Jiehan Zhou

PERVASIVE SERVICE COMPUTING: COMMUNITY COORDINATED MULTIMEDIA, CONTEXT AWARENESS, AND SERVICE COMPOSITION
JIEHAN ZHOU

PERVASIVE SERVICE COMPUTING: COMMUNITY COORDINATED MULTIMEDIA, CONTEXT AWARENESS, AND SERVICE COMPOSITION

Academic dissertation to be presented with the assent of the Faculty of Technology of the University of Oulu for public defence in Auditorium IT116, Linnanmaa, on 2 December 2011, at 12 noon

UNIVERSITY OF OULU, OULU 2011
Abstract

This thesis introduces a novel Web service-centric solution to pervasive computing, called Service-oriented Pervasive Computing (also called Pervasive Service Computing), which enables computer systems to deal with context in the user’s environment, to dynamically discover and compose existing services, and to develop Internet-scale multimedia applications that support users’ activities.

First, this thesis introduces the concept of Pervasive Service Computing and its relation to community coordinated multimedia, context awareness, and service-oriented computing. It then investigates the state of the art, the practices, and techniques which have been developed to support such services. Building on these tools, this study adopts a service-oriented methodology to design a reference model for Pervasive Service Computing, for accommodating specified technical requirements. This model can serve as a guide for research and development towards Pervasive Service Computing.

Second, the thesis examines the nature of community coordinated multimedia, and develops the concept of Community Coordinated Multimedia (CCM). To discover the potentials of discoverability and compositability of multimedia applications, the thesis introduces a model for Service-oriented Community Coordinated Multimedia (SCCM), and demonstrates the idea of “multimedia application as a service.” Furthermore, the thesis presents a content annotation service and evaluates its feasibility as an end-user prototype.

Third, the thesis investigates the nature of context awareness in Pervasive Service Computing, to broaden the definition of context and context-awareness. This research introduces context-aware pervasive service composition (CAPSC) applications, and specifies three-levels of context awareness. Building on this framework, the context-aware service composition prototype is implemented.

Fourth, the author examines the overall potential of service composition in Pervasive Service Computing, distinguishes its two main functions as service collaboration, and service coordination, and then develops an ODPSC (Ontology-Driven Pervasive Service Composition) ontology. To address the availability and scalability of service composition, the thesis introduces options for dynamic service composition in the Cloud, and develops an accelerated Cloud architecture for service composition in the Cloud (namely CM4SC middleware). Last, the CM4SC middleware as a service prototype is implemented.

Keywords: cloud computing, community coordinated multimedia, context awareness, pervasive computing, service composition, service-oriented computing
Tässä työssä käsitellään uutta jokapaikan tietotekniikan Web-palvelukeskeistä ratkaisua, palveluorientoitetun jokapaikan tietotekniikkaa (Pervasive Service Computing). Tämän avulla tietokonejärjestelmät voivat ottaa huomioon käyttäjän ympäristön tilanteen, löytää ja koota palveluja dynaamisesti, ja näin voidaan kehittää Internetin laajuisia käyttäjiä tukevia multimediansovelluksia.

Ensiksi työssä esitellään jokapaikan tietotekniikan palvelujen käsite sekä tällaisten palvelujen suhde yhteisöllisesti koordinoituun multimediaan, tilannetietoisuuteen ja palveluorientoituneeseen tietotekniikkaan. Tieteen nykytila sekä tällaisia palveluja tukevat kehitetyt tekniset ja kehitysvälineet esitetään. Tällaisilla tuotteilla pohjautuen työssä omaksutaan palveluorientoitunut metodiikka, jonka avulla voidaan kehittää jokapaikan tietotekniikan palvelujen tutkimusta ja tuotekehitystä.


Kolmanneksi työssä tutkitaan jokapaikan tietotekniikan palvelujen tilannetietoisuutta laajentamalla tilanteen ja tilannetietoisuuden määritelmiä. Tutkimus esittelee tilannetietoiselle jokapaikan tietotekniikan palvelujen kokoamiseen (Context-Aware Pervasive Service Composition, CAPSC) perustuvia sovelluksia ja määrittelee kolme tasoa tilannetietoisuudelle. Tämän avulla voidaan kehittää jokapaikan tietotekniikan palvelujen dynaamisen kokoamisen prototyyppin.

