting accounting and billing applications and the author of this thesis has combined this input with business aspects. For Paper VIII, the other authors have provided their expertise about security aspects of virtual home environments (VHEs). The research work of paper VIII has been managed by the author of this thesis.

In Paper IV, the principal author is Ms. Anne Taulavuori. However, the author of this thesis has provided new ideas and her expertise in component documentation for the paper. Furthermore the author of this thesis has significantly contributed to the papers in terms of planning, managing the writing process as well as co-writing the papers.

List of abbreviations

2G, 2.5G, 3G	Second Generation, 2.5th Generation, Third Generation
3GPP	Third Generation Partnership Project
AP	Application Provider
API	Application Programming Interface
ASP	Application Service Provider
BOBO	Billing On Behalf Of
CBSE	Component-Based Software Engineering
CDMA	Code Division Multiple Access
cHTML	compact Hyper Text Markup Language
CN	Core Network
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
COTS	Commercial Off-The-Shelf
CP	Content Provider
DB	DataBase
EDGE	Enhanced Data Rates for Global Evolution
ESA	European Space Agency
ETSI	European Telecommunications Standard Institute
FTP	File Transfer Protocol
GGSN	Gateway GPRS Support Node
GPS	Global Positioning System
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communication
HTTP	Hyper Text Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ISP	Internet Service Provider
J2EE	Java 2 Platform, Enterprise Edition
J2ME	Java 2 Platform, Micro Edition
JMS	Java Message Service
JVM	Java Virtual Machine
kbps	Kilobytes Per Second
LAN	Local Area Network

MDA	Model Driven Architecture
MFC	Microsoft Foundation Classes
MIDP	Mobile Information Device Profile
MMS	Multimedia Messaging Service
NASA	National Aeronautics and Space Administration
NO	Network Operator
OCM	Original software Component Manufacturing
OMA	Open Mobile Alliance
OMG	Object Management Group
OMT	Object Modelling Technique
ORB	Object Request Broker
OS	Operating System
OSA	Open Service Access, formerly Open Service Architecture
OSGi	Open Service Gateway initiative
PC	Personal Computer
PDA	Personal Digital Assistant
PDC	Personal Digital Cellular
PPP	Point to Point Protocol
PSTN	Public Switched Telephone Network
RAN	Radio Access Network
RMI	Remote Method Invocation
SLA	Service-Level Agreement
SMS	Short Messaging Service
SP	Service Provider
TCP/IP	Transmission Control Protocol/Internet Protocol
TDMA	Time Division Multiple Access
TOL	Trading On-Line
UI	User Interface
UML	Unified Modelling Language
UMTS	Universal Mobile Telecommunications System
USA	United States of America
UTRA-TDD	UMTS Terrestrial Radio Access Time Division Duplex
VHE	Virtual Home Environment
VPN	Virtual Private Network
WAN	Wide Area Network
WAP	Wireless Application Protocol

W-CDMA	Wideband Code Division Multiple Access
WISE	Wireless Internet Service Engineering
WLAN	Wireless Local Area Network
XML	eXtensible Markup Language

1. Introduction

This chapter explains the background to the research and the research problem and method, and gives an outline of the dissertation.

1.1 Background

The success of the Internet has had a dramatic impact on software development as it has enabled the interaction of applications over the network to form end-to-end services (Messerschmitt & Szyperski 2001). For business actors in the wireless field, the 3G (Third Generation) networks create new revenue opportunities by enabling fast and efficient delivery of personalised end-user services, i.e. services delivered to end-users via wireless terminals – i.e. physical wireless devices. An end-user is a company or individual using the service. The UMTS (Universal Mobile Telecommunications System) Forum (2003) forecasts that, for example, mobile operators will earn more than €1 trillion in customer revenues by 2010. In developing wireless services, Japan has, in recent years, taken the lead over Europe, and, lately, South Korea is even overtaking Japan (Henten et al. 2003). Taking advantage of the revenue potential of wireless services requires taking into account many socio-cultural, technological, economical and regulatory factors, and creating an appropriate business model, as the success of a service greatly depends on the way they are introduced to the users. (Funk 2004, Sharma & Nakamura 2003)

Wireless service is a set of various service functions provided as an end-to-end service over a network to an end user's terminal when requested. A wireless *application* is application-level software that operates on a host server and/or on a wireless terminal (Heyes 2003). Examples of wireless services are ringing tones, messaging services, mobile marketing, navigation services, payment services, etc. (Funk 2004). Wireless technology allows information to be transmitted between devices without use of physical connections (Heyes 2003). Mobility is a unique function of a wireless system and means the end-users' ability to move freely from one place to another (Sharma & Nakamura 2003). All wireless services are not mobile, although these concepts are often overlapping. In this research, wireless services mean mobile wireless services.

Business models have been defined in the literature in numerous ways (cf. Cartwright & Oliver 2000, Hamel 2000, Mahadevan 2000, Mintzberg & Quinn 1996, Tapscott et al. 2000). This research uses Timmer's definition of a business model that is "architecture for product, service and information flows, including a description of the various business actors and their roles, and a description of the potential benefits for various business actors, and a description of the sources of revenue”. Business actor is a company or individual acting in a network in one or several business roles. Business role is the role a company plays in a value chain – that is, a series of different functions a company performs in order to deliver a service (Afuah & Tucci 2001).

In business networks the business actors perform activities and create value through transforming resources such as know-how or capital (Anderson & Narus 1999). In networked service development the service is developed by a companies' network that is a long-term, purposeful arrangement among distinct companies that allows them to gain competitive advantage over their competitors outside the network (Jarillo 1998). The enablers of stronger co-operation between different business actors are uniform standards (cf e.g. OMA 2002), focusing on customer requirements, and interoperable terminal offering and service platforms.

The need to cut costs and reduce time-to-market has substantially increased business actors’ interest in the use of commercial components provided by third parties – i.e. other business actors. In this research, component means a software component that is a unit of composition with contractually-specified interfaces and explicit context dependecies (Szyperski 1998). Software is basically anything that can be stored electronically, i.e.computer instructions or data (Internet.com 2002). The use of commercial components is not entirely problem-free; the blackbox character of the components causes, for example, interoperability problems to users of commercial components.

The following section handles aspects that have to be taken into account when developing wireless services: the capabilities and needs of wireless teminals, networks and software consisting of components, and the interfaces between them (Heyes 2003).

1.2 Aspects influencing wireless service development

Wireless service development influences the terminal, networks and software that locate either in the terminal or on a server.

The wireless *terminals* that differ in features like shape, form, memory, display size and processing power can be divided into mobile phones, communicators/smartphones (including multi-purpose terminals), laptops and notebooks (Heyes 2003, Sharma & Nakamura 2003). The case studies in this research use mobile phones and communicators/smartphones as terminals.

The main components of a wireless *network* are a core network (CN) and a radio access network (RAN). The CN is the basic platform for all the communication services provided to the subscribers and it enables the handling of different kinds of radio accesses. Due to the traditional infrastructure basis, the CN has not evolved as a radio network in time (Kaaranen et al. 2001). In this research, wireless network means commercial, digital cellular networks; other wireless networks, like satellite, Bluetooth and WLAN (Wireless Local Area Network) networks, are beyond the scope of this research. Digital cellular networks include 2G (2nd Generation), 2.5G (2.5th Generation) and 3G networks.

Figure 1 presents the transmitting of service requests and the replies to them over the network, which can, in addition to CN and RAN, include connections to external networks like the Internet and PSTN (Public Switched Telephone Network). Usually, the servers are connected to private or public IP (Internet Protocol) -networks. The arrows in Figure 1 indicate the requests and replies between terminal and server.

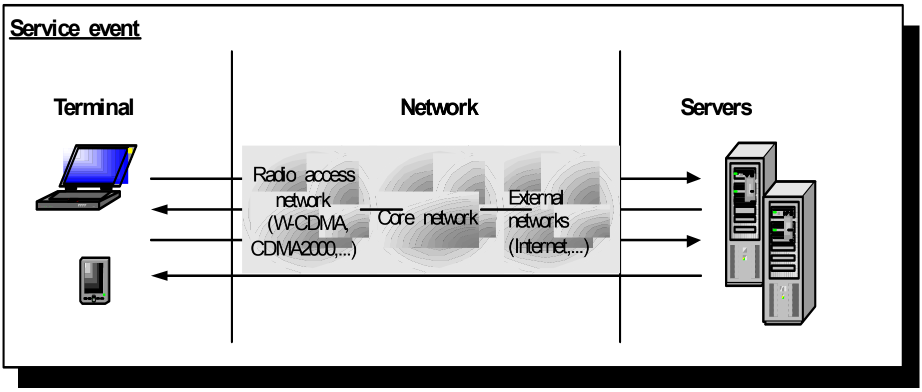


Figure 1. Concept of service event (modified from García et al. 2002).

Software comprises the application, middleware and system-levels as Figure 2 shows. Middleware is "software that insulates applications from the underlying wireless network, making it easier to develop new wireless applications" and port applications to the wireless environment (Sharma & Nakamura 2003). System-level components – i.e. components at the operating system (OS) level – are not within the scope of this research.

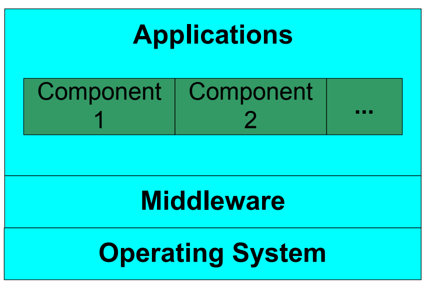


Figure 2. Structure of software.

Software architecture that is high-level design of the sofware is “the structure or structures of the system, comprising software components, the externally visible properties of those components and the relationships among them” (Bass et al. 1998). Software components can locate in the terminal, on the network (protocol components) and on the server, where the service is run (ref. Figure 1).

1.3 Research idea

The aim of this thesis is to enlighten the emergence of wireless services, focusing on the business models and roles of the business actors in component-based development of wireless services. As Figure 3 depicts, the life cycle of a service consists of four phases: service development, deployment, usage/ provisioning and maintenance. Figure 3 illustrates the business roles and their connection to component-based development in each phase of the life cycle. The business roles are explained in more detail in Chapter 3. As the arrows in Figure 3 indicate, components and their functions are provided by application providers (APs) that develop vertical and horizontal applications and deployed and used by other business actors in later phases of the service life cycle. Providing content that is the information the end-user wants and is willing to pay for, and the quality of the delivery is what has been agreed (UMTS 2000), is outside the scope of this research.

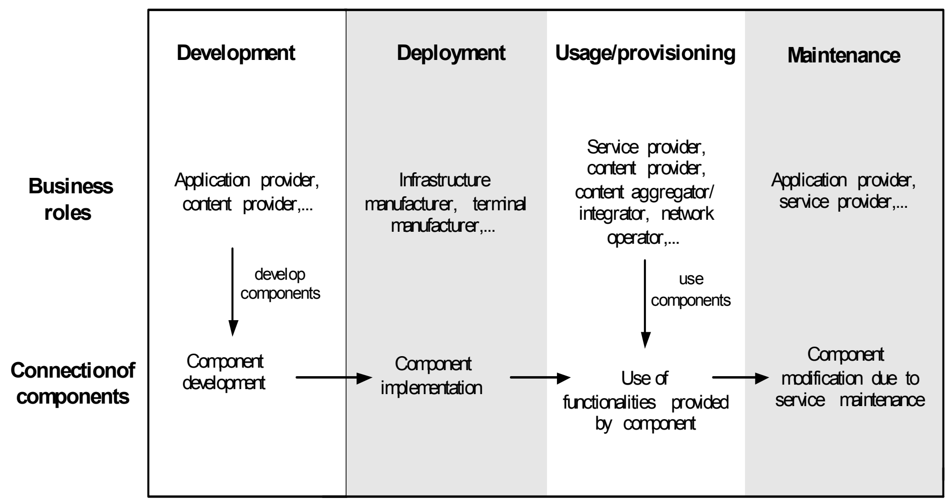


Figure 3. Service life cycle.

The service development phase includes requirements specification, architectural design, implementation (coding) and testing (Adamopoulos et al. 2000, Jaring et al. 2004); in this phase business actors earn revenue by selling application/ content to other business actors. The service deployment phase includes installation and activation of the service (Adamopoulos et al. 2000); in this phase business actors earn revenue by implementing the service and needed infrastructure and terminals. The usage/provisioning phase includes provisioning of the service and its use by end users; in this phase business actors earn revenue by providing the service to the end user. The maintenance phase includes all the activities required to evolve a service over its lifetime; in this phase business actors earn revenue from maintenance of the service. In this thesis service development includes the development and deployment phases of the service life cycle and those aspects of the usage/provisioning and maintenance phases that affect service development.

1.4 Research problem

The aim of this thesis is to investigate how wireless services can be developed more efficiently in the context of 3G networks by looking for answers to the following problems:

1. What is the status of the wireless service industry today?
2. What kinds of roles and transactions emerge in networked wireless business models?
3. What are the risks involved in the use of commercial components in wireless service development?
4. How can component documentation help the development of wireless services?
5. What has to be taken into account in component-based development of wireless services?

And as summary:

6. How can wireless services be developed more efficiently within the context of 3G networks?

This research first tries to put together an overall picture of the wireless industry from the technological and business points of view. As the interaction of the wireless business actors defines their responsibilities in component-based service development, this research then analyses wireless business actors and their roles and interaction in wireless business networks. Thirdly, this research evaluates the potential risks involved in component-based development and evolution of commercial component use. Fourthly, this research shows how qualitative documentation of components helps wireless service development. Finally, this research evaluates the usability of component-based development in wireless service development and development of wireless services within 3G networks in general. Chapter 6 of this thesis covers the more widespread effect of the findings in this research.

1.5 Research method

This research used several research methods. Table 1 presents the utlilised research approaches based on the classification of Järvinen (2001).

Table 1. Utilised research approaches.

Article	Research approach	Objective	Research method
I	Theory-creating	Analyse the state of Application Service Provisioning business in Finnish companies.	Case study research (four cases)
II	Theory-creating	Develop a business model for the future wireless Internet services.	Action research – case study research (three cases)
III, IV	Constructive	Develop a standard documentation template for commercial software components.	Action research
V	Theory-testing	Test the hypothesis that significance of commercial software components in architectural design has increased and evaluate risks encountered in component-based development.	Case study research (three cases)
VI, VII, VIII	Theory-creating	Validate suitability of component-based development to development of wireless services.	Action research – case study research (three cases)
Thesis summary	Theory-creating	Enlighten the emergence of wireless services and their component-based development.	Action research – case study research, cross-case analysis

Most of the publications in this research belong to the theory-creating research approach that aims at creating new theories that would best explain and describe reality. Theory-creating research is best suited to situations in which there is no prior knowledge of a phenomenon or part of reality (Järvinen 2001). Theory-creating research includes grounded theory, case study, phenomenography, contextualism and ethnographic approaches. Cunningham (1997) divides the case study research method used in Articles I, II and V–VIII into three classes: intesive, comparative and action research. Articles I and V used comparative case study, where theory is developed by comparing cases (Cunningham 1997). The cases of Article I were selected according to their expertise in the research area. Action research is a qualitative research approach that aims to improve practice through the collaborative work of researchers and practioners (Avison 2002). Process of action research includes: identifying the problem, action planning and taking, evaluating and specifying learning (Susman & Evered 1978). Action research is suitable for increasing the understanding of change processes in systems (Hult & Lennung 1980). In Articles II, VI and VII practioners participated in the theory-creating process by giving feedback about the requirements for the services to be developed and about received experiences in the forms of interviews. For Article II, the practioners were interviewed before defining the model, and they checked the correctness of the model afterwards. In Articles VI and VII the practioners participated in the whole research process.

The constructive research approach used in Papers III and IV builds a new innovation – construct, model or method – based on existing knowledge and/or new technical advancements (Järvinen 2001). The generalizations of Papers III and IV were based on comparing the research results of the related literature with the needs defined by the users of the component documentation in industrial organizations.

The theory-testing research used in Paper V is applied in empirical studies for testing if the experiments carried out confirm or falsify the theory. Theory-testing research includes controlled experiments, longitudal field studies and theory-testing case research (Järvinen 2001). Testing the hypothesis in Article V called for a timely sequence of the cases; thus the selected cases are consequent in time.

The thesis summary compares the case studies with the results of Articles I–VIII to gain a deep understanding of the state-of-the-art of wireless service industry and the suitability of component-based development to the development of wireless services. Cross-case analysis enables comparison of several cases to draw deeper conclusions about them. Eisenhardt (1989) suggests three tactics for making cross-case analysis: 1) select categories and look for within-group similarities, 2) select pairs of cases and list the similarities and differences between pairs, and 3) divide data by data source to "exploit unique insights from different types of data collection". From the above-mentioned alternatives, this thesis used alternatives 1) and 3) for making the cross-case analysis in Chapters 4 and 6.

1.6 Outline of the dissertation

Figure 4 shows the overall structure of the research described in this thesis.

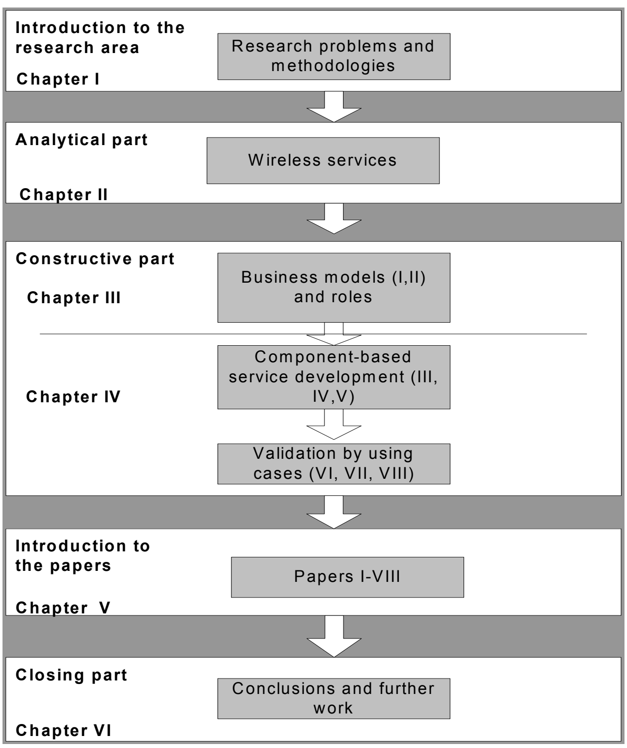


Figure 4. The overall structure of the research.

As Figure 4 shows, this research is divided into introduction to the research area and papers, and analytical, constructive and closing parts. The constructive part of this research is based on the information content and analysis of Papers I–VIII.

This thesis consists of six chapters, as follows:

Chapter 2 of this dissertation presents the different categories of wireless services. The factors influencing wireless service development are also handled in this chapter.

Chapter 3 handles business models, actors and roles in wireless businesses. This chapter also presents a business model framework for two wireless case services that are under development for 3G networks.

Chapter 4 handles component-based development of wireless services and its evolution and risks. This chapter also presents a documentation template for software components and illustrates the component-based development of two wireless case services.

An introduction to the original Papers I to VIII is given in Chapter 5.

Chapter 6 draws the conclusions of the thesis and indicates some ideas for further research.

Papers I to VIII are included as appendices.

2. Wireless services

The power relationships of the wireless business actors have evolved a lot in recent years. In Japan and South Korea introduction of mobile services has been led by network operators (NOs) that sell network capacity of wireless networks to consumers, but in Europe they did not manage to create such markets as in Japan and South Korea. The reasons for the slow introduction of wireless services based on 3G platforms in Europe have been the financial crisis in telecom sector, the high prices of frequency bands and the lack of an appropriate business model (Henten et al. 2003). In spite of the European delay in service implementation, the belief in wireless business is still strong in all continents.

According to the forecast of UMTS Forum (2003), the biggest growth in the wireless service arena will occur in rich-voice, customised infotainment and mobile intranet/extranet access. Young people are often the initial users of wireless services and simple entertainment like ringing tones and screen savers are the largest content markets (Funk 2004). When developing wireless services, business actors should adapt the best practices of wired services, but also consider that providing a service as bit-pipe – as in the Internet – to customers does not work in the wireless context (Henten et al. 2003).

The I-Mode service, which can be considered as a successful example of wireless service introduction, was launched in Japan in 1999 and provides continous Internet access, email and other services based on packet switched technology. I-Mode is built around four components: mobile phone, mobile network, I-Mode server and information providers. I-Mode uses open standards throughout the system and is based on cHTML (compact HTML), which is more suitable for small displays and makes pages faster to download (Henten et al. 2003). On DoCoMo's I-Mode service in 2002, the service access percentage by content category was the following: 37% ringing tones and screen savers, 20 % games and horoscopes and 20 % entertainment information. (Funk 2004)

Table 2 presents some essential sources of information on the mobile and wireless arena that are used in this research. The first column of Table 2 presents the name of the source and the second column the topic handled in the source. Many of the organizations providing mobile standards and specifications are strongly represented by terminal manufacturers and telecom operators.

Table 2. Information sources of wireless area.

Source of information	**Topic**
http://www.3gpp.org/	Technical specifications and standards for 3G mobile systems
http://www.ericsson.com/technology/	Mobile phone and network technologies
http://www.etsi.org/	Telecom standards
http://www.itu.int/home/	Documents and recommendations about telecommunication
http://www.nokia.com/nokia/0,,32888,00.html	Mobile phone and network technologies
http://www.openmobilealliance.org	Architectural framework, open standards and specifications for the mobile industry
http://www.tinac.com/	Specifications for telecommunications networking infrastructure
http://www.umts-forum.org/servlet/dycon/ztumts/umts/Live/en/umts/Resources_Reports_index	Reports about 3G technologies and markets
http://www.wireless-world-research.org/	Reports and documents about future directions of wireless field
IEEE Transactions on wireless communications	State-of-the-art and applications of wireless communications
IEEE Transactions on mobile computing	Mobile computing
International journal of mobile communications	Mobile communication, technology and applications
[Baldauf et al. 2001]	Networking in telecommunications business
[Capra et al. 2002]	Middleware for mobile computing
[Durlacher 2000]	Future mobile business models and applications
[Durlacher 2001]	Future and current state of m-commerce
[Funk 2004]	Technologies and applications of mobile Internet
[Henten et al. 2003]	Mobile systems and services in Europe, Japan and South Korea
[Heyes 2003]	Basics of wireless technology
[Sharma & Nakamura 2003]	Business models of wireless data services

The information sources presented in Table 2 provide a picture of the status of the wireless industry today.

This chapter seeks an answer to research problem 1 (p. 18) by discussing the status of wireless services and factors affecting wireless service development. The factors affecting wireless service development are discussed as they have crucial effect on the decisions made prior to developing the service and the responsibilities of the business roles and actors appearing in it.

2.1 Wireless service categories

The following sections present in more detail the service categories of the case services in this research. The categorization of wireless services into end-user, service management and application support services is based on how directly the service is related to the end-user's reason for using the service. The discussion about the wireless services and their taxonomy is mainly based on Durlacher (2000, 2001), Funk (2004), Heyes (2003), Sharma & Nakamura (2003), Telemanagement Forum (2003) and UMTS (2000) due to their suitability for the scope of this research.

2.1.1 End-user services

The success of a wireless service depends on its added value to the customer, which means the increased value that the customer gains from the use of the service. Entertainment services and personalised access to information and corporate networks are expected to have the biggest profit potential in the future (UMTS 2003*)*. Categories of end-user services are, for example, mobile entertainment, mobile information, mobile banking, and mobile commerce. This section presents the end-use service categories that the case services of this research represent.

Mobile entertainment services like mobile games, screen savers, ringing tones and video have huge revenue potential in the wireless service market, particularly as the memory capacity of the terminals develops even further (Funk 2004). Higher processing and network speeds, always-on environment, gaming-centric terminals and reduced packet charges will probably increase the usage of wireless games by young people (Durlacher 2001, Funk 2004). The profitability of games depends on obtaining enough subscribers to cover the development costs (Funk 2004). The case service 2 of this research – a multi-player game – belongs in this service category.

Wireless terminals can deliver a great variety of *mobile information* like news, traffic, weather, maps, travel and timetables of aeroplanes, buses, trains, etc., to the end user. The disadvantage of those information services that provide information about where to go is that setting up a partner network for them sometimes takes a lot of time. Map services have not succeeded very well due to the small display size of the terminals and their poor resolution. The future challenge for CPs (Content Providers) that provide value added content to different business actors is to combine maps, destination information, timetables, etc. Navigation services, such as car navigation services, are based on a GPS (Global Positioning System) system that enables the positioning of the user's location based on satellites (Funk 2004). In the case service 1 of this research – an online trading system – stock exchanges provide users with information about stocks and news providers with the latest news.

Mobile banking services that enable consumers to check exchange and interest rates, manage their accounts, pay invoices and transfer funds (Durlacher 2000, 2001) are an additional distribution channel for the traditional banking services. A bank's incentive for mobile banking is its ability to cut costs by saving on salaries and real estates (Durlacher 2001), but so far the mobile banking services have not been successfull anywhere in the world as they are hard to use (Funk 2004). Mobile commerce also includes mobile marketing, which can be used to deliver various forms of sponsored advertisements. In Japan young people already use wireless terminals to redeem coupons, and receive free samples and product information (Funk 2004). In case service 1 of this research the banks act as a customer interface by providing the service to their customers.

To *communication services* belong, for example, rich call, messaging, email and VHE, which is a service concept introduced by the 3GPP (Third Generation Partnership Project) for UMTS. VHE aims at enabling its users' personal service environment that is portable across network boundaries and between terminals, and increasing the occupational productivity of mobile workers anywhere and anytime by ensuring that they are consistently presented with the same personalised features, UI (User Interface) customisation and service preferences (UMTS 2000, Yew et al. 2001). VHE is dynamic in the sense that clients can register with the network and unregister from it at any time. Case service 3 of this research uses a VPN (Virtual Private Network) and OSGi (Open Service Gateway initiative) server to create a secured VHE.

2.1.2 Application support services

Application support services are services that provide generic services for a specific application domain on which end-user applications rely, but they are not usually targeted at the end user as a service (e.g. a game engine). The application support services provide building blocks from which similar services can be easily constructed with minimal development costs (cf. OMA 2002). An end-user application might typically use the services in one or two of these categories, such as wireless gaming support, wireless multimedia support, communication support (like SMS (Short Messaging Service), MMS (Multimedia Messaging Service) and unified messaging), location-based services (like positioning) and payment (Kalaoja et al. 2003). Case services 1 and 2 of this research utilize application support services such as gaming support.

SMS has become immensely popular in countries with GSM (Global System for Mobile communications) systems and the cost of delivering it is much lower than the prices charged (Hensen et al. 2003). In the future, MMS that allows users to create unique messages by using all types of multimedia will partly replace SMS. The average size of MMS (30,000 bytes) is huge compared with SMS (140/160 bytes), but it allows the user to get the message directly into the terminal. Unified messaging enables the user to access various messages via one single interface (Sharma & Nakamura 2003, UMTS 2000).

In the United States, wireless carriers are forced to provide *location* identification as part of E911 (Sharma & Nakamura 2003). Implementation of the location identification features in the United States will be completed by the end of 2004 (Spread Spectrum Scene 2004).

A smartcard embedded in a wireless terminal can replace money as a *payment* tool as it enables payments by using the wireless terminal. This kind of payment system has been criticized owing to the chance of a government monitoring our transactions (Funk 2004).

2.1.3 Service management services

Service management services are services that are needed to make the service available and link it to the business processes. The enhanced Telecom Operations Map of the Telemanagement Forum (2003) process model includes several horizontal layers necessary for service management processes. Service management services are not normally directly related to the purpose of end-user applications as such. The service management operations that should be transparent to the wireless service and its developer include such categories as end-user fulfillment support (i.e. service provisioning), service assurance support (i.e. end user quality management) and service billing support (Kalaoja et al. 2003). The wireless service architecture should support various ways of revenue sharing, but as each NO often has its own system for handling wireless customers and collecting billing information the development of operator-independent charging mechanisms is difficult. In I-Mode the NO keeps all the earnings generated by access and airtime and pays back 91 % of the revenues to official SPs (Service Providers) (Poulbère 2003). Chapter 4.5.1 presents the implementation of a service management service for accounting and billing the user transactions in two case services.

2.2 Factors influencing wireless service development

Wireless service development requires taking into account the capabilities and needs of wireless terminals, networks and software, and its architecture, including components, their interfaces and middleware. The factors affecting wireless service development were briefly introduced in Chapter 1 and are explained in more detail in the following sections.

2.2.1 Wireless terminals

The evolution of wireless terminals is developing into direction, where the terminals have the following features:

- they appear in every imaginable shape and form, have high application handling, storage and processing capability (Sharma & Nakamura 2003),

- they support seamless interoperability between different underlying infrastructures, circuit and multiple parallel packet switched connections,
- they are able to generate various categories of tones, take photos with increasing resolution, locate their geographical cordinates, download protocol software and applications, transmit real-time video and audio, and control electronic devices at home and in the office (Funk 2004, Sipilä 2002), and
- they are programmable, safe and secure, enabling the use of personalized services and support context awareness (Baldauf et al. 2001).

Increased memory and other features of the terminals, together with lower package charges, will impact on the service market as people use terminals for purposes maybe not even seen today (Funk 2004, Sharma & Nakamura 2003). The memory, processing power, OS, display size and data input mechanism of wireless terminals set limitations on the services provided and their functionalities. The terminals also present challenges to wireless service development, because they face a temporary loss of network when they move, have scarce resources and are required to react to frequent changes in the environment (Capra et al. 2002).

For deploying the case services of this research, it was used mobile phones and the *multipurpose terminal RIM Blackberry* that belongs to communicators/smartphones. Blackberry that is aimed at corporate users is equipped with a keyboard for easy writing of mail, and it ensures synchronization of actions between the terminal and the corporate e-mail account (Henten et al. 2003). *Mobile phones* are manufactured by dozens of manufacturers around the world, usually designed to work with specific network technologies. Mobile phones provide various kinds of capabilities from voice-only to combined PDA (Personal Digital Assistant) and voice capabilities, and these capabilities define what kind of applications a mobile phone can run. The advantage of mobile phones is their size and the disadvantages are a small display and memory and dependence on network technology (Heyes 2003). The processing power, memory and network speed of terminals are improving all the time, and this enables the introduction of new terminal UIs based on voice recognition and synthesis. Terminal manufacturers nowadays provide Java-enabled phones that can access data for a Java program from different servers (Funk 2004).

The I-Mode case in Japan has proved that terminals play an important part in wireless services as the value of the service is combination of attractive terminals and useful content. In order to provide a good service, the design of the wireless terminal should also be optimized (Sharma & Nakamura 2003). In the European market the limited choice of I-Mode terminals has been a major problem for the success of I-Mode (Henten et al. 2003), but the number of I-Mode users in Europe is increasing; at the moment it is two million (NTT DoCoMo 2004).

2.2.2 Networks

The network in use affects the development of the services, especially via available bandwidth, latency times and connectivity. The design efforts of wireless services to meet network challenges involve delivering the maximum amount of content supported by each terminal and reducing unnecessary requests over the high-latency wireless link.

2G network technologies include GSM, CDMA (Code Division Multiple Access), PDC (Personal Digital Cellular) and TDMA (Time Division Multiple Access) (Funk 2004). GSM has been the prevailing mobile 2G standard in Europe and the Asia–Pacific region (Oliphant 1999). Japan’s 2G standard PDC is based on TDMA standards and operates in the 800 and 1500 MHz bands. CDMA is used in Korea, Japan and USA (United States of America).

2.5G technologies like GPRS (General Packet Radio Service) and EDGE (Enhanced Data rates for Global Evolution) provide a nice transition to 3G network technologies (Sharma & Nakamura 2003). GPRS that is an enabler of 3G networks permits transimission speeds up to 115 kbps (Kilobytes Per Second) and, theoretically, 171.2 kbps (Henten et al. 2003). GPRS provides an "always-on" connection between the mobile terminal and the network, and the network capacity is only used as data is actually transmitted (Sharma & Nakamura 2003).

3G networks are packet-switched networks that enable better use of bandwidth (Heyes 2003) and most of them use different parts of the frequency spectrum more effectively than 2G services. For this reason, packet charges for 3G services in Japan are much lower than 2G packet charges (Funk 2004). Upgrading of 2G to 3G networks may be quite problematic between some techniques as it requires software and hardware upgrades of existing infrastructure and separate base station cabinets that require ground space and, possibly, separate antennae (Sharma & Nakamura 2003). The network's increased data transmission rate and quality also enable new kinds of applications and a richer wireless service environment. The increased transmission speed and improved quality are the main reasons for all the "fuss" that 3G has caused in the recent years.

UMTS, a 3G system standardized by ETSI (European Telecommunications Standard Institute), uses W-CDMA (Wideband CDMA) or UTRA-TDD (UMTS Terrestrial Radio Access Time Division Duplex) as the radio access technology (Sharma & Nakamura 2003, UTMS 2003). CDMA2000 is a 3G technology standard used in South Korea (Henten et al. 2003). CDMA2000 and W-CDMA provide users with data rates up to 2 Mbps (Baldauf et al. 2001, Henten et al. 2003). W-CDMA that is positioned in the 1.9 and 2.1 GHz frequency bands is the universally accepted standard for the 3G mobile networks (Henten et al. 2003).

2.2.3 Software and its architecture

Software architecture includes the application, middleware and, in a broader sense, also system levels. The software architecture also includes the principles that guide the design and evolution of the architecture (IEEE 1992, Perry & Wolf 1992, Shaw & Garlan 1996). The latter definition suits service engineering that is based on integrated software components developed in multi-organizational settings according to a predefined principle. An interface determines how an application component can be used and interconnected with other components, and it enables interoperability between services developed by different APs. APIs (Application Programming Interfaces) are well-defined mechanisms that are built to connect resources like application server, middleware or database. Initiatives like OMA (Open Mobile Alliance) and OSA (Open Services Access) by 3GPP have been established for developing open application interfaces that would enable seamless interoperability between services in multi-AP development. There are several architecture development approaches that can be adopted in wireless service engineering. The Model-Driven Architecture (MDA) (ISO/IEC 2001b) defines a normative model that guides the specification of information systems and increases the integration of the system by separating descriptions of functionality from the implementation specifications.

Middleware that underlies the software architecture and architectural choices provides the illusion of a global system in which separate components behave like a centralized system (Emmerich 2000, Niemelä et al. 1999). The middleware has a crucial role in component-based development as it makes the integration of components designed for certain platforms easier and increases the interoperability. Communication middleware consists of a realization of protocols and architecture that supports distributed systems and computing based on the object-oriented paradigm and message passing. Security middleware contains the essential functions of authentication and access control. Integration middleware serves the integration of the computing platform and enterprise-wide applications (Lerner et al. 2000). Several standardization groups related to wireless service engineering – 3GPP, Tele Management Forum, OMG (Object Management Group) Telecom Task Force and Parlay – aim at community-wide specifications of middleware technologies. Current middleware products vary in their ability to integrate applications, their ease and amount of customization permitted, the programming languages used and the architecture (Heyes 2003). Most of the current middleware products do not support customization and adaptation and are too heavy for resource-scarce wireless terminals. Middleware for wireless terminals should be lightweight, support asynchronous communication between components and make application designers aware of the execution context (Capra et al. 2002; Niemelä et al. 1999).

2.3 Summary

This chapter has analysed state-of-the-art of wireless services that were divided into end user, application support and service management services. In Europe NOs have not yet managed to create functional markets. The reasons for the slow introduction of 3G services in Europe have been the financial crisis in telecom sector, high prices of frequency bands, loose connections between business actors and lack of an appropriate business model (Henten et al. 2003). Entertainment services are believed to be the most profitable service areas (UMTS 2003), but as the real introduction of services based on 3G platforms have been delayed in Europe, it is not quite clear what the "killers" are.

Development of wireless services affects terminals, networks and software. Terminals with constantly emerging new features play an important part in a wireless service, as the value of the service is a combination of an attractive terminal, useful content and an easy-to-use service package. 3G networks enable better use of bandwidth (Heyes 2003) and different parts of the frequency spectrum, and for this reason, packet charges for 3G services are lower than in 2G networks (Funk 2004). The current variety of networks and network technologies will diminish in the future as W-CDMA has been accepted as the universal standard for 3G networks.

The actors in the mobile business interviewed for this research (cf. Paper II) regard mobile applications as easy to make, but managing the whole delivery of the service from design to marketing is very complicated. The development of wireless services has gone towards open interfaces and architectures, but their real implementation still takes time and requires even more changes of attitude from wireless business actors.

3. Business models and business actors' roles

As defined earlier, the business model defines the architecture for the product, service and information flows, including the description of the business actors, their roles, potential benefits and sources of revenue (Timmers 1998). The relationships between the business actors have been handled in many articles, of which those most suited to this research are presented below.

Afuah & Tucci (2001) present a theoretical framework for analyzing Internet business models and analyse several cases in the Internet area. According to these authors, the primary activities of the business network are network promotion and contract management, service provisioning and infrastructure operations.

Camponovo & Pigneur (2003) present the principal classes of actors participating in the mobile business and give an overview of mobile business models. They divide the mobile business domain into five categories: technology, services, network, regulation and user. Camponovo & Pigneur (2003) classify the players in the wireless field according to three main classes: application content, communication and technology.

Plepys (2002) analyses the advantages and disadvantages of the application service provisioning business model. Although application service provisioning has many advantages, realisation requires removing the existing technological and legal barriers.

Sharma & Nakamura (2003) analyse the technologies and business models and their impact on the future of wireless communications. These authors also present business models for different business services, such as a video mail service.

This chapter attempts to answer research question 2 (p. 18) by discussing the business models and networks in wireless services, and focusing on the roles of the business actors and the transactions between them. Defining business models, the roles of the actors and the transactions between them is a basic element of the component-based development of wireless services in a networked business environment. Due to the networked nature of mobile business, the business models presented here focus more on partnership than other elements. This chapter presents a business model framework for two case services that are under development for 3G networks. In the presented services the NO is assumed to have the billing relationship with the customer.

3.1 Wireless business networks

Nowadays, business actors have to both co-operate and compete in order to survive. Co-operation between business actors occurs in business networks that change dynamically over time and usually lack a given center (cf. for example, Håkansson & Snehota 1995), although in some cases one actor is the center of the business network (NTT DoCoMo 2003). The value of a network increases in proportion to the square of the number of nodes on the network, which means that the more subscribers the network has the more valuable it is.

Wireless business networks are customer centric and deliver new, unique value (Tapscott et al. 2000). Business networks have the following benefits:

- zero inventory, distribution, product merchandising and marketing costs,
- near zero product liability, content development, marginal growth costs and revenue risk, and
- focusing on networks makes the final total cost of the product/service lower because of the diminished transaction costs (Jarillo 1988).

A functional business network requires much trust between the network partners, and this raises the question of how this can be created. Jarillo (1988) considers that relating to partners that have similar values and/or a worse situation without the network can create trust. The juridical responsibilities of the network actors are defined by SLAs (Service Level Agreements).

3.2 Wireless business models and actors' roles

In the literature, wireless business models have been defined in many different ways. The business models and roles presented by Afuah & Tucci (2001), Camponovo & Pigneur (2003), Plepys (2002) and Sharma & Nakamura (2003) are discussed in the following section. These models have been selected because of their suitability to this research and their way of thoroughly presenting the various aspects of business models.

Table 3 summarizes the functions of the main business roles that companies play in wireless businesses and the phase of life cycle they are attached to. Every business model has a certain main role, but, as can be seen from Table 4, every model includes other additional business roles.

Table 3. Functions of the wireless business roles.

Role	Function	Life cycle phase
Application provider (AP)	Develops horizontal and vertical applications	Development, Maintenance
Application service provider (ASP)	Provides outsourced computing power on a rental basis	Usage/ provisioning, Maintenance
Content aggregator	Catalyses content and markets it to other actors	Usage/ provisioning
Content integrator	Integrates content and provides it to other actors	Usage/ provisioning
Content provider (CP)	Provides value-added content for SPs, content aggregators and integrators	Usage/ provisioning
End user	Company or individual using the service	Usage
Infrastructure provider	Provides application, network and system infrastructure	Deployment
Network operator (NO)	Sells network capacity of wireless networks to consumers via SPs.	Usage/ provisioning
Service provider (SP)	Provides services for end users	Usage/ provisioning, Maintenance
Terminal manufacturer	Manufacturers wireless terminals	Deployment

Table 4 presents the taxonomy of wireless business models, including their revenue logic, target customers and roles in the network. Reveue logic means a way to make money in the business model. Roles in the network mean the business roles appearing in a certain business model, and the role mentioned first in the list is the main role of the model. The taxonomy of business models illustrated in this section is not exhaustive, as they evolve all the time.

Table 4. Wireless business models (modified from Afuah & Tucci 2001; Camponovo & Pigneur 2003, Plepys 2002 and Sharma & Nakamura 2003).

Model	Revenue logic	Target customers	Roles of the network
Advertising	Fee from advertisers	Advertiser	CP, end user, SP
Application provisioning	License and installation fees, operation and maintenance services	NOs, terminal manufacturers, end users, other actors	APs, NOs, terminal manufacturers
Brokerage	Fee for each enabled transaction	Business and consumer – customer	Broker, buyer, seller
Content provisioning	Subscription and usage fees, revenue sharing	Content aggregator, end user, financial SPs	CP, Content aggregator, end user, SPs, NO
Infrastructure provisioning	Fee for sale or leasing the infrastructure	NO	Infrastructure provider, AP, CP, NO, terminal manufacturer
Network operating	Fees from transmission, subscribers and network services	ASP, end user, ISP (Internet Service Provider)	NO, ASP, infrastructure providers, ISP, terminal manufacturers
Service provisioning	License and installation fees, various methods	Various actors in the value chain	SP, all other roles
a) Application service provisioning	License and installation fees	Various actors of the value chain	ASP, AP, CP, end user, SP
b) Internet service provisioning	Subscriptions and traffic agreement fees	End user, NO, other ISP	ISP, end user, NO
Terminal manufacturing	Fixed fee from sale of terminal	Terminal retailer, NO's distribution channels, end user	Terminal manufacturers, AP, CP, NO

Every business actor can function in the main business role in one or more business models, as well as have a participatory role in the business models of the other actors. Content aggregation and integration are not defined as separate business models in the used literature.

3.2.1 Wireless business models and actors' roles

In the *advertising* model a website provides content and services mixed with advertising messages in the form of banner ads. The website owner makes money by charging advertisers fees for banners and other ways of informing users about their products or services (Afuah & Tucci 2001). One form of advertisement is sponsorship, where users accept marketing as they get free access to content they should otherwise have to pay for (Sharma & Nakamura 2003). Advertising portals have turned out to be very popular in Japan (Funk 2004).

In *application provisioning* APs provide wireless applications and sometimes also platforms (such as middleware and application servers). The value proposition of APs may include different application-related services, such as remote access to a variety of applications that are managed in a central location, with implementation, integration, support and maintenance services. (Camponovo & Pigneur 2003)

In the *brokerage* business model buyers and sellers are brought together and a business actor functioning as a broker charges a fee for transactions they enable. Examples of the brokerage business model are travel agents, online brokerage firms, etc. (Afuah & Tucci 2001).

Content provisioning provides and distributes content delivered by the wireless service, and different companies, such as media houses, use this business model. Categories of wireless content include news, ringtones, music, video, etc. (Poulbère 2003). As the variety of content has been central to DoCoMo's success, the meaning of its importance should not be underestimated (Sharma & Nakamura 2003).

Infrastructure provisioning provides the application, network and system infrastructure needed to run the services. Application infrastructure provides the infrastructure for managing and hosting the applications. Network infrastructure provides the radio and core network infrastructure, such as base stations and radio controllers. System infrastructure provides the systems needed for using the services (Poulbère 2003).