Neljänneksi työssä arvioidaan jokapaikan tietotekniikan palvelujen mahdollisuuksia, tunnistetaan jokapaikan tietotekniikan palvelujen mahdollisuuksia, tunnistetaan sen kaksi keskeistä toiminnallisuutta, palvelujen yhteistoiminnallisuus (service collaboration) ja palvelujen koordinointi (service coordination), sekä kehitetään ONDSC (Ontology-Driven Pervasive Service Composition) -ontologia. Työssä esitetään saavutetut mahdollisuudet ja laajennettavaa jokapaikan tietotekniikan palvelujen dynaamisen kokoamisen prototyyppi.

Asiasanat: jokapaikan tietotekniikka, palvelujen yhteisöllisesti koordinoitu multimedia, tilannetietoisuus ja palvelujen kokoaminen.
Preface

The research work related to this thesis has been carried out at MediaTeam Oulu research group, Department of Computer Science and Engineering, University of Oulu, Finland. More specifically, the research has been carried out at the CAM4Home project, which was funded by the Finnish Funding Agency for Technology and Innovation (TEKES), and the Pervasive Service Computing project, which is funded by the Ubiquitous Computing and Diversity of Communication (MOTIVE) program of the Academy of Finland.

I would like to express great thanks to my supervisors, Professor Dr. Jukka Riekki and Docent Dr. Mika Ylianttila, for supervising this thesis. Their contributions in directing and supervising the research work were crucial. Without their direction and guidance, this thesis work would not have been possible.

I would like to gratefully acknowledge the reviewers Professor Jari Veijalainen, from the University of Jyväskylä, Finland and Professor Noël Crespi, from the Institut Telecom, Telecom SudParis, France, for their valuable and extensive review comments which helped to improve the final outcome. I would also like to thank Docent Timo Korhonen, from Aalto University, Finland, for accepting serving as the opponent in the defence.

The writing of the thesis was partly conducted during research visits to Middleware Systems Research Group (MSRG), University of Toronto, supervised by Prof. Hans Arno Jacobsen, and to Henri Tudor Research Centre, Luxembourg, supervised by Mr. Patrick Blandin. I would also like to thank my colleagues, Dr. Mika Rautiainen, Dr. Junzhao Sun, Timo Koskela, Zhonghong Ou, Erkki Harjula, Oleg Davidyuk, Meirong Liu, Arto Heikkinen and Jouni Sarvanko, Ekaterina Gilman, Xiang Su from University of Oulu, Dr. Chunyang Ye, Dr. Balasubramaneyam Maniymaran, Chen Chen, Vinod Muthusamy, Kaiwen Zhang, and Young Yoon from University of Toronto for their friendship, their insights, and their suggestions for my research.

Finally, I would like to thank my beloved family.

Oulu, July, 2011

Jiehan Zhou
Abbreviations

AmI  Ambient Intelligence
API  Application Programming Interface
BPEL Business Process Execution Language
CAM4Home Collaborative Aggregated Multimedia for Digital Home
CAPSC Context-Aware Pervasive Service Composition
CCM Community Coordinated Multimedia
CD  Compact Disc
CM4SC Cloud-based Middleware for Dynamic Service Composition
C/S  Client/Server
DOI  Digital Object Identifier
EU  European Union
IA  Information Appliances
ICT  Information and Communication(s) Technology
IDE  Integrated Development Environment
IPTV  Internet Protocol Television
ITEA  Information Technology for European Advancement
MAWS  Multimedia Application as a Web Service
MANET  Mobile Ad hoc Network
MDA  Model-Driven Architecture
MPEG  Moving Picture Experts Group
MUI  Multimodal User Interface
ODPSC Ontology-Driven Pervasive Service Composition
P2P  Peer-to-Peer
PDAs  Personal Digital Assistants
PSC  Pervasive Service Computing
PSCL Pervasive Service Composition Language
QoS  Quality of Service
RDF  Resource Description Framework
SCCM Service-oriented Community Coordinated Multimedia
SOA  Service-Oriented Architecture
SOAP  Simple Object Access Protocol
UML  Unified Modeling Language
URI  Universal Resource Identifier
WCF  Windows Communication Foundation
WF  Windows Workflow Foundation
| XML | eXtensible Markup Language |
List of original papers