In *network operating* NOs provide wireless network services and capacity to ASPs, end users and ISPs (Camponovo & Pigneur 2003). In Europe the financial markets are pressuring NOs to take a large part of the value in the content market to compensate for their heavy investments in 3G licenses (Sharma & Nakamura 2003).

The aim of *service provisioning* is to provide end users with various services. The target customers in service provisioning are the various actors in the network. The main source of revenue for SPs is license fees, but other payment methods can also be used (Camponovo & Pigneur 2003). In I-Mode the official I-Mode SPs receive 91 % of the charges for approved services from the NO (Poulbère 2003). Application service provisioning and Internet service provisioning are special forms of service provisioning. In *application service provisioning* an ASP offers outsourced computing power on a rental basis, so that the "client accesses the ASP host and runs available applications over a communication line" (Plepys 2002). Application service provisioning enables companies to access the latest technology – which is upgraded when necessary – and use applications via browsers without the need to physically install them in PCs (Personal Computers). The main difference between application service provisioning and the traditional sales of software is its service character. Services are produced and consumed simultaneously, and the customer takes part in the consumption process (Grönroos 1990). In application service provisioning every company can offer various services and thus participate in one or several stages of the value chain (Kallio 2001). Due to companies' reluctancy to put sensitive information in the hands of third parties, and other technological and legal barriers, application service provisioning has not been as successful as was expected some years ago (Plepys 2002). *Internet service provisioning* provides the hardware (like switches, servers) and software that enables users to access the Internet (Afuah & Tucci 2001). Experience of the development of wireless services in Japan has proved that co-operation between NOs and ISPs is important (Sharma & Nakamura 2003).

*Terminal manufacturin*g provides end users with physical wireless terminals that enable them to access a network and to run applications (Camponovo & Pigneur 2003). The competition in terminal manufacturing is intense, and new models with improved features come out even weekly. Additionally, terminal manufacturers have to diversify their products to keep up with the competition (Sharma & Nakamura 2003).

3.2.2 Summary

Succeeding in the wireless business requires co-operation with other actors, as stand-alone companies with a profitable business model will remain limited in number in the long term (Poulbère 2003). The business roles and positions of the business actors in wireless business are dependent on the relationships they have with other firms, and in order to navigate in a network the management must understand how these relationships affect the network and the position between the current marketers (Möller & Halinen 1999).

Effective division of the responsibilities between the business actors improves cost effectiveness, and thus the structure of the business model is a crucial decision to be made before starting service development. When choosing a business model, a business actor has to consider the availability and suitability of existing business models, their current partners and chances to acquire new ones, their core competence and ability to manage several roles at the same time and expectations of the customers. Additionally, charging for the services and tracking the customer transactions should be taken into account. The financial transactions between the actors are tracked by the aid of accounting and billing. Accounting is "the process of keeping track of a user's activity while accessing the network resources" and sharing the revenue from the use of the services (Internet.com 2003). Billing generates records by way of the charging function and transforms them into bills requiring payment. Charging processes the data exchange information to determine the usage for which the user will be billed (Koutsopoulou et al. 2001). NTT DoCoMo in Japan has introduced the BOBO (Billing on Behalf Of) function that enables CPs to collect revenues so that a certain portion of the income from each subscriber goes back to the CP automatically (Sharma & Nakamura 2003).

The following sections present a business model framework for two wireless services.

3.3 Case services

For developing a framework for a wireless business model, two wireless case services developed for the 3G networks were evaluated in this research (for more detail, see Papers II and VI). These services were selected as they require complex relationships in their value chain and advanced business models between the actors. The features of the case services are listed in Appendix B. Table 5 presents the general concepts used in the case services and their definitions. Definitions of the business roles used in the case services were presented in the previous section.

Table 5. General concepts used in the case services.

Concept	Definition
Accounting	Process of keeping track of user's activity and sharing revenue
Application	Software that operates on a host server and/or on a wireless terminal
Billing	Process where charging records are transformed into payment bills
Business actor	Company/ individual acting in business network in one or several business roles
Business model	Architecture for the product/service and information flows, including description of actors and their roles, potential benefit for business actors and sources of revenue
Content	Information the end user wants and is willing to pay for; the quality of the delivery is what has been agreed
Information flow	Flow that controls monetary and product flows
Monetary flow	Money flow between business actors
Product flow	Physical or digital delivery of the product to other business actors
SLA	Juridical document that defines the responsibilities of the business actors
Transaction	Monetary, product or information flow between business actors
Wireless service	Service event realized after receiving a service request from the terminal most often by running some part of the componentized application on server over network and some part on terminal.

The concepts presented in Table 5 are applicable in development of all wireless services.

Case service 1 is a wireless interface to the stocks/news TOL (Trading On-Line) system based on the HTTP (Hyper Text Transfer Protocol) protocol. The system is a website that manages the stock quotes, news information and portfolio of the user. The end user can monitor the stocks and news in real time with an auto-refresh pull method and create his/her own portfolio. The effort in developing case service 1 has been to enable wireless access to the current TOL service, which, at the moment, can only be accessed via fixed-line connections.

Case service 2 is a multi-player game, known as the “Labyrinth Game” that will be developed in the WISE (Wireless Internet Service Engineering) project. A game was selected as a case service because games and other forms of entertainment are expected to be the “killer services” of 3G. Case service 2 is an arcade game where multiple players move in a big labyrinth or dungeon and complete a mission. The client side is implemented using MIDP (Mobile Information Device Profile) – that is, a set of J2ME (Java 2 Platform, Micro Edition) APIs that define how software applications interface with mobile phones. The server side is implemented using J2EE (Java 2 Platform, Enterprise Edition). The user can download the game "over the air", directly onto the terminal through a serial cable or infrared connection. Clients use a GPRS/UMTS connection for connecting to the server for data exchange and have access to a GPRS network, which is connected to the Internet by means of a GGSN (Gateway GPRS Support Node). After gaining access to the Internet, clients can connect the Game Server, which uses the service management services provided by a different server running on a node on the same LAN (Local Area Network) (Lago et al. 2003).

3.3.1 Business models in the case services

This section presents the business models of the case services, including the business actors and the transactions between them. The business model framework includes transactions in all phases of the service life cycle.

3.3.1.1 Case service 1: On-line trading system

Figure 5 presents the framework for the business roles and model of case service 1. The transactions between the actors are defined in more detail in Table 6.

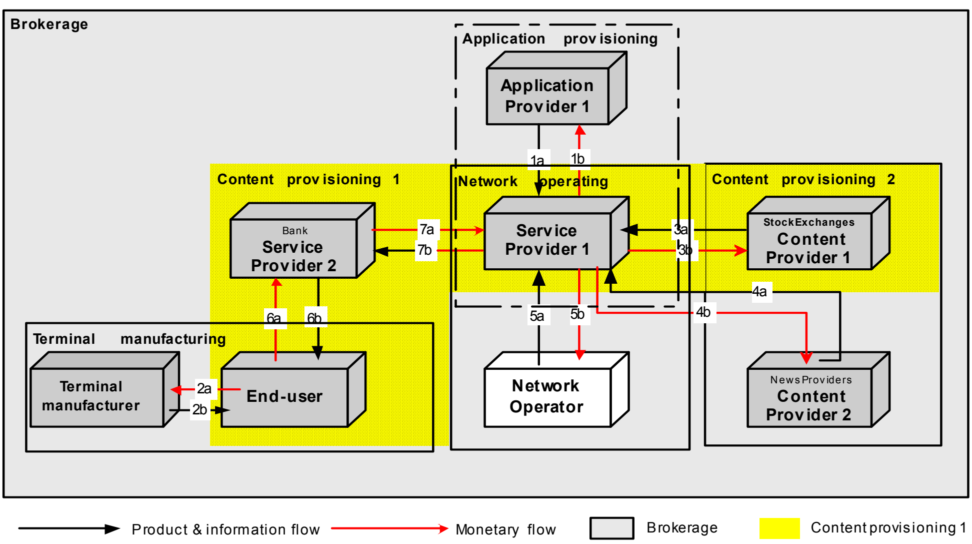


Figure 5. Framework of case service 1's business roles and model.

Case service 1 uses brokerage as the main business model. Additionally, the service includes application provisioning, network operating and terminal manufacturing business models and two content provisioning business models. The business model application provisioning includes the business roles AP1 and SP1; the network operating business roles NO and SP1; the terminal manufacturing business roles terminal manufacturer and end user; the content provisioning 1 business roles end user, SP1, SP2 and CP1; and the content provisioning 2 business roles CP1 and CP2.

In case service 1 the business idea for the bank is outsourcing the service. Bigger banks can afford to develop this kind of application themselves, so the SP1 customers are mainly small banks. SP1 tries to reach competitive advantage by offering a high level of service and tailoring of services. The service level between SP1 and the banks is defined in SLAs. The current problem with the use of case service 1 is reliability: dropping out of the network and changing operator means disconnecting from the network and, in the worst case, a disconnected purchase operation. Table 5 presents the transactions between the business actors and revenue logic used in the life cycle phases: development, deployment and usage/provisioning of case service 1.

Table 6. Transactions (flows) between business actors and revenue logic used in case service 1.

Flow nr	Flow ↔	Revenue logic
Development phase		
1a, 1b	AP ↔ SP1	AP provides the application to SP1 that pays a certain fee for it
Deployment phase		
2a, 2b	Terminal manufacturerer ↔ end user	Terminal manufacturer provides the end user with the terminal and the user pays a fixed fee for it
Usage/ provisioning phase		
3a, 3b	CP1 (stock exchanges) ↔ SP1	SP1 receives stock updates from the stock exchanges and SP1 pays a certain fee per user until 1,000 users is exceeded. After 1,000 users, additional users cause no costs to SP1
4a, 4b	CP2 (news-providers) ↔ SP1	SP1 receives news from news providers and pays a certain fee per view
5a, 5b	NO ↔ SP1	SP1 pays for the use of bandwidth during service use to NO
6a, 6b	End user ↔ SP2 (bank)	User pays the bank a fee to gain entrance to the application
7a, 7b	SP1 ↔ SP2 (bank)	SP1 provides SP2 the service that pays a certain fee for it

As can be seen from Table 6, there are numerous flows in the network of case service 1 and successful management of the service development and implementation requires proper defining of the actors' interaction and scalable business network.

3.3.1.2 Case service 2: Multi-player game

Figure 6 presents the framework for the business roles and model in case service 2; the SLAs define the responsibilities of the actors.

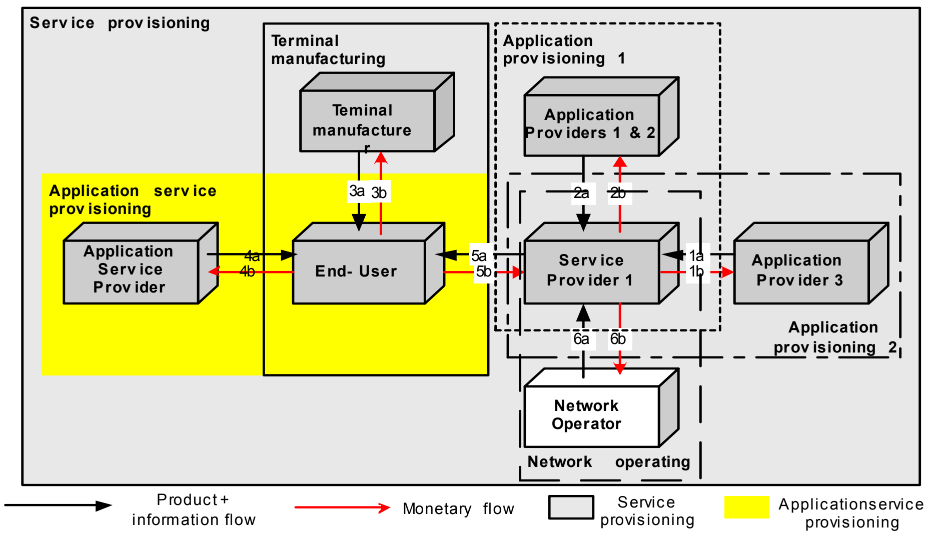


Figure 6. Framework of case service 2's business roles and model.

The main business model of case service 2 is service provisioning that includes all the business roles of the service. Additionally, case service 2 includes two business models for application provisioning and a business model for application service provisioning, network operating and terminal manufacturing. The business model application provisioning 1 includes business roles AP1, AP2 and SP1 and application provisioning 2 business roles AP3 and SP1. The application service provisioning business model includes the business roles ASP and end user; the network operating business roles NO and SP1, and the terminal manufacturing business roles terminal manufacturer and end user.

In case service 2 the NO provides the bandwidth and SP1 handles the service delivery. The user pays a certain fee for the network traffic plus a "lump sum" every time he wants to play. The ASP provides SP1 access to the game application and AP3 a service management component that enables services like authentication, billing, etc. The terminal manufacturer sells the wireless terminals needed for using the service.

Table 7 explains in more detail the transactions between the business actors and revenue logic used in the life cycle phases development, deployment and usage/provisioning of case service 2.

Table 7. Transactions (flows) between business actors and revenue logic used in case service 2.

Flow nr	**Flow ↔**	**Revenue logic**
Development phase		
1a, 1b	AP3 ↔ SP1	AP3 provides service management component to SP1 that pays a fixed fee for it
2a, 2b	AP1 & AP2 ↔ SP1	AP1 and AP2 provide SP1 with the server side of the game and SP1 pays a fixed fee for it
Deployment phase		
3a, 3b	Terminal manufacturer ↔ end user	Terminal manufacturer provides the end user with the terminal and the user pays a fixed fee for it
Usage/ provisioning		
4a, 4b	ASP ↔ end user	ASP provides the end user with the client side of the game and the user pays a fixed fee per download or nothing
5a, 5b	SP1 ↔ end user	SP1 provides access to the service and the end user pays for using it based on the number of bits transferred, the character chosen, and the information requested during the game
6a, 6b	NO ↔ SP1	NO provides the SP1 bandwidth and SP1 pays for it according to the number of users

Table 7 also proves that the huge number of flows between the business actors sets challenges for the management of the service and its development in a dynamic network environment. Chapter 4 presents an accounting and billing application for tracking the transactions between the business actors.

3.4 Comparison with related work

The following section compares the ideas of this research with related work, and draws some conclusions.

3.4.1 Business roles and networked business models

Table 8 compares the business models and their revenue logic, target customers and business roles in articles presented by Camponovo & Pigneur (2003), Afuah & Tucci (2001), Plepys (2002), Sharma & Nakamura (2003) and the case services of this research. Case service 1 of this research used the brokerage and content provisioning business models (yellow background) and case service 2 the application service provisioning and service provisioning business models (blue background). Both case services used the application provisioning, network operating and terminal manufacturing business models (pink background).

Table 8. Comparison of the business models and their revenue logic.

	Revenue logic		**Target customer**		**Roles in network**	
	Plepys (2002)	**This research**	**Plepys (2002)**	**This research**	**Plepys (2002)**	**This research**
Application service provisioning	License and installation fees	Fixed fee per download	Various actors	SP	ASP, AP, CP, end user, SP	ASP, end user
	Afuah & Tucci (2001)	**This research**	**Afuah & Tucci (2001)**	**This research**	**Afuah & Tucci (2001)**	**This research**
Brokerage	Fee for each enabled transaction	Fixed fee per user/ transaction	Businesses, consumers	SP	Broker, buyer, seller	CP, NO, end user, SP
	Sharma & Nakamura (2003)	**This research**	**Sharma & Nakamura (2003)**	**This research**	**Sharma & Nakamura (2003)**	**This research**
Content provisioning	Subscription & usage fees, revenue sharing	Fee per view/ per user	Content aggregator, end user, financial SPs	SP	CP, Content aggregator, end user, SPs, NO	CP, end user, SP
	Camponovo & Pigneur (2003)	**This research**	**Camponovo & Pigneur (2003)**	**This research**	**Camponovo & Pigneur (2003)**	**This research**
Application provisioning	License and installation fees, operation and maintenance services	License fees; later also maintenance	NOs, terminal manufacturers, end users, other actors	SP	NOs, other APs, terminal manufacturers	Other APs, SP
Network operating	Fees from transmission, subscribers and networks services provided	Number of users, bandwidth used	ASP, end user, ISP	SP	NO, ASP, infrastructure providers, ISP, terminal manufacturers	NO, SP
Service provisioning	License and installation fees, various methods	Usage fees (various methods)	Various actors	End user	SP, all other actors	SP, AP, NO, end user, technology provider
Terminal manufacturing	Fixed sum per terminal	Fixed sum per terminal	Terminal retailers, NOs, end users	End user	Terminal manufacturers, AP, CP, NO	Terminal manufacturer, end user

The comparison of revenue logic, target customers and business roles of the business models in Table 8 proves that the business roles of the case services equate with the roles presented by the related literature, but the case services also include roles not mentioned in the literature. The analysis of the target customers and revenue logic of the case services reveals that the current business models do not take all the possible ways of earning money into account extensively enough. According to this research, new options should also be added to the current revenue logic and possible target customers of the business models in the literature.

3.5 Conclusions

This chapter sought an answer to research question 2 (p. 18) by presenting the potential business models and roles of the wireless business actors and the transactions between them; this thesis did not, however, calculate profit margins for the various services and business actors.

The related literature defines at least 20–30 various terms for business roles, but as some of them overlap as the main roles in the wireless service business, with which most wireless business models can be formed, they can be regarded as AP, ASP, CP, infrastructure provider, NO, SP and terminal manufacturer. Study of the case services in this research showed that defining a general business model is hard and, therefore, the most appropriate business model has to be defined case by case based on circumstances, the core-competence of the actors and end-user needs. The study of the business models of the case services also showed that the current business models need to be modified by adding new roles and concepts of revenue logic.

The development of services in a networked environment requires defining the business actors and their roles and transactions, as an appropriate business model orders the success of a service, especially in component-based development.

4. Component-based development of wireless services

Component-based development is based on use of software components that are either developed in-house or bought from the commercial market. The need to cut costs and reduce time to market has substantially increased the use of commercial, third-party components, including COTS (Commercial-Off-The-Shelf) and OCM (Original software Component Manufacturing) (Seppänen et al. 2001). COTS-based services are constructed by integrating large-scale components provided by third parties. To make the integration of components into a service easier, the architectural assumptions of the components should be properly documented by using standards, document design and quality assurance processes (Sommerville 1992).

This chapter presents the chances of component-based development in development of wireless services. This chapter also evaluates the risks and evolution of commercial component use in three case projects. As proper documentation is required to verify the capabilities of a third-party component and assess its applicability to the architecture, this chapter also presents a documentation template for software components. Finally, this chapter evaluates the suitability of component-based development to wireless services by developing two wireless case services: an accounting and billing service and a secure home network.

4.1 Quality demands of components

Quality is the degree to which a component meets the specified requirements and user's expectations (IEEE 1990). The developer of a service also expects the component to fulfill certain quality requirements, and, in case these are not met, the use of a component may cause various risks in use, as section 4.3.4 presents. Evaluation of component quality needs quality attributes, of which the following are appplicable to this research:

- *Functionality* is the ability of the component to do the work for which it was intended (Bass et al. 1998). In case the component does not function as expected or is incompatible with the system or other components, the service does not do what is expected of it. Functionality is one of the biggest problems of component-based development as many components are designed to work as standalone or they use proprietary interfaces.

- *Interoperability* is the ability of components to interact with one or more specified systems (ISO/IEC 2001a). Interoperability is considered when components and their interactions are defined in detail and finally observed as executable models, simulations and running systems. Interoperability is achieved if the services are generic and new services can easily be integrated by aggregating the old ones. (Niemelä 1999)
- Software *maintenance* includes all the activities required to evolve a software system over its lifetime (Vigder & Dean 2000). *Maintainability* is the ability to modify a software component in order to correct faults, improve performance or adapt to a changed environment (IEEE 1990). *Modifiability* determines how the component can be modified for a new environment and new changes quickly and cost effectively (Bass et al. 1998). Modifications to a wireless service include adding, deleting and changing structures, and, therefore, expandability; portability can be considered a special form of modifiability (IEEE 1990). *Expandability* describes how new features can be added to the component to expand its functionality (Niemelä 1999). *Portability* is the ability of a component to be transferred from one environment to another (IEEE 1990). Middleware standards and platform-independent languages address portability issues (Brereton & Bugden 2000). Commercial use of a non-portable component is difficult, especially if it is designed to work in a platform that is not in general use.
- *Performance* can be measured, for example, 1) by response time that is the time required to respond to one or several events processed in some time or interval (Bass et al. 1998), 2) by transfer capacity that is the maximum sustainable rate of information transfer (American National Standards for telecommunications Agency 1993) or 3) by latency that is the amount of time it takes for a packet to travel from source to destination (Internet.com 2004b). Performance is one of the key features of wireless services, because if, for example, a user has to wait too long for a site to download, he may move to another site. Depending on the sensitiveness to delay, services have different performance requirements (3GPP, 2003b).

- Information *privacy* exists when the usage, release and circulation of personal information can be controlled (Culnan 1993). Invasions of privacy occur when individuals cannot maintain a substantial degree of control over their personal information and its usage (Lim 2000). If privacy cannot be guaranteed, use of the service enables fraud, which causes millions of lost dollars every year (Sharma & Nakamura 2003).
- *Reliability* of a component is its probability of failure-free operation in a given environment (Krishnamurthy & Mathur 1997). Reliability assesses the tolerance of the wireless application towards problems caused by, for example, the network. Applications that depend upon the integrity of transmitted data, such as credit card transactions, cannot afford a network that often drops data (Heyes 2003).
- *Security* is the ability of a component to protect information and data, so that unauthorised persons or systems cannot read or modify it and authorised persons or systems are not denied access to it (ISO/IEC 2001a). Security of components can be guaranteed by, for example, building security properties inside components (Khan & Han 2002). Flaws in the logic and implementation can result in security holes that will be exploited by attackers or malicious websites. The physical limitations of a wireless terminal often force application developers to make security and performance trade-offs, and, for example, security features in advanced languages may be omitted in APs' implementations (Ghosh & Swaminatha 2001).
- The number of *side effects* caused by the use of a component can also be regarded as a quality feature of a component. Even if the component itself works as expected, in new environments and use situations its use can cause unexpected side effects on the total service functionality.