This thesis is based on the following original articles (I–XI):


Papers I, II, and III deal with the research problem Q1 set forth in section 1.2, which investigates the motivation for studying Pervasive Service Computing, identifies its defining characteristics, and presents a reference model approach to the subject.

Papers IV, V, VI, and VII deal with the research problem Q2 set forth in section 1.2, which explores Service-oriented Community Coordinated Multimedia, its vision, conceptualization, modeling, and implementation. Paper XIII deals with the research problem Q3 set forth in section 1.2, which addresses research issues related to context awareness and adaptive service composition. Finally, Papers IX, X, and XI deal with the research problem Q4 set forth in section 1.2, which addresses research issues related to service composition modeling, description, and its implementation in the Cloud.

The author had the main responsibility for developing system concepts (i.e. pervasive service computing, service-oriented community coordinated multimedia, community coordinated multimedia, dynamic service composition in the Cloud), and for creating proposals for system analysis and design, technical management, and for writing the papers I–XI. Dr. Mika Rautiainen had the main responsibility for the concept design and development of content annotation service in the CAM4Home project. Mr. Arto Heikkinen and Mr. Jouni Sarvanko had the main responsibility for implementing the content annotation service prototype. Mr. Juha Palola and Ms. Ekaterina Gilman had the main responsibility for implementing CAPSC prototype. Ms. Kumaripaba Athukorala and Ms. Ekaterina Gilman had the main responsibility for implementing CM4SC prototype. In addition, the author has actively authored or co-authored journal articles (Gilman et al. 2011; Liu et al. 2011; Zhou et al. 2011b), book chapters (Zhou et al. 2009a; Zhou et al. 2011a), and conference papers in various research areas, including digital television (Zhou et al. 2009b), P2P networks (Ou et al. 2009) and wireless sensor networks (Sun & Zhou 2008, Sun et al. 2010), and
emotion-oriented computing (Zhou et al. 2009c). These studies\(^1\) supplement the research work presented in this thesis from their own perspectives.

# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td></td>
</tr>
<tr>
<td>Tiivistelmä</td>
<td></td>
</tr>
<tr>
<td>Preface</td>
<td>7</td>
</tr>
<tr>
<td>Abbreviations</td>
<td>9</td>
</tr>
<tr>
<td>List of original papers</td>
<td>11</td>
</tr>
<tr>
<td>Contents</td>
<td>15</td>
</tr>
<tr>
<td>1 Introduction and motivation</td>
<td>17</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>17</td>
</tr>
<tr>
<td>1.2 Motivation and research problems</td>
<td>20</td>
</tr>
<tr>
<td>1.3 Scope, objectives, and methodology</td>
<td>22</td>
</tr>
<tr>
<td>1.4 Contribution of the thesis</td>
<td>24</td>
</tr>
<tr>
<td>1.5 Organization of the thesis</td>
<td>25</td>
</tr>
<tr>
<td>2 Overview of Pervasive Service Computing</td>
<td>27</td>
</tr>
<tr>
<td>2.1 Evolution of Pervasive Service Computing</td>
<td>27</td>
</tr>
<tr>
<td>2.2 Concepts related to Pervasive Service Computing</td>
<td>30</td>
</tr>
<tr>
<td>2.3 Community coordinated multimedia</td>
<td>34</td>
</tr>
<tr>
<td>2.4 Context-aware computing</td>
<td>35</td>
</tr>
<tr>
<td>2.5 Service-oriented computing</td>
<td>37</td>
</tr>
<tr>
<td>2.6 Summary</td>
<td>41</td>
</tr>
<tr>
<td>3 Research contribution</td>
<td>43</td>
</tr>
<tr>
<td>3.1 Pervasive Service Computing: vision, significance, conceptualization, and reference model</td>
<td>43</td>
</tr>
<tr>
<td>3.2 Service-oriented Community Coordinated Multimedia</td>
<td>54</td>
</tr>
<tr>
<td>3.3 Context-aware service composition</td>
<td>63</td>
</tr>
<tr>
<td>3.4 Service composition modeling, description, and implementation in the Cloud</td>
<td>68</td>
</tr>
<tr>
<td>3.5 Summary</td>
<td>75</td>
</tr>
<tr>
<td>4 Discussion, future work, and conclusions</td>
<td>77</td>
</tr>
<tr>
<td>4.1 Discussion</td>
<td>77</td>
</tr>
<tr>
<td>4.2 Future work</td>
<td>82</td>
</tr>
<tr>
<td>4.3 Conclusion</td>
<td>83</td>
</tr>
<tr>
<td>References</td>
<td>85</td>
</tr>
<tr>
<td>Original papers</td>
<td>95</td>
</tr>
</tbody>
</table>
1 Introduction and motivation

This thesis explores opportunities for enhancing and taking advantage of community coordinated multimedia, context awareness, and service-oriented computing. The focus is on developing service-oriented solutions or platforms for facilitating user activities through pervasive computing.

Pervasive computing (i.e. ubiquitous computing) introduces a post-desktop computing environment that provides people with information and services when and where desired (Weiser 1991). Within this field, Pervasive Service Computing (PSC) is a Web service-centric solution to pervasive computing (i.e. service-oriented pervasive computing) for facilitating people’s daily activities. The services this can provide include community coordinated multimedia, context awareness, and service composition. Community Coordinated Multimedia (CCM) enables maintaining virtual communities for the creation, aggregation, and consumption of community-created content. CCM offers enhancement of end user experience, with applications to enrich, extend, enhance, and share multimedia content. Context awareness enables applications to sense context, and adapt to changes in the pervasive environment. Service composition enables people to develop new applications through integrating existing loosely-coupled services via the Internet. In this case, a service composition is a software application developed through integrating Web services. Section 2.2 presents additional definitions of all relevant concepts used in the thesis.

1.1 Background

The author envisions the Pervasive Service Computing paradigm in the context of progress in pervasive computing, service-oriented computing, multimedia, and social networking. The research aims to develop a generic Pervasive Service Computing model for strengthening the capabilities of both service collaboration and coordination in pervasive systems. The recent developments and research relevant to Pervasive Service Computing are described as follows.

Pervasive Service Computing is a generic and sustainable solution to enable context-sensitive, environment-aware computing systems. The concept of pervasive computing was first coined by Mark Weiser (1991). Since then, a variety of other terms have been used to describe this idea, including ubiquitous computing (Abowd & Mynatt 2000), and ambient intelligence (Aarts et al. 2003). Characteristics and capabilities taxonomies were developed (Costa 2008;
Dombroviak & Ramnath 2007) to categorize and characterize pervasive applications such as invisibility, mobility, scalability, transition, location awareness, object awareness, operational awareness, etc.

The potential for pervasive service computing has evolved beyond distributed computing and mobile computing platforms. It is fuelled by the proliferation of palm-sized computing devices, wireless networking standards such as IEEE 802.11 or Bluetooth, and micro/nano sensors embedded in artifacts. The kinds of pervasive computing envisioned by science fiction writers have been steadily moving into the real world. Well-known early developments and applications are the T-engine, by the Univ. of Tokyo (Krikke 2005); aware home, by Georgia Tech (Kidd et al. 1999); infoPad by UC Berkeley (Truman et al. 2002); Cups and smart-Its by Karlsruhe (Holmquist et al. 2001); and wearable computing by MIT (Smailagic 2003).