The business actors in wireless services prefer different qualities. An external quality provided by one business actor is a prerequisite for the quality of another actor in the quality 'stack'. The quality is only achieved if the prerequisite qualities are met. In order to be profitable from the SP's point of view, the services should be portable, modifiable for different end-user services, expandable, maintainable and easily used and accessed by APs.

4.2 Component-based software engineering (CBSE)

In recent years, CBSE has received interest from numerous sectors of industry due to its potential for improving quality, productivity, performance, reliability, flexibility and interoperability (Brown & Wallnau 1998, Sametinger 1997). The success of CBSE is also due to a need for a new software development approach that addresses the needs of business actors for managing complexity and rapidly adapting to change (Brown & Wallnau 1998). The foremost distinction between a traditional and a component-based system is that CBSE is principally an effort of iterative selection and integration of components (Gentleman 1997, Oberndorf et al. 2000). Component-based architectures should be modular, which eases their composability and independent development (Messerchmitt & Szyperski 2001).

Table 9 presents sources of information on CBSE, many of which are used in this research. The first column of Table 9 presents the name of the source and the second column the topic handled in the source.

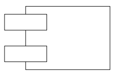
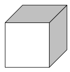
Table 9. Sources of information sources on CBSE.

Source of information	**Topic**
http://cs.ua.edu/components/research/index.htm	Software component research links
http://wwwsel.iit.nrc.ca/projects/cots/bibliography.html	Bibliography of COTS-related literature
[Bass et al. 1998]	Software architectures in practice and reuse
[Brereton & Budgen 2000]	Key issues, stakeholders, and a framework for component-based systems
[Brown & Wallnau 1998]	State and perspectives of CBSE
[Garlan et al. 1995]	Architectural mismatch of component use and their integration
[Meyers & Oberndorf 2001]	How to manage software acquisition
[Morisio et al. 2000]	Evaluation of 15 COTS projects
[Sametinger 1997]	Software engineering with reusable components, and its advantages and disadvantages
[Seppänen et al. 2001]	State-of-the-practice of software components produced in strategic partnerships
[Shaffer & McPherson 2001]	Practical COTS risk-mitigation guide
[Szyperski 1998]	Technical foundations of component software and its importance in the software market
[Vigder 1998]	Defining architecture for COTS-based systems
[Vigder & Dean 1997]	Architectural approach to building systems from COTS software components
[Vigder & Dean 2000]	Maintenance aspects of COTS-based systems
[Voas 1998]	Challenges of using COTS in CBSE

Components make their services available through interfaces. In blackbox components, clients do not see any details about the components and its source; in whitebox components the component and its source can be modified (Szyperski 1998). The blackbox character of commercial components makes their use impossible in cases where they cannot be implemented in the system as such.

Table 10 presents the general elements of component-based software development.

Table 10. Elements of component-based software development.

Element	**Name**	**Definition**
	Component	Unit of composition with contractually specified interfaces and explicit context dependecies. Software component can be deployed independently and is subject to third-party composition
	Node	Run-time physical object that represents a processing resource
	Dependency	Relationship between two elements that indicates that changes to the target element may cause changes in the source element
	Interface	Specifies the externally visible operations of a class, component, package, or other element without specifying internal structure

Name	**Definition**
Compiler	Program that converts another program from some source or programming language to machine language (object code) (for example Java, C++, C)
Component developer	Application or middleware (component) provider that develops the components
Component platform	Platform that allows installation of components
Component integrator	Integrates components into applications
Component technology	Technology used for developing components
Composition	Internal structure of the component
Composition platform	Platform needed to integrate the component with other components
Configuration	Describes the way the component is initiated into different contexts. Defines the variation points where the component behaviour can be changed, and guidelines on how to do it
Design language	Language used for modelling software (like Prosa, UML (Unified Modelling Language))
Development environ-ment	Environment in which the component has been developed
Development language	Programming language used for developing the component
Execution platform	Defines the environment in which the component can be executed
Interface	Defines how a component can be interconnected with other components and reused by defining the operations available to a component
Middleware	Software that insulates applications from the underlying wireless network
OS	Program that controls the execution of application programs and acts as the interface between the applications and the computer hardware
Pattern	Package of design decisions that can be reused to achieve quality
Software architecture	Structure/ structures of the system, comprising software components, the externally visible properties of those components and the relationships between them
Third-party component	Component provided commercially by another business actor

The elements described in Table 10 are used for describing component-based development in general and in detail, and the factors affecting it.

To make the integration of the components into a system easier, the architectural assumptions should be explicit and properly documented. Component developers should provide techniques for bridging mismatches and develop sources of architectural design guidance, such as handbooks (Garlan et al. 1995). Patterns are a package of design decisions that can be reused to achieve system and component quality (Bass et al. 1998). Design patterns that are "descriptions of objects and classes that are customized to solve a general design problem in a particular context" (Gamma et al. 1995) can be used during the whole design phase. An interface defines how a component can be interconnected with other components and reused by defining the operations available to a component. Some interfaces are based on open interface standards, but others are only useful in certain programming languages (Linthicum 1999). The required interfaces are those the component needs to operate, and the provided interfaces are those the component offers to other components.

Maturing of component technologies has increased understanding of component-based development, but use of the technologies is still not totally unproblematic as they impose constraints on the components. Component development environments and languages dominate application development. Component technologies that are somewhat standardized are the OMG's CORBA (Common Object Request Broker Architecture), Sun's JavaBeans and Enterprise JavaBeans, and Microsoft's Component Object Model (COM) and Distributed COM (Brown & Wallnau 1998).

Components have so far been made more or less dependent on component platforms that allow installation of components (Szyperski 1998). Execution platforms define the environment in which the component can be executed, such as OS or hardware. A composition platform that is needed to integrate the component with other components is, for example, graphical UI. Open platforms enable mixing of hardware, software and networks from various APs, and thus high composability and portability of components (Sametinger 1997). A component's dependency on a certain platform is one of the biggest problems when using commercial components (cf. Vigder 1998).

Outsourcing of systems and increased use of third-party components demands many improvements in how components are documented, assembled, adapted, and customized (Brereton & Budgen 2000). Components that can be used by third parties should be clearly identifiable, describe and/or perform specific functions, have clear interfaces and hide details not needed for reuse, and have a defined reuse status that defines the owner, maintainer and status of the component (Sametinger 1997).

4.3 Evolution and risks of commercial component use

The following sections present the evolution and risks of commercial component use.

4.3.1 Evaluation framework for commercial component use

The evolution of software component use has previously been studied mainly from the point of view of programming languages. The views presented in this section have some similarities with this research (like evolvement over time), but the scope of this research is wider and includes design languages, OS and middleware.

Szyperski (1998) researched the evolutions of component development tools and environment. According to him, almost any language can be used in programming software components. Object-oriented programming is expanding alongside component-oriented programming, but other paradigms, such as functional programming, can also be used. Szyperski (1998) thinks that component frameworks are appearing on the market, but the architecture integrating the component system is still missing.

According to Rumbaugh (1987), the object-oriented languages have evolved over time as the standard languages have been inadequate for defining reusable components, because, for example, the granularity of abstractions has been too large and the objects cannot be parametrized.

Vigder & Dean (2000) present that elements attached to the connection infrastructure (CORBA, RMI (Remote Method Invocation)), environment, collaborations, interfaces, control mechanism and interconnection topology cause dependencies between COTS, and management of these dependencies enables the evolution of a COTS-based system.

4.3.1.1 Evaluation framework applied in component use

This section presents the evaluation framework of the commercial component use applied in the evaluation of the three case projects in section 4.2.3. (for more detail, see Paper V). This evaluation framework is presented in Figure 7.

Every point in Figure 7 presents a commercial component used in the case project of this research and is thus not technologically exhaustive. The further a certain component is from the intersection of the axes in Figure 7, the more portable it is into different environments, and the more configurable it is with other commercial components. The four axes of the framework – compilers, OS, middleware and design languages – were selected for their importance in the architectural design.

Compilers are programs that convert programs from a source or programming language to machine language (object code) (Farlex 2004). Java compilers are more configurable and portable than C and C++, which do not provide adequate support for the component-oriented programming principles (Szyperski 1998). Java is a simple, secure, architecture-neutral and portable language that enables Java applications to be executed anywhere on the network with the aid of an architecture-neutral object file format that the compiler generates (Sabharwal 1998, Sun 2003, Szyperski 1998). Java bytecode is programming code that, once compiled, is run through a Java Virtual Machine (JVM) instead of the computer's processor. This approach enables running source code on any platform once it has been compiled and running it through the JVM. Once a Java program has been converted to bytecode, it can be transferred across a network and executed by JVM (Internet.com 2004a). Sun Microsystems developed Java bytecode to support mobile code on the World Wide Web (Hartman et al. 1996).

Operating system is "a program that controls the execution of application programs and acts as the interface between the applications and computer hardware" (Stallings 2001). To respond to the needs of the evolving hardware and applications, a wide range of approaches and design elements – such as object-oriented design – have been tried in OSs (Stallings 2001). Own OSs are designed for a certain purpose and are less portable than real-time or open source OSs. QNX is a real-time OS that has modular architecture, supports multiple users' access to the system and allows the development of easy configurable and modifiable software (QNX 2003, Sankowski et al. 2003). Open-source OSs – like Linux – give anyone the right to modify the source code and redistribute it, which allows total modifiability of the system.

As stated earlier, *middleware* insulates applications from the underlying wireless network, making new applications easier to develop and port to a new environment (Sharma & Nakamura 2003). For connecting components, it can be used with its own middleware, RMI or CORBA. Of these, CORBA is the most portable and configurable, own middleware less. RMI is an object-oriented Remote Procedure Call mechanism of Java environments (Campadello et al. 2000). One implementation of RMI architecture is CORBA – that is, a general mechanism that can be realized with different languages. The CORBA specification defines interfaces and components that compose an ORB (Object Request Broker) that is used for sending messages to the object and receiving the results (Suzuki & Yamamoto 1999). Orbix is an ORB that is available for a large number of platforms (Szyperski 1998). Version 2 of Java supports CORBA by enabling the sending of remote calls to program methods run in different computers. Other middleware technologies used in wireless services are, for example, JavaSpaces and TSpaces (Capra et al. 2002).

Design languages, like Prosa and UML, are used for modelling software. Prosa (2003) supports UML, but as a universally accepted standard, UML is more modifiable and portable than Prosa. Prosa/OMT (Object Modeling Technique) is a visual software-modelling tool that offers an integrated environment for the design documents and structured analysis (Oman et al. 1990). The UML is a language for specifying, visualising, constructing and documenting the artefacts of software and non-software systems, and for business modelling. UML also represents a collection of engineering practices that have proven successful in the modelling of complex systems. Therefore, most developers are expected to choose UML for their modelling work in the long term (Suzuki & Yamamoto 1999).

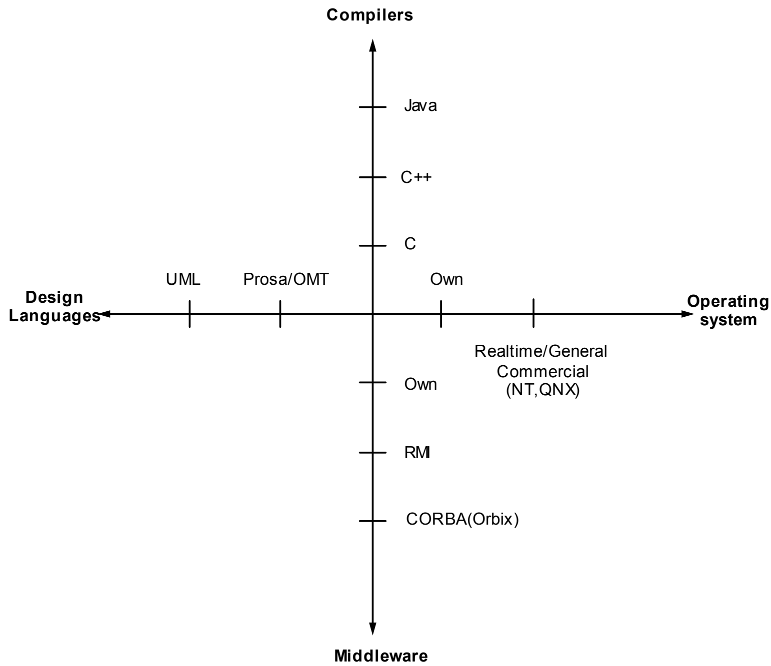


Figure 7. Evaluation framework of commercial component use.

The framework presented in Figure 7 is domain-specific and changes as the technologies evolve, so it should be updated regularly and can be used in a comparison of components at certain moment in time. The distance between the points in the axes does not, however, have any significance because the metrical values vary according to the domain and should, therefore, be defined separately for every domain.

4.3.2 Risks in commercial component use

A fundamental risk issue within the use of COTS components is their interoperability across time and space. A mismatch of COTS components can arise for many reasons. So far, the COTS components have been blackbox components that hide the details of their internal state from its user, thus the components may be used in ways that were not foreseen by the component provider, which, in this case, means AP. This section presents the technological and economic risks related to commercial component use.

Garlan et al. (1995) describe the integration of four subsystems in an Aesop environment. In their research they encountered six main functional problems: excessive code, poor performance, need to modify external packages, a need to reinvent existing functions, unneccesarily complicated tools and an error-prone construction process of the integration. The problems of this research were due to the difference between the requirement specifications and real functionality.

Sametinger (1997) describes the economic, conceptual, technical, managerial and organizational problems of commercial component use. According to him, most research is done on technical support for software reuse by not always taking the risks into account.

Morisio et al. (2000) analyze 15 COTS projects performed at the Flight Dynamics Division at the Goddard Space Flight Center of NASA (National Aeronautics and Space Administration). The analyzed projects represent a variety of software domains and use more than 30 different COTS products. Some of the related problems identified by the leaders of the researched projects were: slippage in schedule because of delays in releasing the COTS by the AP; documentation on the product was not available, incomplete or not reliable; some functions were promised but never implemented; modifications made by the AP altered the compatibility of one COTS with other COTS, or the rest of the system, or introduced new bugs, and communication with the AP was one way as they were hard to reach. The selected COTS filled the requirements to at least some extent while others reported integration problems later in the project. According to Morisio et al. (2000), a COTS project must accept effort and cost as a tradeoff with expected gains in schedule, and managing these tradeoffs is crucial to the success of a COTS project.

The risks related to commercial component use are handled in more detail in the following section. The risks are divided into five main topics according to their character: security, quality, maintenance, functionality and economic risks.

4.3.2.1 Risks in commercial component use in detail

The risks of commercial component use presented in this section include: quality practices in component development and its documentation, functionality, maintenance, security and economic risks.

The quality of commercial components can vary significantly from one commercial component to another due to differences in the *quality practices* of the component development. These differences affect the quality of the component and, therefore, the desirability of the products that are offered for sale. The importance of component documentation was noted in a practice survey, which examined the development needs of component software (Seppänen et al. 2001). In that research the component documentation was seen as one of the bottlenecks in the component development, acquirement and utilisation processes. The three case projects in this research also prove that design, testing, component validation and documentation is often inadequate and/or wrong. The lack of a standard documentation template, the different quality practices of different manufacturers and publicly unknown terminology of the documentation are some causes of the weak documentation quality of commercial components.

Functionality of the components is one of the biggest challenges for commercial component use. The functional mismatch of the component can be due to inadequate or excessive functionality of the component compared with requirements that are set to it by the user (Vigder 1998). A functional mismatch can also be due to the inadequate knowledge of the component user regarding the system functionality and product and integration requirements. The architecture mismatch occurs when components fail to meet the architectural constraints (Brereton & Budgen 2000). Some components are designed to work as standalone applications and may not easily interact with other components or software due to a lack of integration mechanisms or suitability for running in a dedicated environment. The component may be incompatible with the domain and architecture, or with system features like hardware platforms, business processes and operational environment. Even software designed for integration may not meet the functionality requirements of the component integrators. Additionally, users may use commercial components in a way that was not foreseen by the component developer, and this causes unexpected side effects (Vigder 1998). The component providers can limit their level of responsibility for the components in use, and for the integrated products, by SLAs (Brereton & Budgen 2000). A functional mismatch can also occur between the requirements of the standards on a certain domain and the practices of the component provider. The ability of components built to a variety of commercial standards to operate with each other can be limited due to the use of proprietary interfaces. The rapid obsolescence and replacement of commercial components requires stable hardware interfaces and software protocols designed to be “open” to the use of many products from different sources (McDermid 1997).

The location of system components across multiple organizations complicates the evolution and *maintenance* of a component – based system (Brereton & Budgen 2000). The difficulty with maintenance of a system based on commercial components is due to the fact that the evolution of the components is outside the direct control of the system developers and acquisition organizations as the components are maintained and supported by the component developer (Vigder & Dean 2000). Maintenance of the system is especially difficult when the components are not supplied in source form (Gentleman 1997). The AP support and stability causes risks for maintenance if the existence of the AP is uncertain due to its economic difficulties or ability to provide customer support. To avoid warehousing expenses and an unmarketable inventory, the component manufacturer often limits the support period of the product by using a “just-in-time” ordering strategy (Shaffer & McPherson 2001).

The use of commercial components introduces potentially significant information *security* risks for three reasons: 1) the increased inter-operability among different components that meet commercial standards raises the chances of unauthorized access, 2) the use of commercial standards allows a greater number of people to be familiar with the software protocols used to manage the information and thus access or disrupt the information flow (Shaffer & McPherson 2001) and 3) the component can include functionality that is not documented. In using commercial components, security risks cause also their delivery over unsecured delivery channels, the possibility of deliberately built-in hidden doors and the need for the system to have a level of security that is not available within the individual COTS components (Vigder 1998).

An often occurring *economic risk* of commercial component use is the underestimated total costs that are caused by not taking the investment required in own infrastructure, methodology, etc., into account (Sametinger 1997, McDermid 1997, Meyers & Oberndorf 2001). In some circumstances the unit costs of commercial component use are more expensive than combined use of commercial and self-made components (Hall & Naff 2001). Component adaptation may increase maintenance costs for integrated systems because developers must re-adapt or re-glue components as they evolve (Brereton & Budgen 2000). Cost considerations that also need to be taken into account in component-based development include inadequate planning costs, test and integration costs, modification costs, configuration management costs, continuous system engineering costs and obsolescence management costs (Shaffer & McPherson 2001). The following sections present the use of commercial components in three case projects, and their evolution and realized risks.

4.3.3 Commercial software component use in the case projects

This section discusses the evolution and experiences of commercial component use and its risks in three projects: Arttu, Dynamo and VHE. The concept of "software component" in this context means a COTS component. The first research project lasted from 1996 to 1997, the second 1997–1998, and the third 2000–2002 (for more detail, see Paper V).

Figures 8, 9 and 10 present the adaptation of a framework presented in Figure 7 in the researched projects. The standard buses, protocols and other COTS components are not included in the axes as comparing them is currently difficult because of the immature technology. As the aim of this project evaluation is purely to illustrate commercial software component use, more exact details of the implemetations are beyond the scope of this research.

The following sections present the evaluation of commercial component use in the Dynamo, Arttu and VHE projects (Paper V). The evaluation serves research problems 3 and 5 (pp. 18–19) by illustrating the risks and evolvement of component use over time and the effects as the technology develops further. Although Arttu and Dynamo are distributed systems, experience received from them serves research problematic for this thesis, as problems of commercial component use are generic.

4.3.3.1 Dynamo

The Dynamo project (1996–1997) evaluated the use of commercial middleware by installing Visibroker with Orbix (Borland 2003, Iona 2003). Visibroker is a CORBA-compliant ORB implementation written in Java (Szyperski 1998). Figure 8 presents the framework of the Dynamo project. When the Dynamo project was started in 1996, a draft version of UML existed, but version 1.0 did not come out until 1997.

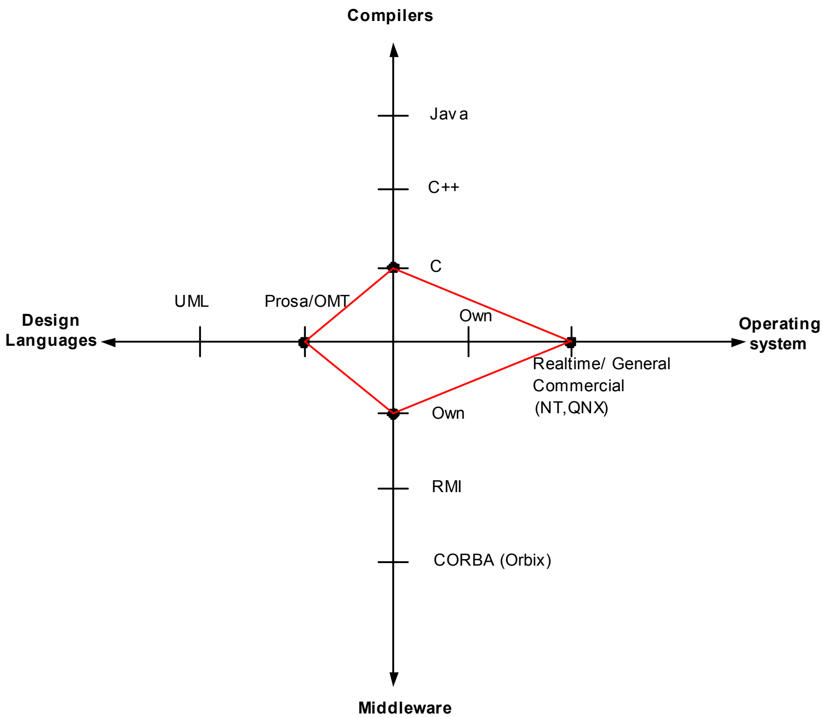


Figure 8. Framework of the Dynamo project.

The tools/technologies used in Dynamo were Prosa/ OMT, C, QNX and own middleware. The following experiences were obtained from Dynamo:

- installing of the Visibroker was easy and there were no problems in establishing connections between client and server when both ends were using the same ORB,
- connection between Visibroker on the server side and Orbix on the client side was unsuccessful (Niemelä et al. 1999), and
- porting QNX caused unexpected side effects in Dynamo.

The experiences of Dynamo prove that predicting the success of integration of commercial components can be very difficult due to unexpected side effects.

4.3.3.2 Arttu

The Arttu project (1997–1998) tried to improve the modifiability and flexibility of heterogeneous distributed real-time systems. The functionality of a commercial adapter in the integration of applications between a commercial Objectivity database and Orbix was tested in the Arttu project (Tuominen et al. 1998). Figure 9 presents the framework of the Arttu project.

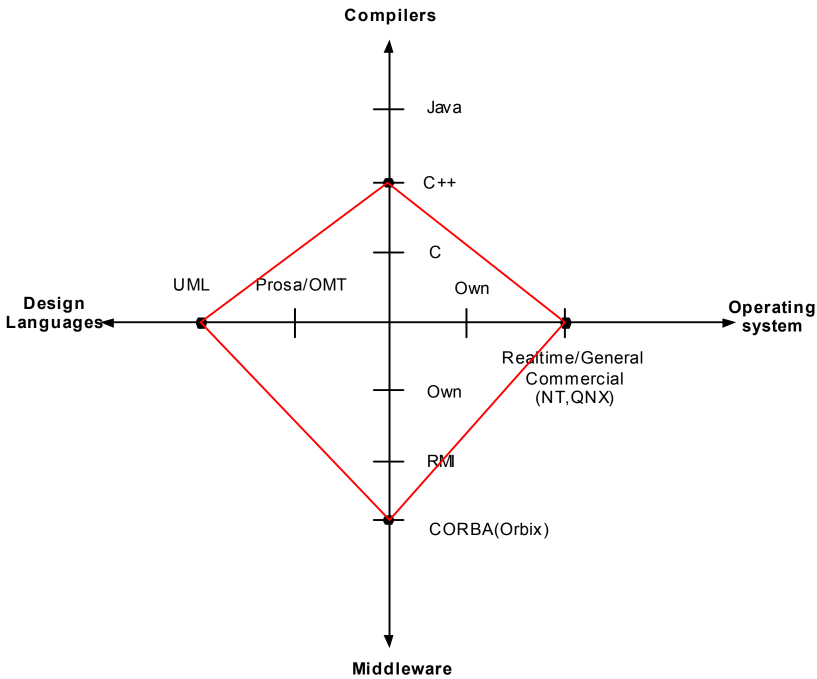


Figure 9. Framework of the Arttu-project.

The tools used in Arttu were UML, C++, Windows NT OS and CORBA. Comparison of Figure 9 with Figure 8 proves that the evolution of commercial component use has already gone into a more general direction. The following experience was gained from the Arttu project:

- studying MFC (Microsoft Foundation Classes) required lot of time, even from experienced people; acquiring a basic knowledge of the use of MFC takes at least six months and if UI requires more developed features, mastering the whole system possibly takes several years,
- the use of CORBA products, like Orbix, is easy for an experienced programmer, although it is good for managing the basic knowledge of information networks,
- the amount and quality of the documentation concerning implementation of TCP/IP (Transmission Control Protocol/ IP) was weak,
- the constraints of the COTS were not known before their implementation due to inadequate documentation of the COTS,
- unsolved compatibility problems between the components still existed, and
- the delayed delivery of a missing integration part caused problems.

The Arttu experiences prove that commercial components still suffer from quality problems and studying the new tools may considerably delay implementation of the system.

4.3.3.3 VHE

The VHE project (2000–2002) studied how a networked platform for home appliances, external services and mobile and stationary terminals could be built in order to support user-oriented services through a variety of interfaces. VHE carried out research into several technologies, so describing one overall framework for COTS use in the project is not possible. Figure 10 presents a framework for the middleware services of the project. The tools used in the VHE middleware were UML, Java and RMI. The VHE middleware allows users to do the following tasks:

1. build a home network by switching on and, optionally, plugging into a wired network,
2. control in-home appliances with terminals having different kinds of UIs,
3. access networked home appliances from the outside world through public networks, and
4. personalise in-home and external services.

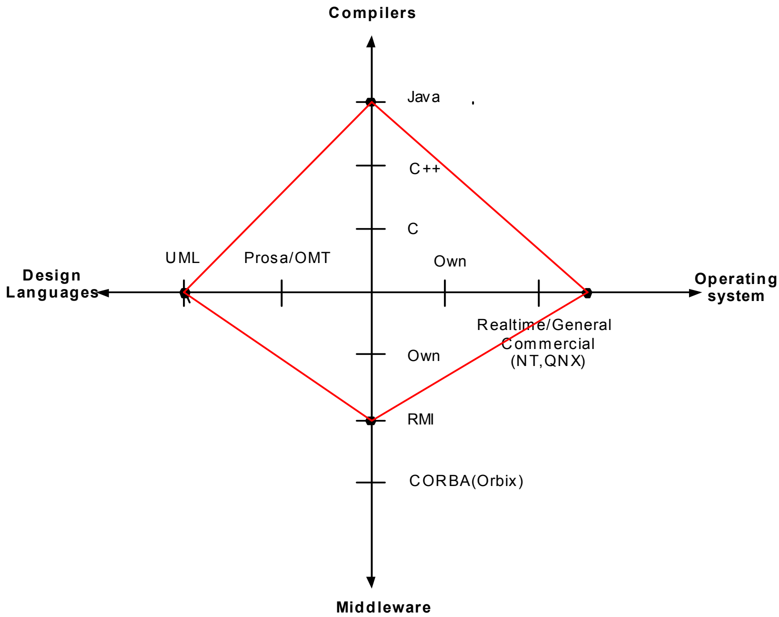


Figure 10. Framework of the middleware services in the VHE project.

The middleware services of the VHE project include application server, directory and distribution service, and media gateway. The services were developed for several OSs and, therefore, most significant in the software architecture is the axis "Operating system". A comparison of Figure 10 with Figures 8 and 9 proves that use of commercial components had gone into a more general direction. Only the use of RMI as middleware diminishes the architectural significance of the commercial components used in VHE.

4.3.4 Realised risk factors of commercial software components

This section summarizes some problems that may occur when developing wireless services by using commercial components. The problems presented here, related to the research question 3 (p. 18), prove that many of the development problems are caused by the involvement of several business actors and technical environments – not the components themselves.

Table 11 describes the realised risk factors of commercial software component use in Arttu, Dynamo and VHE. The realisation of the risk factors was evaluated by studying the project documentation and by sending the project managers an empty risk factor sheet that they were supposed to fill in. The risk factors have been collected from the literature concerning commercial software use (Brereton & Budgen 2000, Shaffer & McPherson 2001, Vigder 1998 and Vigder & Dean 2000).

Table 10 only includes the existence of the risk – not its importance. The risk factors are classified under an appropriate topic. Table 10 shows that in the Dynamo, Arttu and VHE projects there have been problems with:

- the quality practices in component development – section 4.3 presents a documentation template for improving the quality of the component documentation of commercial components –,
- functionality, especially concerning the interoperability of the component ,
- maintenance, especially concerning AP support and stability,
- security, and
- underestimated total costs of the component use. The original list of economic risks in Paper V (p.6) was longer, but they are not included in Table 11 due to the non-existence of the risks in case projects.

Table 11. Realised risk factors of commercial component use in the case projects.

	Dynamo	Arttu	VHE
Technical risks			
Quality practices of component development			
Different quality practices of different APs	X	X	X
Inadequate and/or wrong design, testing, validation and documentation of component	X	X	X
Publicly unknown terminology of the documentation	X	X	X
Functionality			
Changing standards			In non-IP-based commu-nication
COTS product obsolescence affects systems in different ways			X
COTS products are developed to commercial standards			X
Domain/ architecture incompatibility		in adapters	
Inadequate component functionality	Update	Update	Update
Incompatibility of the component with the system	X		X
Inadequate knowledge about system functionality and integration requirements			X
Interoperability with the system and multiple components	X	X	X
Non-portability to other environments			X
Unexpected side effects	in DB (database)	in DB	X
Maintenance			
Compatibility of the new versions with different functions		in mid-dleware	
Rapid changes of COTS products due to product changes by the AP			Market do not exist
Time-limited support of AP	in DB		
Unsecured update			in OSGi
AP support and stability cause risks for maintenance	X	X	X
Security			
Information security susceptibility due to COTS product interoperability			X
Unsecured delivery channels			X
Economic risks			
Underestimated total costs	X	X	Not estimated

Table 11 proves that using more adaptable tools has not removed or diminished the risk in commercial software component use during the six-year research period. The basic problems of component use have also remained the same, although some of the problems are partly due to the emergence of even more complicated development environments and technologies.

4.4 Component documentation

Although software components are not yet documented in a standard way, some attempts have been made to create general documentation templates. These are presented in the following.

The requirement specification standard of the European Space Agency (ESA 1995) classifies software requirements into functional and non-functional requirements. Functional requirements should be organised top-down and non-functional requirements should be attached to them, and can, therefore, appear at all levels in the hierarchy. The standard defines the requirements that should be considered in the user requirements definition phase, such as functional, performance, security, interface, operational, quality and safety requirements. Thus the requirements are determined by the intended use of the software, i.e. when developing the software for a specific purpose.

Sametinger (1997) classifies software documentation into parts according to a purpose: user documentation, system documentation, process documentation and reuse manual. User documentation provides the information necessary for users to use the software system. System documentation provides the development information of a software system or component, such as requirements and implementation details. Process documentation describes the dynamic process of the creation of the component. The last part, the reuse manual, includes a coarse-grained functional specification and information that gives guidance on how to use, manage and assess software. The purpose of the reuse manual is to assure the software engineers that the components of the system fit their needs.

The standard defined by the Institute of Electrical and Electronics Engineers (IEEE 2000) establishes a conceptual framework for architectural descriptions and defines the content of an architectural description. According to IEEE

(2000), an architectural description shall contain the following information: document identification, version and overview (issuing organization, change history, summary, scope, context, glossary and references), stakeholders, and architectural views and viewpoints. The user organization has to define the more detailed content of the description.

Forsell and Päivärinta (2002) define a documentation model of a software component that is divided into two parts: the reusable part and a part that supports reuse. The reusable part defines the objectives and design rationale for the component, the results of the production work and the test procedures used to certify that the component works correctly. The reuse support consists of maintenance information, consumption information, and management information. The idea is to map the roles of a component provider, a maintainer, a software developer and a reuse manager to all the pieces of information produced during the development and use of a component. Therefore, the model is only appropriate within the context of a reuse process inside an organisation.

The general documentation templates presented above have similarities with the template of this research (for more detail, see Papers III and IV), but the latter also takes into account the multi-organizational stakeholders of the documentation process and the architectural viewpoints. The following section tries to answer research question 4 (p. 18) by illustrating the component documentation of wireless services.

4.4.1 Component documentation template

This section presents a documentation template for commercial components that has been developed while examining how third-party components are developed and used in the context of a product line that is a collection of software systems with similar or overlapping functionality and a common architecture that satisfies the requirements for that set of systems (NPLACE 2002) by the component integrator (Seppänen et al. 2001). The information in the component documentation is divided into four categories that are derived from the literature (Wallace et al. 1998, Niemelä & Ihme, 2001, Niemelä et al. 2000) and refined further (Forsell & Päivärinta 2002, Salmela et al. 1999). Figure 10 shows the basic structure of the component document (for more details, see Paper IV, p. 11).

1. Basic information

1.1 General information

- 1.1.1 Identification
- 1.1.2 Type
- 1.1.3 Overview
- 1.1.4 History
- 1.1.5 Special terms and rules

1.2 Interfaces

- 1.2.1 Version usage
- 1.2.2 Required interfaces
- 1.2.3 Provided interfaces

1.3 Configuration and composition

1.4 Constraints

- 1.4.1 Protocols
- 1.4.2 Standards

1.5 Functionality

1.6 Quality attributes

- 1.6.1 Modifiability
- 1.6.2 Expandability
- 1.6.3 Performance
- 1.6.4 Security
- 1.6.5 Reliability

2. Detailed information

2.1 Technical details

- 2.1.1 Application area
- 2.1.2 Development environment
- 2.1.3 Platforms
- 2.1.4 Interdependencies
- 2.1.5 Prerequisites
- 2.1.6 Special physical resource needs

2.2 Restrictions

2.3 Implementation

- 2.3.1 Composition
- 2.3.2 Context
- 2.3.3 Configuration
- 2.3.4 Interface implementation

2.4 Delivery

3. Acceptance information

3.1 Test criteria

3.2 Test overview

3.3 Test environment

3.4 Test cases

3.5 Test summary

3.6 Test support

4. Support information

4.1 Installation guide

4.2 Customer support

4.3 Tailoring support

Figure 11. Basic structure of the component document.

The documentation template presented in Figure 11 has been developed because the current component documentation models:

- are not appropriate for third-party components because they mainly concentrate on component reuse inside an organisation, and
- do not consider the possible multi-organisational development of a component and the architectural properties of a component to be important.

The presented documentation template has the following advantages for both component providers and integrators:

- it provides clear guidelines for component providers on how to document the properties of a component, and thus guarantees the consistency and quality of the component documents, and
- it assists integrators of components in selecting, validating, using, modifying and maintaining the component by providing information about the component's design and implementation, as well as customer and tailoring support.

The fields of the documentation template presented in Figure 10 are described in more detail in the following section. The fields presented in the template can be applied to all kinds of domains, including wireless services.

4.4.1.1 Fields of component documentation template

This section describes the fields of the documentation template: basic, detailed, acceptance and support information.

The *basic information* has six parts: general information, interfaces, configuration and composition, constraints, functionality, and quality attributes. The general information gives an overview of a component – that is, what the component is and why it is – with the following pieces of information:

- identification of the component,
- type of component,
- overview of the component,
- history of the component, and
- special terms and rules used.

Component identification separates the component from other components, which is compulsory for tracking the components in a networked environment. A component type expresses the way the component is intended to be used. The overview of a component is needed in order to understand why the component is developed or purchased. The history describes the life cycle of the component, including the initial version and modifications. The design rationale may identify the architectural choices and the reasons behind the design decisions (Matinlassi et al. 2002). The interface information describes interface versions and the provided and required interfaces of a component (Bosch 2000, Clements & Northrop 2001). The forms of configuration and composition define how the behaviour of the component can be changed and how the component is included in a service. Functional and object-oriented compositions are based on the activation of the components by function calls. In object-oriented composition, different components are combined through polymorphism and dynamic binding. Constraints on components are usually common constraints for all the components in a service, both those made in-house and by third parties. Constraints are here defined as including the protocols and standards used. Protocols define the global behaviour of a component, describing the interaction between two components needed to achieve a specific objective. The functional specification describes the inputs and outputs of the component, and the functionality of the component, in detail. The required data is described, together with functional exceptions, if the architecture requires their being handled in a specific way; otherwise, they have to be defined in the technical details. Quality in this context means the quality attributes the component embodies.

The *detailed information* has four parts: technical details, restrictions, implementation and delivery. The technical details define the application area in which the component is intended for use, as well as the development, execution and composition environment of the component. In order to select a component, the architect needs information about the physical resources required for the execution of the component, as well as the interdependencies of the component with other components. The prerequisites define all the other requirements that component may have to fulfil. In the design phase, the restrictions should describe all items that will limit the developer's options when building the components. The restrictions described in the detailed information are related to the quality attributes defined in the basic information, but they are based on prior metrics collected during the development of the component. The rationale is to give detailed information or proof of how the required qualities are met. The implementation information includes details about the composition, context, configuration and interface implementation. The composition information describes the internal structure of the component. The context describes the things that exist or events that may transpire in the environment. The interface implementation gives the details of the interfaces. The link to the realisation, the code of the component, is also given in this part of the component document. The delivery information provides details of the format of the component, which can be, e.g., binary format, library format, or source code. The delivery time of the component may be useful for tracking versions when the supplier adapts the component.

The purpose of the validation of a component is to prove the quality of the component. The *acceptance test information* on the component should describe:

- the test criteria the component has to meet,
- the test methods, in the form of guidelines,
- the test cases used in testing,
- a summary of the test results,
- the test environment in which the component has been tested, and
- test support for an easy evaluation of the component.

The scope of the acceptance test should be restricted. The selection of the test criteria for the component is guided by how well the requirements specified for the component have been met. Therefore, the test criteria should be selected during the component specification – for example, by blackbox tests that the component should pass (Salmela et al. 1999).

The user of the component also needs *support information* on how to install the component and what to do if it doesn't work as expected. For these cases, the component document should have an installation guide, and contact details for the customer support service and tailoring support. This information is essential when using commercial, third-party components made by someone else. The necessity of the support services and guides depends on the size and complexity of the component, as well as the form in which it has been delivered. When delivering the component as binary, installation and configuration information is necessary. In the case of source code, tailoring support is needed if the component requires adaptation to the service in question.

4.5 Component-based development of wireless services

The following articles illustrate the use of component-based development in developing wireless services.

Gschwind et al. (1999) propose a framework whereby the agent program is composed using a well-defined set of categories of software components: navigational components, performers and reporters that determine how the results are collected and reported back. In their approach to distributed network and systems management mobile agents, the management actions to be performed were relatively simple, the itinerary of an agent was well defined and the global operation and side effects of the algorithm were well understood.

Marques et al. (2000) present a component-based framework that enables applications to use mobile agents in an easy and flexible way. By using this approach, the development of applications that can make use of mobile agents is greatly simplified, as applications can be developed using current object-oriented approaches and become capable of sending and receiving agents by the simple drag-and-drop of mobility components.

Yang et al. (2002) describe the development of a wireless embedded system that uses component-based software. Their paper introduces utilizing J2ME technology for developing a mobile enterprise environment based on design patterns and component software. The design patterns used in their research were property container, strategy, decorator and model-view controller.

The following sections present the component-based development of two wireless case services – accounting and billing service, and secured VHE – by integrating VPN and OSGi. The following sections seek an answer to research question 5 by illustrating the aspects to be taken into account in the component-based development of wirelesss services.

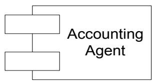
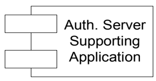
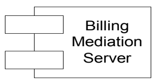
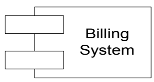
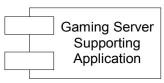
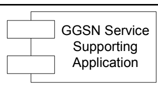
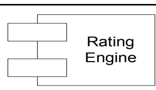
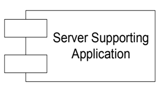
4.5.1 Accounting and billing service in case services 1 and 2

Accounting and billing plays an important role in wireless business models as it enables tracking the financial transactions of the business actors. The accounting and billing service presented in this research is based on the technical specifications of 3GPP (2002, 2003a), although the charging principles, and charging and billing, for different kinds of services presented by 3GPP (2002, 2003a) are not within the scope of this research. The following sections will present the implementation of the accounting and billing service to case services 1 and 2 (more detail in Paper VII).

4.5.1.1 Main components of the billing chain

This section describes the major components of the billing chain. It is assumed that all the parties involved in the value chain allow access to their service elements; if this is not feasible, the components that the partners do allow interfacing with in order to collect usage data act as the service elements. Service elements are software and hardware components that provide the service or interact with each other for providing the service. In the presented accounting and billing service the concept 'service element' also means both software and hardware components and the concept 'component' just software components. Table 12 presents the concepts of the accounting and billing service used in Figures 12 and 13.

Table 12. Concepts used in the accounting and billing service.

Element	**Name**	**Definition**
	Accounting agent	Java application that collects raw usage data, converts it into a certain form and provides usage data for the billing mediation server
	Authentication Server Supporting Application	Component for collecting accounting records
	Billing Mediation Server	Component that collects usage data from the accounting agents and sends it in standard format to the rating engine
	Billing System	Component that applies pricing and discounting policies to the received service usage records by generating an invoice to the final customer and generates revenue sharing data to third-party providers
	Gaming Server Supporting Application	Component for collecting value-added service usage data in order to charge for the service per transaction
	GGSN Service Supporting Application	Component for collecting usage data in order to evaluate the network service usage in terms of data volume/time
	Rating Engine	Component that receives all the usage detail records from the billing mediation server and correlates them in an end-to-end service generating records related to the entire service to be billed
	Server Supporting Application	Component of AP, CP or SP for collecting value-added service usage data in order to charge for the service per transaction and per content
	Accounting records	Records aggregated to bill the final customer, defining incoming revenue of the NO and revenue sharing fees for the AP and CP
	Revenue sharing record	Record indicating the division of monetary share every partner receives from the total revenue
	Service usage record	Record indicating how much users have used a particular service

The main components of the service – accounting agent, billing mediation server, rating engine and billing system – are explained in more detail in the following.

Accounting agents are attached to all the relevant service elements in order to collect raw usage data. The accounting agent needs to interface with a variety of usage data sources, provide collected data encoded into different output formats and be extensible to support new data source types and output formats. Accounting systems are in charge of measuring or collecting the usage of any type of resources and they could charge the end user for using network resources (e.g. a GPRS connection, but also a PPP (Point to Point Protocol) connection on a wired line) for browsing web pages, etc. The accounting agent provides the following functions:

- Acts as an interface for the service elements and for the billing mediation server to provide usage detail records. Accounting agents basically interact with service elements to collect usage data. In real-time billing scenarios they also grant or deny the access to a service (or interrupt the service), so the service element in charge of serving the end user will suspend the call, ask the accounting agent if the user "has enough money", wait for the answer and, if the answer is positive, continue the call.
- Converts raw usage data from multiple source data formats to a set of standard records,
- filters, validates and aggregates raw usage data extracted from the source service elements,
- translates usage detail records into a common XML (eXtensible Markup Language) syntax, and
- provides feedback to the service elements in a prepaid or service usage check.