The following examples illustrate recent work on pervasive applications: MeasureIT, by the Open Wearable Computing Group (Lawo et al. 2011), is dedicated to measuring everything, everywhere, through the widespread dissemination of smart phones, in order to provide human-centered services that improve quality of life. SFIT (Smart Fabrics, Interactive Textiles) by the MIT Media Lab and Georgia Tech proposes to “weave computers into the fabric of everyday life until they are indistinguishable from it” (Tröster 2011). Pervasive healthcare (Dey & Estrin 2011) applies pervasive computing, proactive computing, and ambient intelligence technologies to enable healthcare and wellness management, and to make healthcare available everywhere, anytime. The program vSked (Hirano et al. 2010) focuses on using ubiquitous computing technologies to support parents in caring for premature infants. The ABC project (Jakob et al. 2007) utilizes pervasive computing to support the work of clinicians in the treatment and care of hospital patients. Monarca (Setz et al. 2010) aims at designing ubiquitous computing technologies for patients with bipolar disorder. Using a mobile phone, the patient can report things like mood, medicine compliance, sleep, and personal warning signs. The device combines this information with monitoring of physical and social activity. The Escort System (Taub et al. 2011) is designed to protect people living with Alzheimer’s disease. User-worn mesh-networked badges transmit indoor location information obtained in real time. A central server sends real-time short messaging service (SMS) alerts when a user might be at risk, giving caregivers information to address the situation before problems occur. Smart energy systems will also be a prime area of application for pervasive computing (Medland et al. 2011; Paradiso et al. 2011).
These systems dynamically mitigate energy use by measuring, inferring, manipulating, and leveraging human behavior and context across various domains or environments. Pervasive shopping, retail (Narayanaswami et al. 2011), and advertising are also major areas for applications of pervasive computing (Krumm 2011). These applications include mobile shopping and checkout applications, price comparison tools, location-based services, and targeted mobile promotions, among other consumer-targeted technologies.

There are, of course, criticisms against this rise of pervasive computing applications. Many people raise concerns about the privacy implications of these devices, especially the cameras in public spaces that record much of people’s everyday lives in the UK (Davies 2011).

Pervasive Service Computing incorporates service-oriented computing into a conventional pervasive computing paradigm, which takes advantage of options for registration, discoverability, compositability, and open standards in a service orientation approach. In this thesis, service (also called Web Service) is treated as an autonomous, standards-based component, whose public interfaces are defined and described using a machine-processable format (specifically Web Services Description Language, or WSDL) that supports interoperable machine-to-machine interaction over a network using mainly Web-based standards\(^2\). Service-orientation perceives functions as distinct service units, which can be registered, discovered, and invoked over a network. Technically, a service consists of a contract, one or more interfaces, and an implementation. The service orientation principles for modern software system design are promoted through contemporary service-oriented architecture (SOA). This architecture introduces standardizations to service registration, semantic messaging platforms, and Web services technology (Dirk et al. 2005; Thomas 2005).

A number of research efforts have focused on applying dynamic composition of services to support user activities in terms of service orchestration and choreography in mobile computing environments. A project named Ozone (Georgantas & Issarny 2004) elaborated a task synthesis middleware for providing dynamic, situation-sensitive composition and reconfiguration of user tasks within an AmI (Ambient Intelligence) environment. The Aura project (Garlan et al. 2002; Sousa & Garlan 2002) developed an architectural framework which explicitly represents user tasks as collections of services, and allows tasks to be configured in the context of a computing environment. Project Gaia (Chetan

2005; Roman et al. 2002) introduced a mobile Gaia middleware for integrating resources of various devices. It provides functions such as forming and sharing resources among devices, and enables seamless service interactions.

Pervasive Service Computing incorporates multimedia and social networking to provide users with multimedia-intensive services and to strengthen the capability for coordination in pervasive applications. Digital convergence between audiovisual media, high-speed networks, and smart devices becomes a reality, and helps make media content more directly manageable by computers (ERCIM 2005). Such digital convergence presents new opportunities for people to improve the ways they work. For example, hospitals will be able to provide multimedia-intensive medicating services at the patients’ bedsides, even when the patients are being moved.

Social networking has already revolutionized the way users communicate and share information with one another in everyday life (Boyd & Ellison 2007). But the software services people use to build online social networks for communities, or to share and explore interests and experiences, can be greatly enhanced. Most of these software services are primarily Web-based, such as video and voice chat, messaging, email, file sharing, blogging, discussion groups, and so on. All these can be integrated with context-sensitive, environment-aware systems.