The *billing mediation server* collects all the usage detail records from the accounting agents distributed over the network. Due to the valuable nature of the data, a secure transfer mechanism is needed in order to guarantee the security, integrity and confidentiality of the communication infrastructure. The accounting agent and the billing mediation server communicate by means of FTP (File Transfer Protocol), RMI or JMS (Java Message Service). All these are based on TCP/IP. In a real scenario TCP/IP is not applicable as it is because it is not secure. The security could be solved by offering a secure transport layer and by using objects that recognise each other by using private/ public keys or digital signatures.

The *rating engine* receives all the usage detail records from the billing mediation server in standard formats and correlates them in an end-to-end service generating records related to the entire service to be billed. The service usage records are correlated to rate customer services according to the subscribed contracts.

The *billing system* applies pricing and discounting policies to the received service usage records by generating an invoice to the final customer. It also generates revenue sharing data to third-party providers like CPs, APs and SPs. The billing function takes account of the pricing and discounting schemes configured in the billing system.

The billing, rating and service nodes of AP's, CP's and NO' are physical nodes that represent the service processing.

4.5.1.2 Accounting and billing in case service 1

The end-user of case service 1 is subscribed to the service as a bank customer and does not actually see the other actors behind the bank that really provide the service, since the service connection is carried out transparently. In the case of service 1, the bank delivers the invoice to the customer and receives the fee, part of which is afterwards paid to the other actors for revenue sharing. In case service 1 the accounting agents can be attached to:

- GPRS backbone network for collecting usage data,
- authentication server for collecting accounting records and servers of CPs, NO and SPs for collecting service usage data, and
- rating engine and billing mediation server for collecting the usage data and correlating the usage records.

After all the collection, filtering/validation, correlation and rating phases, the service accounting records are aggregated to define the revenue sharing fees for the AP, CP and NO. Figure 12 presents the accounting and billing context of case service 1 and the mapping of the component-development responsibilities to different business actors.

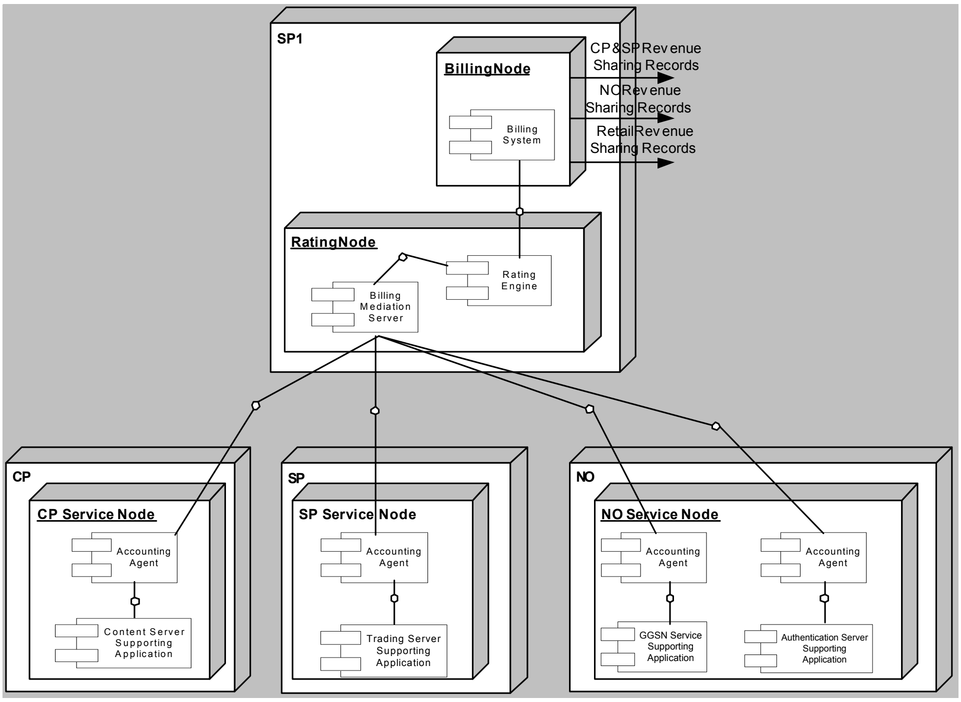


Figure 12. Accounting and billing context of case service 1.

In Figure 12 the usage data is collected from the accounting agents of AP and NO by the billing mediation server of the SP1. The rating engine of the SP1 correlates the usage detail records by generating records related to the entire service. The billing system exports the AP, CPs and NO revenue sharing records and retail service usage records to the service retailers.

4.5.1.3 Accounting and billing in case service 2

In case service 2 the SP1 delivers the invoice to the customer and receives the fees for delivery to revenue sharing. The billing of case service 2 takes place according to the number of bits transferred, the character chosen by the player, and the information requested during the game. In case service 2 the accounting agents could be attached to:

- GPRS backbone network,

- authentication server, gaming server for collecting service usage data in order to charge for the service per-transaction, and servers of APs and NO, and
- rating engine and billing mediation server for collecting the usage data and correlating the usage records.

Figure 13 presents the billing context of case service 2 and the mapping of the responsibilities of component development to different business actors.

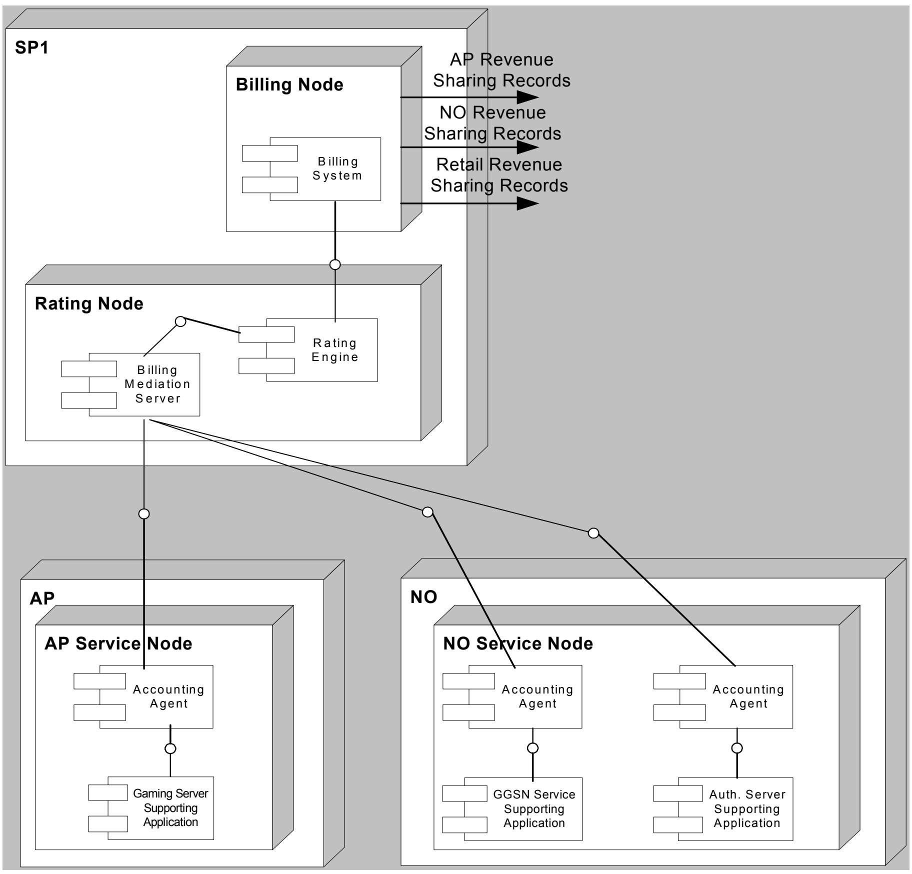


Figure 13. Billing context of case service 2.

In Figure 13 the accounting agents of AP and NO collect the usage data that the billing mediation server of the SP1 collects from the agents. The rating engine of the SP1 correlates the usage detail records. The billing system exports the AP and NO revenue sharing records, and the retail service usage records to the retailers of the service.

4.5.2 Case service 3: integration of VPN and OSGi

This section discusses the integration of two commercial components, VPN and OSGi, to create a secured VHE that consists of home devices that are networked to enable their flexible use and is able to use appropriate services remotely over untrusted public networks and to download applications and install services (for more detail, see Paper VIII).

The OSGi architecture shown in Figure 14 is composed of two key components: OSGi framework and bundles.

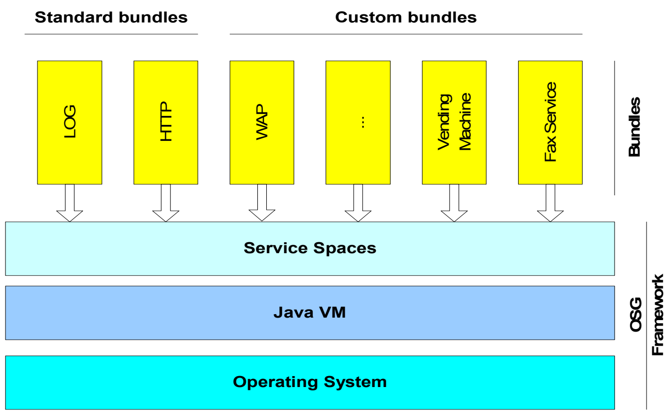


Figure 14. OSGi architecture (modified from Joong-Han et al. 2001).

OSGi was founded to create an open specification for the network delivery of multiple managed services over a WAN (Wireless Area Network) to local networks and devices (OSGi 2003). The OSGi services gateway is technically an embedded server that is attached to the WAN to connect external SPs to internal clients and separate the topology into the external and internal network. The OSGi services gateway functions as the platform for many kinds of communications to and from the home, office and other locations, and it can also function as an application server for a range of high-value services, such as security. The

OSGi framework allows the partition of applications into smaller components called bundles and provides a consistent programming model during application development (Joong-Han et al. 2001). The LOG service of the standard bundles offers other bundles the facility to log important information to the gateway operator. Custom bundles that vary according to the application (Figure 14) include, for example, vending machine, fax service, WAP (Wireless Application Protocol), etc.

VPN is a virtual channel that connects two private networks and enables the transmission of data across different platforms without concerns about the equipment and its associated protocols. VPN allows the user to tunnel through the Internet, or another public network, in a manner that provides the same security features formerly only available in private networks. The VPN connection is only set up on an as-needed basis and broken when the transaction is completed, and this is why the private network is considered virtual. VPN technology is designed to address issues surrounding the current business trend towards increased telecommunications and widely distributed global operations, where workers must be able to connect to central resources and communicate with each other. In order to provide the ability to connect to home computing resources, a corporation must deploy a scalable remote access service (Lerner et al. 2000).

4.5.2.1 Integration of VPN and OSGi

In case service 4 the OSGi gateway was integrated with the VPN system so that the VPN server was replaced by the OSGi server. Figure 15 illustrates the high-level architecture or VPN and OSGi integration.

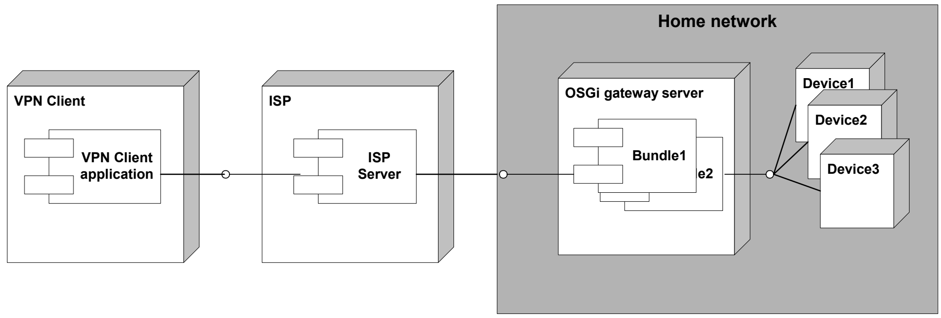


Figure 15. Integration of VPN and OSGi.

The elements of Figure 15 are described in Table 13. The VPN client accesses the Internet by contacting the ISP. After this, a connection to the OSGi home gateway is established through the Internet. The OSGi gateway manages all the devices and services in the home network. The wireless VPN client is used to view the information in the gateway, modify configuration information and receive notifications in a secure manner. In order to support authentication, authorization and secure communication, the OSGi gateway acts as a VPN Server. After authentication and authorization have been done, the VPN Client and OSGi gateway use a secure channel for communicating to prevent different kinds of attacks against confidentiality.

Table 13. Elements of VPN and OSGi integration.

Element	**Description**
Bundle	Component of OSGi gateway server
Device	Physical device in home network
ISP Server	Component that enables VPN clients to access the Internet
OSGi gateway server	Embedded server that provides security services and functions as a platform for various communications
VPN client	Wireless end-user terminal where VPN client application runs
VPN client application	Component that encrypts all data from the client to a gateway or server

This integration of VPN and OSGi proved that their integration is suitable for creating a secure VHE. The integration of OSGi and VPN turned out to be a relatively easy task and improved the security of the VHE, but its suitability for the home is questionable because of the need for a technically competent administrator who is able to install the system and is aware of possible security risks. It is possible that these kinds of security applications will be embedded in home gateway devices and the homeowner will buy administrative services along with the gateway device. Case service 4 also proved that although integrating two commercial components to create a wireless service is easy, the need for technically competent users in the implementation of a service can prevent more widespread use of the service (cf. Mobile Payment Forum 2002).

4.6 Comparison with related work and conclusions

This section compares the related work with the related literature and draws conclusions about component-based development and component documentation in wireless services.

4.6.1 Evolution and risks of commercial component use

Table 14 compares the evolution of component use found in this research with that found by Rumbaugh (1987), Szyperski (1998) and Vigder & Dean (2000).

Table 14. Evolution of component use.

Rumbaugh (1987)	**Szyperski (1998)**	**Vigder & Dean (2000)**	**This research**
• object-oriented languages have evol-ved over time as the standard languages have been inadequate	• object-oriented programming is expanding alongside component-oriented programming • component frameworks are appearing on the market • architecture integrating component system is missing	• elements attched to connection infrastructure, environment, collaborations, interfaces, control mechanism and interconnection topology causes dependencies between COTS • management of component dependencies enables evolution of COTS-based system	• significance of commercial software components in the software architectures has increased • many functional and quality problems. such as weak documentation and interoperability problems, still make the use of commercial software components difficult

Analysis of Table 14 proves that the use of software components is moving into a general direction. Management of component dependencies would enable evolution of COTS-based systems, but dependencies between components, the lack of integration methods and many functional and quality problems still make the use of commercial components difficult. This research proves that the components' significance in the architecture has increased by shifting towards the use of more portable and configurable components.

Table 15 below compares the main risks of commercial component use found in this research with the research of Garlan et al. (1995), Samtinger (1997) and Morisio et al. (2000).

Table 15. Risks of commercial component use.

Garlan et al. (1995)	**Sametinger (1997)**	**Morisio et al. (2000)**	**This research**
• excessive code • poor performance • need to modify external packages • need to reinvent existing functions • unneccesarily complicated tool • error-prone construction process	• adaptation • integration • certification • blackbox character of components • lack of abstraction of reusable components	• delays in releasing the COTS • documentation on the product is inadequate • some functions were promised but never implemented • modifications made by the AP alter the compatibility of COTS with other COTS, or the system, or introduce new bugs • communication with the AP one way • integration problems	• different quality practices • design, testing, validation and documentation of the component is inadequate and/or wrong • inadequate functionality and unexpected side effects • non-portability to other environment • system, domain/ architecture and version incompatibility • support and stability of the AP • security • underestimated total costs

Comparison of the risks of commercial component use in Table 15 proves that most problems in commercial component use are due to changes of AP and a lack of support, integrability, interoperability and modification of components. Table 15 shows that the problems for commercial component use have evolved from the risks attached to the component itself (code) and the integration problems caused by weak documentation and AP support.

4.6.2 Component documentation

Table 16 presents the documentation models of this research, IEEE (2000), ESA (1995), Sametinger (1997) and Forsell & Päivärinta (2002).

Table 16. Comparison of the component documentation models.

ESA (1995)	Sametinger (1997)	IEEE (2000)	Forsell & Päivärinta (2002)	This research
Contents:				
• introduction (purpose, scope, definitions, references and overview) • general description (relation to other projects and systems, function and purpose, general constraints, model description) • specific requirements • requirements traceability matrix	• general information • reuse information (installation, interface descriptions, integration and usage, adaptation) • administrative information • evaluation information (known performance problems, test support, etc.) • other information	• document identification, version and overview (issuing organization, change history, summary, scope, context, glossary and references) • stakeholders • architectural views and viewpoints. The user organization has to define the more detailed description of the content.	Reusable Part • objectives and design rationale for the component, result of the work, test procedures Part Supporting Reuse • brokering information (information about modifying a component) • component's consumption information • management information	• basic information (general information, interfaces, configuration and composition, constraints, functionality, quality attributes) • detailed information (technical details, restrictions, implementation, delivery) • acceptance information • support information
To whom:				
• "all active participants in the software requirements phase"	• "the software engineers who decide whether a certain component fits their needs"	• Users, operators, acquirrers, architects, deliverers, maintainers, evaluators of the system	• "primarily for the reusers, but also for the brokers and reuse management. "	• All stakeholders
Difference to this research:				
• Less information about the component (only includes the requirements)	• Meant for technically competent users • Less information about the component	• User of the document has to define the content in more detail ⇒ templates are not uniform • Less information about the component	• Meant for reusers • Less information about the component	

The comparison of the template of this research with the component documentation models presented in Table 16 reveals that the template of this research provides more information on several aspects of the components and takes more account of the stakeholders' multi-organizational development.

4.6.3 Component-based development of wireless services

Table 17 compares the experiences received from the component-based development of wireless services in this research with the articles by Gschwind et al. (1999), Marques et al. (2000) and Yang et al. (2002).

Table 17. Experiences from component-based development of wireless services.

	Gschwind et al. (1999)	**Marques et al. (2000)**	**Yang et al. (2002)**	**This research**
Service	Mobile agent application	Application-centric mobile agent system	Mobile enterprise environment	Accounting and billing service, VHE
Advantages of component-based development	Flexibility, reduced development time, improved reliability	Flexibility, easy software development, fault-isolation, seamless integration, user acceptance	Improved reusability, platform independence	Easiness, reduced development time
Disadvantages and risks of component-based development	Scalability	Increased security threat without proper authentication mechanism	Some components need modification, deve-lopers have to identify variation points of components	Need for technically competent user may prevent more wide-spread use, components need modification

The comparison of this research with the related literature in Table 17 reveals that the basic advantages (like reduced development time) of the component-based development in the related literature and this reseach are the same. The disadvantages of component-based development vary according to the domain. The need for technically competent users experienced in this research is a potential prohibiting factor for component-based development in other domains as well.

4.7 Conclusions

The case studies illustrated in this chapter proved that component-based development suits the development of wireless services and it can be quick and easy, but gaining full advantage from it requires overcoming the existing interoperability and quality problems. In some services the high level of technical competence required can prevent widespread use of the service. The developed services should be easy to use, secure, portable and reliable, and the componentized application used for realizing the service should cause no side effects.

The three case projects presented indicate that the use of software components is evolving into a more portable and configurable direction. To increase the offer of commercial components, the software business should increase the interoperability of the components, get rid of the "made here" principle and offer their components commercially, as well as use components made by the others. Use of open standards, interfaces and development environments should be increased to reduce the technology-dependency of the components.

For improving the component documentation, this chapter also presented a documentation template for commercial components that takes into account all business actors and architectural aspects of the service. The presented documentation template provides clear guidelines on documenting components. The documentation that follows the template assists component integrators in selecting, validating, using and maintaining the component.

5. Introduction to the papers

The papers included in this thesis were written over a period of three years between 2001–2004. They illustrate the different aspects of wireless services, from business models, roles and networks to component-based development of wireless services. Table 18 summarizes the papers.

Table18. Summary of the issues presented in the papers.

Paper ⇒ Issue ⇓	**I, II**	**III, IV, V**	**VI, VII, VIII**
Domain	Business models, networks and roles	Component-based application development	Case studies
Contribution	Describes the business actors and their roles and interaction in wireless business networks.	Describes the use of commercial components and component documentation in component-based application development.	Describes the creation of three case-services to validate the ideas presented in papers I–VI

5.1 Business roles and networks

5.1.1 Paper I: Application service provisioning – current state and partnership strategies

This paper, which is connected to the usage/provisioning phase of the service life cycle, explores the state of the software business, application service provisioning and the ability of Finnish software companies to gain partners in the international application service provisioning competition based on the situation in November 2000. The state of the application service provisioning business is examined by researching literature and four Finnish software companies that use or are going to use application service provisioning in the future. The state of software companies in this paper is based on a study made by Forsell and Nukari (1999) and on an international research of 100 software firms made by Harvard Business School (Hoch et al. 1999). This paper finds that:

- Finnish ASPs seek value-added service and technology partners on the basis of their quality and trustworthiness, and regard finding competent partners difficult because of the immature market,
- software companies co-operate in marketing, support and research,
- the application service provisioning market is rapidly changing and growing, and is immature and uncertain, but faith in it is very strong,
- technology and security do not prevent the widespread use of application service provisioning, and
- the keys to success in the application service provisioning market are co-operation with qualified partners, specializing in existing strengths, and qualitative and secure software products.

As the amount of related literature on application service provisioning is rather limited, this paper serves as a good summary of the state of the art in this topic. The summary of the interviews in this paper requires some tough criticism due to the small number of case companies. At the time of writing this thesis (2004), application service provisioning is still not very succesfull in the four companies researched and their income mainly comes from traditional software development. The state of the application service provisioning described in this paper serves as an input to Section 3.2.3 of this thesis.

5.1.2 Paper II: Business Models in Wireless Internet Service Engineering

This paper, which is connected to development, deployment and usage/provisioning phases of service life cycle, proposes a general framework for the business model of 3G wireless services and analyses the state of the wireless Internet services. This paper also describes the interaction of business actors with the aid of three case companies acting in the wireless business. This paper draws the following conclusions from the researched companies:

- their business strategy is to sell products the customers want and tailor them according to their needs,

- they interact with third parties that provide them with bandwidth, content and the technology they need, and
- they have chosen their role in the wireless business based on focusing, cost savings, time to market, risk reduction and macroeconomic facts.

Compared with the related literature, this paper provides a novel framework for business models of wireless companies. The limitation of the framework is its domain – and situation –specificity, which should be taken account when evaluating its suitability for the business in question. The emergence of new business roles and actors will require redefining of the framework in the future. The coarse-grained business models of the two wireless services presented in this paper are used as the input for the more detailed analysis of business models in Chapter 3.3.1.

5.2 Component-based application development

5.2.1 Paper III: Documented quality of COTS and OCM components

This paper illustrates some aspects of component documentation and the stakeholders participating in making and using the document. This paper illustrates the problems encountered when COTS and OCM components are applied to software systems. This paper also provides a general template for documenting software components in a standard way, whereby the component buyer and developer's views are taken into account. This paper proposes that:

- Although there is a need to utilise COTS and OCM components, there are still barriers that prevent them from getting full benefit from the software component business.
- As a means to qualify a component, its documentation is a fundamental factor.
- In order to assure reusers that the quality of a component is in conformance with the quality of the product or products the component will be used in, thorough documentation is the only way. That is why a standard documentation template is needed for software components, especially for COTS and OCM components.

Although components always have to be documented, a standardized model for the documentation of commercial components does not yet exist. Thus this paper presents a generalizable documentation template. The presented documentation template – compared with the related literature – takes better account of component development in a multi-organizational environment. The template presented in this paper is coarse-grained and therefore lacks a more detailed specification of the fields. A more detailed definition and validation of the template is presented in Paper IV. Papers III and IV are connected to all phases of service life cycle.

5.2.2 Paper IV: Component Documentation – a key issue in software product lines

This paper presents a standard documentation template for components and an integrated documentation system, which is validated by an experiment. The developed documentation template defines the information content and the structure of the documentation. The template is divided into four logically separate parts: basic information, detailed information, acceptance information and support information. The presented documentation template has advantages for both component providers and integrators. Validation of the documentation template shows that:

- component documentation work is easy when following the template,
- it is possible to produce documentation data easily while developing the component, and
- use of the template requires a documentation system that provides the required tools and technology for documentation and ensures that the documentation is in accordance with the template.

Compared with the related literature, the presented documentation template provides more information on several aspects of the components – such as quality, testing and architectural properties – than recent component documentation models and takes more account of the stakeholders of multi-organizational development. A limitation of the template is the resistance its implementation may encounter in companies due to existing documentation practices. Implementing a new documentation system requires modification of the exisiting documents and, especially for bigger companies, this causes a lot of work. The documentation template presented in this paper is used in Chapter 4.4.1 to prove that qualitative documentation can improve the development of wireless services.

5.2.3 Paper V: Evolution of the use and risks of the Commercial Components and Software

This paper, which is mainly connected to the usage/ provisioning phase of the service life cycle, presents an evaluation of commercial software component use and its significance in software architecture, and encountered risks in three projects in the years 1996–2001. This paper proves that:

- the significance of commercial software components in software architectures has increased,
- many functional and quality-risks, such as interoperability problems still exist,
- the quality of the commercial software component documentation and AP's support is immature, and
- delivery of the components via the Internet has increased the security risks.

This paper proves the general tendency that the use of software components is moving into a general direction due to the deficiences of the current systems, but dependencies between components, a lack of integration methods and many functional and quality problems still make the use of commercial components difficult.

The novel evaluation framework for commercial component use presented in this paper can be applied to all kinds of components. The risks encountered in the case projects are mainly found in the related literature, but this research found additional risks attached to the security and cost of the components. The generalisation of the results in this paper could be improved by increasing the number of case projects. Evaluation of the commercial component use and risks described in this paper serve as input to Section 4.3 of this thesis.

5.3 Case studies

5.3.1 Paper VI: Wireless Internet Service Development

This paper examines the development of a wireless service from the third-party view and proposes use of open standards (like OMA and OSA) and interfaces as one alternative to more effective use of third-party applications. This paper illustrates wireless Internet service development with the aid of a wireless case service – a multiplayer game – that is under development for the 3G wireless networks, and defines different aspects of wireless service development. The development model for wireless services presented in this paper is based on the views of several 3G participants. This paper concludes that:

- in the future, third parties will provide service enablers that are currently provided by telecommunication operators,
- the development of wireless services is moving in the direction of open standards and architectures, and
- the current technology limitations make wireless service development difficult.

This paper complements the related literature by summarizing the main aspects of the wireless service development process. To improve the validity of the results the development process sketched in this paper should be defined in more detail according to the input received from the development of various services. The results of this paper are used to describe the wireless arena and development in general in Chapter 1.

5.3.2 Paper VII: Accounting and Billing of Wireless Internet Services in the Third Generation Networks

This paper, which is connected to the development, deployment and usage/provisioning phases of the service life cycle, presents an accounting and billing model and its architecture applied to two wireless services. This paper concludes that:

- development of the wireless Internet market and its structure is driven by differing industry fundamentals,
- revenue derived from content and content-related services is expected to significantly increase for all actors within the wireless industry,
- providing flexible and scalable accounting and billing systems will be essential for success when offering wireless services to end customers,
- the number of roles and actors in the wireless service business is big and methods for accounting agents and billing are needed for tracking customer transactions and directing the accounting and billing between the actors,
- wireless SPs typically have difficulties in billing their customers due to their inability to associate customer transactions and network usage.

The accounting and billing model presented in this paper provides a more flexible and adaptable architecture with which to account and bill customers than other publicly available alternatives. A limitation of the presented model is that resistance to its implementation may be encountered due to existing accounting and billing models. Applying the application presented in this paper requires adequate knowledge of architetural design. This paper provides input to Chapter 4.4.1 of this thesis by illustrating the component-based development of two wireless services.

5.3.3 Paper VIII: Integrating VPN and OSGi to create a secured Virtual Home Environment

This paper, which is connected to usage/provisioning phase of service life cycle, presents the suitability of VPNs in the prevention of security breaches and creation of a secure home network, and presents the integration of the OSGi platform and VPN to solve the problem of how to make a secured remote connection to access appliances and services within a residential area network. This paper also investigates the use of the technological aspects of VPN middleware, OSGi and VHE. This paper proves that:

- VPN technology enables increased security and cost savings, but requires both an in-depth understanding of the network security issues and a maturing of the market.
- although the integration of OSGi and VPN is a relatively easy task, its suitability for the home is questionable with today's products because of high implementation costs and the need for a technically competent administrator.
- without a technically competent, secure-aware administrator, the advantage of a VPN-based service is negligible.

This paper presents a novel way to create a secure home network using OSGi. Even if the integration of VPN and OSGi in this paper was easy, it should be noted that the integration of two commercial components may encounter problems due to various technical environments and platforms, and for this reason the COTS market is still immature. The integration of two components made in this paper is used in Section 4.4.2 to prove that component-based development is suited to the development of wireless services.

6. Conclusions and further research

This chapter answers the set research questions, examines the limitations of the study and suggests further research topics.

6.1 Answers to research questions

The aim of this thesis was to investigate how wireless services can be developed more efficiently within the context of 3G networks. As answering the question requires several factors being taken into acount, this thesis analysed several aspects of wireless service development. The following sections give more detailed answers to the six research questions.

1. What is the status of the wireless service industry today?

Chapters 1 and 2 drew some conclusions about the status of today's wireless service industry based on a study of the related literature and the views of wireless service actors.

The introduction of wireless 3G services has been delayed in Europe due to a financial crisis in the sector, the lack of an appropriate business model, high prices for frequency bands and loose connections between business actors. The success of wireless services in Japan and South Korea has proved that in shifting to 3G services, NOs play a leading role. In spite of the difficulties in the European market, the belief in the success of the wireless business is strong in the arena and the increased revenue opportunities are believed to provide more turnover for all actors.

There are many opinions about the 3G killer services and some of them are forecasted to be the mobile entertainment services, personalized access to information and corporate networks. Lessons learned from mobile services have proved that young people are often the initial users of the services, simple entertainment is the largest content market initially, and succeeding in the wireless business requires combining attractive terminals with useful content and offering them to users in a ready, easy-to-use service package.

The biggest challenges for the wireless industry are:

- achieving interoperability of the applications and platforms via open standards and tighter interoperability between business actors in Europe,
- taking account of the security and other risks attached to delivering the service requests over the network,
- defining appropriate business models, and
- managing the decentralized, networked value chain.

The wireless terminals present challenges to application designers as they may face a temporary loss of network as they move, have scarce resources, and are required to react to frequent changes in the environment. Design efforts to combat network challenges like limited bandwidth involve delivering the maximum amount of content supported by each terminal, and reducing unnecessary requests over the high-latency wireless link.

2. What kinds of roles and transactions emerge in networked wireless business models?

A business model defines the success of a service, as without its decent conceptualization the principal idea of the service remains unclear. Business networks that should be scalable aim at allowing firms to focus on their core activities by outsourcing other activities. Success in the wireless business requires cooperation with other actors, as stand-alone companies with a profitable business model will remain limited in number in the long-term.

This thesis presented a novel business model framework for two wireless case services (Chapter 3). In the presented case services, the wireless business actors had the following roles in the business networks, which are regarded as the main roles in the wireless business: AP, ASP, CP, NO, SP and terminal manufacturer. Although the concepts used for the business roles vary, the main roles are same in all markets. This research proved that defining a general business model for wireless services is difficult, and, therefore, an appropriate business model for a service should be defined case by case.

The transactions that occur between the business actors are either monetary, product or information flows. Information flows control the monetary and product flows. Use of services is charged for using fixed fees, fees per usage or airtime, etc., and new charging mechanisms are expected to enter the market. For tracking monetary flows between the companies, this research presented an accounting and billing service that provides a flexible architecture and enables monitoring transactions that are based on generating monetary flows (Section 4.4.1).

3. What are the risks involved in the use of commercial components in wireless service development?

Gaining the advantages of component-based development requires a thorough evaluation of its possible risks and overcoming them by proper design of the development process and the responsibilities of the actors involved. The risks and evolution of commercial component use in three case projects were analysed in this research based on responses to a questionnaire sent to the project managers of three case projects (Section 4.2). The analysed case projects used commercial components. The responses to the questionnaire proved that component-based development includes the following risk issues:

- Varying quality of components due to different *quality practices* of the APs.
- *Functionality risks,* such as insufficient or wrong functionality of the component, unexpected side effects, and interoperability problems across time and space.
- *Maintenance risks* caused by, for example, AP support and stability.
- *Security risks* like increased chances of unauthorized system access and chances to access, misuse and disrupt information flow.
- *Economic risks,* such as underestimated total costs that are due to not taking into account that component-based development requires investment in its own infrastructure, methodology, tools, etc.

4. **How can component documentation help the development of wireless services?**

Evaluation of the three case projects (Section 4.3) proved that current documentation of the components is inadequate and causes interoperability problems and extends the service's time to market. Therefore, based on a survey of the related literature and the input of component developers, this research presented a documentation template for commercial components (Section 4.4) that takes into account development in a multi-organizational environment. The presented documentation template provides clear guidelines for component providers on how to document the properties of a component, and thus guarantee the quality of the components. Documentation based on the presented template assists component integrators in selecting, validating, using and maintaining the component. Qualitative component documentation also helps wireless service development by providing tools for evaluating the component quality, required modifications and interoperability in the service.

5. **What has to be taken into account in component-based development of wireless services?**

In recent years, CBSE has generated tremendous interest in numerous industry sectors due to its potential for improving quality, productivity, performance, reliability, flexibility and interoperability. Component-based systems are constructed by integrating large-scale components acquired in-house or commercially from third parties.

An analysis of the three case projects (Section 4.3) proved that the use of commercial components has evolved into a more general direction and availability of more general design and development tools would enable even more effective utilization of component-based development. The development of two wireless services (Section 4.5) proved that component-based development is suited to the development of wireless services, but to achieve the best results the following aspects must be taken into account:

- the effort, costs and risks of component-based development compared with other development methods,

- the technological limitations of networks and terminals,
- the technological knowledge needed when using the service, and
- the ability of business actors to deliver the required component in time.

Case services 1 and 2 have not yet been implemented, but the experiences gained from the development so far have proved that the main problems in service development concern technological aspects, such as the limitations of display dimension and the lack of manufacturer support for various protocols like TCP/IP and J2ME.

6. How can wireless services be developed more efficiently within the context of 3G networks?

The related literature has proposed several ways of improving the effectiveness of wireless service development, but as there is no uniquely best solution for solving the problem, the most reasonable way is to seek an answer to the research question by using several viewpoints – as was done in this thesis. According to this research and related literature, wireless 3G services can be developed more efficiently by:

- using commercial components, open interfaces and platforms and technologies that are best suited to the service in question,
- having precise knowledge of the customers' needs and ways to manage the whole life cycle of the service,
- planning the service and its business properly and qualitatively by taking into account possible risks,
- having succesful combination of content, terminal and business model, and
- focusing on core businesses in the business network and partnering with other actors to outsource non-core businesses.

This research also proved that as combining different viewpoints is difficult, the business and technological aspects of the service development should be considered simultaneously – not separately as at the moment. Efficient development of the services is still not fully possible due to the immature market and proprietory systems.

6.2 Limitations of the study

There are some limitations to this thesis that need to be acknowledged. The first limitation is the limited number of cases studied, which implies that when making broad generalizations of the findings it should be used with care. However, the cases present diverse aspects of wireless service development and serve as good examples of the wireless service arena. The limitation that concerns the evaluation of risks in component use is the non-wirelessness of the two case projects, Arttu and Dynamo, which are distributed systems. But, from the viewpoint of component use, they serve this research as relevant cases. The limitation concerning the business model framework presented in this thesis is its domain- and service- specificity and inadequate revenue logic due to the unclear revenue logic of the business actors. As creating a general framework for all services is impossible, the presented framework serves as a good example of how business models can be concptualized. The limitation that concerns the documentation template presented in this thesis is the uncertainty about its applicability to the industrial development environment. Implementation of the documentation template into industrial components was attempted, but it failed due to resistance from the companies with existing documentation practices. However, the template is based on an extensive survey of the existing literature and the developers of industrial companies who were asked for feedback about it regarded it as functional and applicable in practice.

6.3 Future research topics

This thesis researched the business actors and their roles, and networked component-based development of wireless services. Despite the extensive scope of this research, the dynamic state of the wireless industry leaves much room for further research.

As the implementation area of wireless services is constantly changing, the state of the art of the wireless service industry presented in this research, and the potential business models and roles of the actors, need constant updating via further research and re-evaluation, especially after the implementation of 3G networks and role changes for the network operators.

The case services presented in this research will be implemented in the future to see if they have real business potential and if the defined models work in practice. Based on the developed models, business models for new kinds of wireless services could be defined and the presented accounting and billing service could be implemented in an existing service to track users' transactions. As this thesis has concentrated on the development and deployment phases of the service life cycle, in future it would be interesting to define business models for the usage/provisioning and maintenance phases of the service life cycle.

Although the documentation template presented in this thesis has been validated in some industrial cases, its wider adaptation requires further development with the aid of practioners. Evaluation of the real usefulness of the template requires implementation of the template in a huge number of commercial components and development environments.

This thesis proved that component-based development of services can increase cost effectiveness. As the number of available components, and commercial components in particular, is quite small at present, the real advantages of component-based development need to be researched as the number of available components and their more effective use increases; ways of increasing commercial component use and their quality also needs further investigation. The evaluation of component use could be widened from the presented three projects into a larger number of cases.

References

3GPP, 2002. 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Service aspects; Charging and Billing (Release 5). 3GPP TS 22.115 v.5.2.0 (2002-03). 21 p.

3GPP, 2003a. 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Telecommunication Management; Charging Management; Charging Principles (Release 5). 3GPP TS 32.200 v.5.3.0 (2003-03). 83 p.

3GPP, 2003b. 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Quality of Service (QoS) concept and architecture (Release 6). 3GPP TS 23.107 V6.0.0 (2003-12). 41 p.

Adamopoulos, D., Pavlou, G., Papandreou, C., 2000. Development of New Telecommunications Services in Distributed Platforms. Proceedings of the IEEE International Conference on Communications (ICC '00), New Orleans, USA, Vol. 1, June 2000. Pp. 222–226.

Afuah, A., Tucci, C.L., 2001. Internet Business Models and Strategies. McGraw et Hill: New York, USA. 358 p. ISBN 0-07-239724-1

Anderson, J.C., Narus, J.A., 1999. Business Market Management: Understanding, Creating and Delivering Value. Upper Saddle River, NJ: Prentice Hall. Thousand Oaks, CA: Sage. 430 p. ISBN 0-13-522657-0

American National Standards for telecommunications, 1993. Standard ANSI T1.627-1993 (R1999), B-ISDN-ATM Layer Functionality and Specification.

Avison, D., 2002. Action research – a research approach for cooperative work. The 7th International Conference on Computer Supported Cooperative Work in Design, 2002. Pp. 19–24.

Baldauf, L., Karesto, J., Lovejoy, M., Paija, L., 2001. Global Networking in Wireless Teletechnology Business. Tekes: Helsinki, Finland. Tekes Technology Review 114/2001. 93 p. ISSN 1239-758X, ISBN 952-457-04-0

Bass, L., Clements, P., Kazman, R., 1998. Software Architecture in Practice. Reading, Massaschusetts: Addison–Wesley. 452 p. ISBN 0-201-19930-0

Borland, 2003. Visibroker product information. Available on-line at: http://info.borland.com/techpubs/visibroker/ [22.8.2003]

Bosch, J., 2000. Design and use of software architectures. Adopting and evolving a product line approach. Addison–Wesley: Harlow. 354 p. ISBN 0-201-67494-7

Brereton, P., Budgen, D., 2000. Component-based systems: a classification of issues. IEEE Computer, Vol. 33, Issue 11, Nov. 2000, pp. 54–62

Brown, A.W., Wallnau, K.C., 1998. The current state of CBSE. IEEE Software, Vol. 15, Issue 5, Sept.–Oct. 1998, pp. 37–46.

Campadello, S., Koskimies, O., Raatikainen, K., Helin, H., 2000. Wireless Java RMI. Fourth International Enterprise Distributed Object Computing Conference, 2000, EDOC 2000. Pp. 114–123.

Camponovo, G., Pigneur, Y., 2003. Business Model Analysis Applied to Mobile Business. Proceedings of the 5th International Conference on Enterprise Information Systems (ICEIS 2003).

Capra, L., Emmerich, W., Mascolo, C., 2002. Middleware for Mobile Computing. In: Advanced Lectures in Networking 2002, editors: Gregori, E., Anastasi, G., Basagni, S. Springer Verlag. LNCS 2497.

Cartwright, S. D., Oliver, R. W., 2000. Untangling the value web. Journal of Business Strategy, Vol. 21, No. 1/2000, pp. 22–27.

Clements, P., Northrop, L., 2001. Software Product Lines, Practices and Patterns. Addison–Wesley: New York. 608 p. ISBN 0-201-70332-7

Culnan, M., 1993. How Did They Get My Name? – An Exploratory Investigation of Consumer Attitudes towards Secondary Information Use. MIS Quarterly, 17, 3 (September 1993), pp. 341–363.

Cunningham, J.B., 1997. Case-study principles for different types of cases. Quality & Quantity 31. Kluwer Academic Publishers. Pp. 401–423.

Durlacher, 2000. UMTS Report, an Investment Perspective. 146 p. Available on-line at: http://www.dad.be/library/pdf/durlacher3.pdf [5.3.2004]

Durlacher, 2001. Mobile Commerce Report. 75 p. Available on-line at: http://www.durlacher.com/downloads/mcomreport.pdf [26.8.2003]

Eisenhardt, K.M., 1989. Building theories from case study research. Academy of Management Review 14. Pp. 532–550.

Emmerich, W., 2000. Engineering distributed objects. John Wiley & Sons, Ltd. 390 p. ISBN 0-471-98657-7

ESA, 1995. Guide to the user requirements definition phase. ESA Board for Software Standardisation and Control (BSSC). ESA PSS-05-03 Issue 1 Revision 1. 52 p.

Farlex, 2004. Compiler. Computing dictionary. Available on-line at: http://computing-dictionary.thefreedictonary.com/compiler. [18.3.2004]

Forsell, M., Nukari, J., 1999. Suomen ohjelmistoteollisuuden kasvun strategia ja haasteet. Tekes: Helsinki. Teknologiakatsaus 67/99.

Forsell, M., Päivärinta, T., 2002. A Model for documenting reusable software components. In: Forsell, M., Improving Component Reuse in Software Development. Jyväskylä: Jyväskylä University Printing House. Pp. 117–150.

Funk, J.L., 2004. Mobile Disruption – the Technologies and Applications Driving the Mobile Internet. USA: John Wiley & Sons. 211 p. ISBN 0-471-51122-6

Gamma, E., Helm, R., Johnson, R., Vlissides, J., 1995. Design Patterns. USA: Addison–Wesley Publ. Co. 395 p. ISBN 0201633612

García, A-B., Alvarez-Campana, M., Vázquez, E., Berrocal, J., 2002. Quality of Service Support in the UMTS Terrestrial Radio Access Network. Proceedings of the 9th Workshop of the HP OpenView University Association (HP-OVUA), June 11–13, 2002. 15 p.

Garlan, D., Allen, R., Ockerbloom, J., 1995. Architectural Mismatch or Why it's hard to build systems out of existing parts. IEEE Software, Vol. 12, Issue 6, Nov. 1995, pp. 17–26.

Gentleman, W.M., 1997. Effective use of COTS (commercial-off-the-shelf) software components in long lived systems (tutorial). Proceedings of ICSE 1997. Pp. 635–636.

Ghosh, A.K., Swaminatha, T.M., 2001. Software security and privacy risks in mobile e-commerce. Communications of the ACM, February 2001, Volume 44 Issue 2, pp. 51–57.

Grönroos, C., 1990. Nyt kilpaillaan palveluilla. Weilin & Göös.

Gschwind, T., Feridun, M., Pleisch, S., 1999. ADK–building mobile agents for network and systems management from reusable components. Proceedings of the First International Symposium on Agent Systems and Applications and Third International Symposium on Mobile Agents, 1999. Pp. 13–21.

Hall, J., Naff, R., 2001. The cost of COTS. IEEE Aerospace and Electronic Systems Magazine, Vol. 16, Issue 8, Aug. 2001, pp. 20–24.

Hamel, G., 2000. Leading the revolution. Boston: Harvard Business School Press. 333 p. ISBN 1578511895

Hartman, J., Manber, U., Peterson, L., Proebsting, T., 1996. Liquid Software: A New Paradigm for Networked Systems. Technical report, Department of Computer Science, University of Arizona, June 1996.

Henten, A., Olesen, H., Saugstrup, D., Tan, S.-E., 2003. New mobile systems in Europe, Japan and South Korea. Stockholm Mobility Roundtable 2003. 33 p.

Heyes, I.S., 2003. Just Enough Wireless Computing. Prentice Hall. 448 p. ISBN 0-130-99461-8

Hoch, D., Roeding, C., Purkert, G., Lindner, S., 1999. Secrets of Software Success – Management insights from 100 software firms around the world. 1st Edition. Boston, Massachusetts: Harvard Business School Press. 256 p.

Håkansson, H., Snehota, I. (eds.), 1995. Developing Relationships in Business Networks. London: International Thomson Business Press. 418 p. ISBN 0-415-11570-1

Hult, M., Lennung, S.-Å., 1980. Towards a definition of action research: A note and bibliography. Journal of Management Studies 17, pp. 241–250.

IEEE, 1990. IEEE Standard Glossary for Software Engineering Terminology. IEEE Std 610.12-1990. 84 p.

IEEE, 1992. IEEE Std-1061-1992: Standard for Software Quality Metrics Methodology. New York: Institute of Electrical and Electronics Engineers.

IEEE, 2000. IEEE Std-1471-2000. IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. New York: Institute of Electrical and Electronics Engineers. 29 p.

Internet.com, 2002. Definition of software. Webopedia. Available on-line at: http://www.webopedia.com/TERM/s/software.html [14.5.2004]

Internet.com, 2003. Definition of accouting. Webopedia. Available on-line at: http://www.webopedia.com/TERM/A/AAA.html. [29.05.2003]

Internet.com, 2004a. Definition of bytecode. Webopedia. Available on-line at: http://www.webopedia.com/TERM/B/bytecode.html [9.4.2004]

Internet.com, 2004b. Definition of latency. Webopedia. Available on-line at: http://www.webopedia.com/TERM/l/latency.html [11.5.2004]

Iona, 2003. Iona products by name – Orbix. Available on-line at: http://www.iona.com/products/names/ [22.8.2003]

ISO/IEC, 2001a. Software Engineering – Product Quality – Part 1: Quality Model. ISO/IEC Standard 9126-1:2001. 32 p.

ISO/IEC, 2001b. Model Driven Architecture (MDA). ISO/IEC Standard 10746-3:1996 (2001). OMG Architecture Board. 31 p.

Jarillo, J.C., 1998. On Strategic Networks. Strategic Management Journal. Vol. 9, 1998, pp. 14.

Jaring, M., Krikhaar, R.L., Bosch, J., 2004. Representing variability in a family of MRI scanners. Software: Practice and Experience. Vol. 34, Issue 1, 2004, pp. 69–100.

Joong-Han, K., Sung-Su, Y., Ramakrishna, R.S., 2001. Context-aware application framework based on open service gateway. Proceedings of ICII (International Conferences on Info-tech and Info-net.) 2001, Beijing, Vol. 3. Pp. 209–213.

Järvinen, P., 2001. On research methods. Tampere: Juvenes-Print. 189 p. ISBN 951-97113-9-2.

Kaaranen, H., Ahtiainen, A., Laitinen, L., Naghian, S., Niemi, V., 2001. UMTS networks – architecture, mobility and services. Chichester, UK: John Wiley & Sons. 302 p. ISBN 0407148654

Kalaoja, J., Niemelä, E., Tikkala, A., Kallio, P., Ihme, T., Torchiano, M., 2003. WISE deliverable D4b: WISA architectural knowledge base (WISA) and Reference Architecture (WISA/RA), version 2.02. 47 p.

Kallio, P., 2001. Application Service Provisioning – Current state and Analysis of Partnership Strategies. Master's Thesis. Oulu: University of Oulu. 73 p. ISBN 951-42-5891-6, ISSN 0786-8421

Khan, K.M., Han, J., 2002. Composing security-aware software. IEEE Software. Vol. 19, Issue 1, Jan.–Feb. 2002, pp. 34–41.

Koutsopoulou, M., Alonistioti, N., Gazis, E., Kaloxylos, A., 2001. Adaptive Charging Accounting and Billing system for the support of advanced business models for VAS provision in 3G systems. The 12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2001), 30 Sept. – 3 Oct. 2001, Vol. 1, San Diego, USA. Pp. B-55–B-59.

Krishnamurthy, S., Mathur, A.P., 1997. On the estimation of reliability of a software system using reliabilities of its components. Proceedings of the eighth International Symposium on Software Reliability Engineering, 2–5 Nov. 1997. Pp. 146–155.

Lago, P., Boggio, D., Tikkala, A., Forchino, F., Matinlassi, M., Kalaoja, J., Kallio, P., Ponzi, M.N., 2003. Wise-Deliverable 11b: Architecture for pilot 2, version 2.0. 48 p.

Lerner, M., Vanecek, G., Vidovic, N., Vrsalovic, D., 2000. Middleware Networks – Concept, Design and Deployment of Internet Infrastructure. USA: Kluwer Academic Publishers. 375 p. ISBN 0-792-37840-7

Lim, E., 2000. Electronic Commerce and the Law. Bcom (Hons) Dissertation, MSIS, University of Auckland.

Linthicum, D.S., 1999. Enterprise Application Integration. Reading, Massachusetts: Addison–Wesley. ISBN: 0-210-61583-5. 376 p.

Mahadevan, B., 2000. Business Models for Internet-Based E-Commerce – An Anatomy. California Management Review, Vol. 42, No. 4, pp. 55–69.

Marques, P., Silva, L., Silva, J., 2000. Going Beyond Mobile Agent Platforms: Component-Based Development of Mobile Agent Systems. 4th International Conference on Software Engineering and Applications (SEA'2000). IASTED/ACTA Press: Las Vegas, USA. 7 p.

Matinlassi, M., Niemelä, E., Dobrica, L., 2002. Quality-driven architecture design and quality analysis method. A revolutionary initiation approach to a product line architecture. Espoo: VTT Publications 456. 128 p. + app. 10 p. ISBN 951-38-5967-3; 951-38-5968-1

McDermid, J.A., 1997. COTS: the expensive solution? IEE Colloquium on Cots and Safety Critical Systems (Digest No. 1997/013), 28 Jan. 1997. Pp. 1/1–1/4.

Messerschmitt, D.G., Szyperski, C., 2001. Industrial and Economic Properties of Software: Technology, Processes, and Value. University of California at Berkeley Computer Science Division Technical Report UCB//CSD-01-1130 and Microsoft Corporation Technical Report MSR-TR-2001-11. 50 p.

Meyers, B.C., Oberndorf, P., 2001. Managing software acquisition: open systems and COTS products. Upper Saddle River, NJ: Addison–Wesley. 360 p. ISBN 0-201-70454-4

Mintzberg, H., Quinn, J. B., 1996. The Strategy Process – concepts, contexts, cases. 3. New Jersey: Prentice–Hall, Inc. 990 p. ISBN 0-132-34030-5

Mobile Payment Forum, 2002. Mobile Payment Forum White Paper: Enabling Secure, Interoperable, and User-friendly Mobile Payments. 19 p. Available on-line at: http://www.mobilepaymentforum.org/pdfs/mpf_whitepaper.pdf [11.11.2003]

Morisio, M., Seaman, C.B., Parra, A.T., Basili, V.R., Kraft, S.E., Condon, S.E., 2000. Investigating and improving a COTS-based software development process. Proceedings of the ICSE 2000. Pp. 32–41.

Möller, K.K., Halinen, A., 1999. Business Relationships and Networks: Managerial Challenge of Network Era. Industrial Marketing Management, Vol. 28, pp. 413–427.

Niemelä, E., 1999. A component framework of a distributed control systems family. Espoo: VTT Publications 402. 188 p. + app. 68 p. ISBN 951-38-5549-X; 951-38-5550-3

Niemelä, E., Ihme, T., 2001. Product line software engineering of embedded systems. Symposium on Software Reusability, SSR'01. 8 p.

Niemelä, E., Korpipää, T., Tuominen, A., 1999. Embedded middleware: State of the art. Espoo: VTT Publications 375. 102 p. + app. 7 p. ISBN 951-38-5359-4; 951-38-5360-8

Niemelä, E., Kuikka, S, Vilkuna, K., Ahonen, J., Lampola, M., Forssel, M., Korhonen, R., Seppänen, V., Ventä, O. 2000. Industrial software components – Development concerns and strategic initiatives. Technology survey 89/2000. Technology Development Center Tekes, (in Finnish). 129 p.

NPLACE, 2002. NPLACE – The National Product Line Asset Center. Available on-line at: http://www.nplace.wvhtf.org/.[11.11.2003]

NTT DoCoMo, 2003. Business strategy. On-line at:
http://www.nttdocomo.com/corebiz/imode/why/strategy.html. [5.3.2004]

NTT DoCoMo, 2004. i-mode contract subscribers outside Japan exceed 2 million. On-line at:
http://www.nttdocomo.com/presscenter/pressreleases/press/pressrelease.html?param[no]=419 [11.5.2004]

Oberndorf, T., Brownsword, L., Sledge, C. A., 2000. An Activity Framework for COTS-Based System. Technical Report, CMU/SEI-2000-TR-010, ESC-TR-2000-010, (October).

Oliphant, M.W., 1999. The mobile phone meets the Internet. IEEE Spectrum, Vol. 36, Issue 8, Aug. 1999, pp. 20–28.

OMA, 2002. Open Mobile Architecture Initiative – Overview Presentation. 15 p. Available on-line at: http://www.openmobilealliance.org. [29.05.2003]

Oman, P.W., Bowles, A.J., Mount, R., Karam, G., Kalinsky, D., Tervonen, M., Bundonis, V., Fischer, H., Fish, M., Longshore, D., Akiha, N., 1990. CASE: analysis and design tools. IEEE Software, Vol. 7, Issue 3, May 1990, pp. 37–43.

OSGi, 2003. OSGi Service Gateway Specification Release 1.0. Available on-line at: http://osgi.org/resources/spec_download.asp. [29.05.2003]

Perry, D., Wolf, A., 1992. Foundation for the Study of Software Architecture. SIGSOFT Software Engineering Notes, Vol. 17, No. 44, pp. 40–52.

Plepys, A., 2002. Software renting – better business, better environment: the case of application service providing (ASP). IEEE International Symposium on Electronics and the Environment, 2002. Pp. 53–58.

Poulbére, V., 2003. Impacts of new business models on the Mobile Multimedia Market. Stockholm Mobile Roundtable 2003. 16 p.

Prosa, 2003. Prosa UML 2003. Available on-line at:
http://www.prosa.fi/fin/pr2Shop.htm#me. [22.8.2003]

QNX, 2003. QNX Neutrino RTOS. Available on-line at:
http://www.qnx.com/products/ps_neutrino/[22.8.2003]

Rumbaugh, J., 1987. Relations as semantic constructs in an object-oriented language. OOPSLA 1987. New York: Association for Computing Machinery. Pp. 466–481.

Sabharwal, C.L., 1998. Java, Java, Java. IEEE Potentials, Vol. 17, Issue 3, Aug.–Sept. 1998, pp. 33–37.

Salmela, M., Korhonen, J., Kalaoja, J., 1999. Automated test reuse for product families. Software Quality Week Europe, QWE '99. Software Research Institute. 13 p.

Sametinger, J., 1997. Software Engineering with Reusable Components. New York: Springer Verlag. 272 p. ISBN 3-540-62695-6

Sankowski, D., Strzecha, K., Kolodziejski, H., Albrecht, A., Wojciechowski, R., 2003. Application of real-time operating system QNX for automatic determination of dynamic properties of resistance furnaces identification. 20th IEEE Instrumentation and Measurement Technology Conference, 2003, IMTC '03. Vol. 2. Pp. 1617–1621.

Seppänen, V., Helander, N., Niemelä, E.. Komi-Sirviö, S., 2001. Original software component manufacturing: survey of the state-of-the-practice. Proceedings 27th EUROMICRO Conference, Los Alamitos, CA, 2001. Pp. 138–145.

Shaffer, G., McPherson, G., 2001. FAA COTS Risk Mitigation guide: Practical Methods for Effective COTS Acquisition and Life Cycle Support, v1.4. Federal Aviation Administration. 37 p. + app 80 p. Available on-line at: http://www1.faa.gov/aua/resources/COTS/Guide/crmg070102.doc. [29.11.2003]

Sharma, C., Nakamura, Y., 2003. Wireless Data Services – Technologies, Business Models and Global Markets. Cambridge University Press. 361 p. ISBN 0-521-82843-0

Shaw, M., Garlan, D., 1996. Software architecture. Perspectives on an Emerging Discipline. Prentice Hall: Upper Saddle River, N.J. 242 p. ISBN 0-1318-2957-2

Sipilä, M. (ed.), 2002. Communication Technologies. The VTT Roadmaps. Espoo: VTT Research Notes 2146. 81 p. ISBN 951-38-6050-7; 951-38-6053-1

Sommerville, I., 1992. Software Engineering. Workingham: Addison–Wesley. 649 p. ISBN 0-201-56529-3

Spread Spectrum Scene, 2004. Enhanced 911 (E911) Position Location Info. On-line at: http://www.sss-mag.com/e911.html [11.5.2004]

Stallings, W., 2001. Operating Systems – Internal and Design Principles. 4th edition. Upper Saddle River, New Jersey: Prentice–Hall. 779 p. ISBN 0-13-031999-6

Sun, 2003. The source for Java technolology. Available on-line at: http://java.sun.com/ [23.11.2003]

Susman, G.I., Evered, R.D., 1978. An assessment of the scientific merits of action research. Administrative Science Quarterly 23. Pp. 582–603.

Suzuki, J., Yamamoto, Y., 1999. Toward the interoperable software design models: quartet of UML, XML, DOM and CORBA. Fourth International IEEE Symposium and Forum on Software Engineering Standards, 1999. Pp. 163–172.

Szyperski, C., 1998. Component Software: Beyond Object-Oriented Programming. Harlow, England: Addison Wesley. 411 p. ISBN 0-201-17888-5

Tapscott, D., Ticoll, D., Lowy, A., 2000. Digital capital – harnessing the power of business webs. Nicholas Brealey Publishing: Finland. 272 p. ISBN 1-85788-209-1

Telemanagement Forum, 2003. Telecom Operations Map. Available on-line at: www.tmforum.org. [1.12.2003]

Timmers, P., 1998. Business Models for Electronic Markets.
In: Gadient, Yves; Schmid, Beat F.; Selz, Dorian. Available on-line at: http://www.electronicmarkets.org/netacademy/publications.nsf/all_pk/949/$file/v8n2_timmers.pdf?OpenElement&id=949. 6 p. [29.5.2003]

Tuominen, A., Holappa, M., Niemelä, E., 1998. Ohjelmiston joustavuusominaisuuksien ratkaisumallit ja niiden hyödyntäminen. Technical report of Arttu project. Version 1.0. 45 p.

UMTS, 2000. Shaping the Mobile Multimedia Future – an Extended Vision from the UMTS Forum. Report No. 10 from the UMTS Forum. On-line at: http://www.umts-forum.org/reports/report10.pdf. 113 p.

UMTS, 2003. Mobile Evolution – Shaping the Future. A white paper from the UMTS Forum, August 203. 20 p.

Vigder, M.R., 1998. An Architecture for COTS Based Software Systems. National Research Council of Canada, Report No. 41603. 22 p.

Vigder, M.R., Dean, J., 1997. An Architectural Approach to Building Systems from COTS Software Components. The 22nd Software Engineering Workshop, NASA/Goddard Space Flight Center Software Engineering Laboratory (SEL), Greenbelt, MD, December 1997. Pp. 99–131.

Vigder, M.R., Dean, J., 2000. Maintening a COTS-Based Systems. Proceedings of the NATO Information Systems Technology Panel Symposium on Commercial Off-the-shelf Products in Defence Applications. NRC 43626. 6 p.

Voas, J. M., 1998. The challenges of using COTS software in component-based development. IEEE Computer, Vol. 31, June, pp. 44–45.

Wallace, D.R., Peng, W.W., Ippolito, L.M., 1998. Software Quality Assurance: Documentation and Reviews. NISTIR 4909.
Available on-line at: http://hissa.ncsl.nist.gov/publications/nistir4909/. [29.05.2003]

Yang, S.H.J., Tsai, J.J.P., Chen, I., 2002. Development of Wireless Embedded Systems using Component Based Software. International Journal of Software Engineering and Knowledge Engineering, Vol. 12, No. 2(2002), pp.135–153.

Yew, A., Bohoris, C., Liotta, A., Pavlou, G., 2001. Quality of Service Management for the Virtual Home Environment. Proceedings of the 12th IEEE/IFIP International Workshop on Distributed Systems: Operations and Management (DSOM '01), Nancy, France. INRIA Press. Pp. 179–190.

Interviews:

Interview with Paolo Falcarin, Investnet, February 2002 in Turin, Italy

Interview with Kari Inberg, TeliaSonera, 19th December 2001 in Oulu

Appendices of this publication are not included in the PDF version.

Please order the printed version to get the complete publication

(http://www.vtt.fi/inf/pdf/)

Published by

Vuorimiehentie 5, P.O.Box 2000, FIN–02044 VTT, Finland
Phone internat. +358 9 4561
Fax +358 9 456 4374

Series title, number and report code of publication

VTT Publications 534
VTT–PUBS–534

Author(s)

Kallio, Päivi

Title

Emergence of Wireless Services
Business Actors and their Roles in Networked Component-Based Development

Abstract

Mobility of wireless services allows their use anywhere and anytime via portable terminals. 3G (Third Generation) technologies with increased transmission speeds are expected to increase the revenue from wireless services significantly for all actors involved in business, especially as they will enable new kinds of services based on versatile business models. A business model is architecture for the product-, service- and information flows, including a description of the various business actors and their roles, the potential benefit for the actors and the sources of revenue.

Success in the wireless business requires combining an appropriate business model with an attractive terminal and content, and offering it to the user as an appealing service package. Companies have to select their role and partners in the business network carefully by taking into account the diversity of terminals, networks and software. A large variety of technologies combined with the need to save costs and a faster time-to-market leads towards open interfaces and standards to allow interoperability between different operators and networks, and a predominately component-based approach to service development.

The aim of this thesis is to analyse how wireless services can be developed more efficiently in the context of 3G networks by focusing on the actors' roles and component-based development. The business models and roles are illustrated in this research with the aid of two types of wireless services that are under development for 3G networks. Development of the two case services – a multi-player game and a trading on-line (TOL) service – in this research proves the suitability of component-based development to the development of wireless services. To achieve the best results one has to take into account the risks of commercial component use and the initial investments component-based development requires. This research also shows that component documentation helps remarkably in the former and is amendable to decrease the latter by providing information about components' use in service development.

Keywords

wireless service development, business actors, business roles, wireless business networks, business models, development, software

Activity unit

VTT Electronics, Embedded Software, Kaitoväylä 1, P.O. Box 1100, FIN–90571 Oulu, Finland

ISBN			Project number
951–38–6386–7 (soft back ed.) 951–38–6387–5 (URL: http://www.inf.vtt.fi/pdf/)			
Date	Language	Pages	Price
June 2004	English	118 p. + app. 83 p.	E
Name of project		Commissioned by	
EETU, MINTTU, WISE			
Series title and ISSN		Sold by	
VTT Publications 1235–0621 (soft back ed.) 1455–0849 (URL: http://www.inf.vtt.fi/pdf/)		VTT Information Service P.O.Box 2000, FIN–02044 VTT, Finland Phone internat. +358 9 456 4404 Fax +358 9 456 4374	

VTT PUBLICATIONS 534 Emergence of Wireless Services

Päivi Kallio

Future network technologies with increased transmission speeds will increase the revenue from wireless services. The revenues will increase for all business actors as the future technologies will enable new kinds of services based on versatile business models. Succeeding in tough competition of wireless business requires combining an appropriate business model with an attractive terminal and content, and offering it to the user as an appealing service package. Companies have to select their role and partners in the business network carefully by taking into account the diversity of terminals, networks and software. A large variety of technologies combined with the need to save costs and a faster time-to-market leads towards open interfaces and standards to allow interoperability between different operators and networks, and a component-based approach to service development.

This thesis analyses the state-of-the-art of wireless service business and ways to develop wireless services more efficiently in the context of 3G networks. For two wireless services that are under development for 3G networks is presented a business model framework. This research also evaluates the risks of commercial component use and suitability of component-based development to development of wireless services. The problematic how the business and technological aspects of the service development can be considered simultaneously – not separately as at the moment is also illustrated.

Tätä julkaisua myy	Denna publikation säljs av	This publication is available from
VTT TIETOPALVELU PL 2000 02044 VTT Puh. (09) 456 4404 Faksi (09) 456 4374	VTT INFORMATIONSTJÄNST PB 2000 02044 VTT Tel. (09) 456 4404 Fax (09) 456 4374	VTT INFORMATION SERVICE P.O.Box 2000 FIN–02044 VTT, Finland Phone internat. +358 9 456 4404 Fax +358 9 456 4374

ISBN 951–38–6386–7 (soft back ed.)
ISSN 1235–0621 (soft back ed.)

ISBN 951–38–6387–5 (URL: http://www.vtt.fi/inf/pdf/)
ISSN 1455–0849 (URL: http://www.vtt.fi/inf/pdf/)