1.2 Motivation and research problems

Existing challenges to development of distributed application and integration in pervasive computing include the lack of service discovery and composition mechanisms. Also, the available systems for information retrieval and sharing are insufficiently community-based. These shortcomings lead to the following overall research question which this thesis seeks to answer:

How can we automatically and efficiently develop and manage service compositions to adaptively support community-scale user activities?

This research aims to answer this question through studying Pervasive Service Computing, community coordinated multimedia, context awareness, and service composition. This thesis suggests that the quality of human daily life (e.g., productivity and creativity) could be significantly improved through Pervasive Service Computing. The research suggests tackling the challenges in achieving the transition to a service-intensive world by combining community coordinated
multimedia, context awareness, and service composition. The concept of Pervasive Service Computing is illustrated in Fig.1.

Fig. 1. Pervasive Service Computing Environment.

The proposed improvements can be achieved through enhancing three aspects of pervasive computing services, namely community coordinated multimedia, context-aware computing, and service-oriented computing. Community coordinated multimedia is capable of maintaining virtual communities, and allowing them to create, aggregate, and use community-created multimedia content or services. Existing websites like YouTube offer some of this functionality. However they are proprietary and limited in functions (e.g. YouTube is only for video sharing). They lack multimedia processing intelligence, and lack service-oriented mechanisms (e.g. service discoverability and composable). Context-aware computing enables applications to sense context and adapt to changes in pervasive environments. The capability of providing service composition is named composability. In other words, composability is the capability of a system to develop applications by combining existing services.
The overall research question of this thesis is divided into the following four sub-questions. The first sub-question serves as an umbrella for the other three, each of which focuses on one aspect of potential for pervasive service computing.

Q1. What is Pervasive Service Computing, what is its paradigm, and why is it emerging as a significant aspect of pervasive computing?

Q2. What is community coordinated multimedia, what is Multimedia Application as a Web Service (MAWS), and how do we apply such Web services into community coordinated multimedia applications?

Q3. What is context awareness in Pervasive Service Computing, what is context-aware pervasive service composition, and how can we combine these features in pervasive service systems?

Q4. How do we describe service composition in the context of Pervasive Service Computing, what is dynamic service composition in the Cloud, and how do we enable pervasive service composition in the Cloud?

1.3 Scope, objectives, and methodology

To accommodate community coordinated multimedia, context awareness, and service-oriented computing, this research proposes a novel Web service-centric solution to pervasive computing called Service-oriented Pervasive Computing (or Pervasive Service Computing). This enables us to enrich content through connecting people, to adapt to changes in the users’ environment through context awareness, to dynamically discover and compose existing services, and to develop Internet-scale applications that support users’ activities. This research aims to provide a Pervasive Service Computing environment in order to improve user productivity and innovation, by delivering on-the-air services that support both basic and complex user activities—anytime, anyplace, on any device, and on any network.

The first basis for pervasive service computing is our personal experiences, which come from our active participation in events or programs, and lead us to accumulate knowledge, skills, and enjoyment. This natural connectivity is enhanced through the ever-growing system of networked multimedia content and services, together with growing penetration of broadband Internet connections. Collaborative use of multimedia and services enhances and extends people’s experience and their capacity to share with other people. On one hand, human
experience is manifested by several web services that have grown popular in recent years, such as Wikipedia\(^3\), Flickr\(^4\), YouTube\(^5\), Joost\(^6\), and Google Maps\(^7\). To enable end users to collaboratively create and deliver rich multimedia, the author first describes all these emerging forms of connectivity with the term “Community Coordinated Multimedia” (or CCM). This research will explore the potential for CCM through an EU ITEA project called CAM4Home.

The second basis for pervasive service computing is the growing demand for context-aware systems. We want systems that are sensitive to changes in the social or spatial environment. But how do we enhance context-awareness in pervasive service composition? In seeking answers to this question, the author broadens the definitions of context and of context-awareness in terms of pervasive service composition. This thesis proposes development of context-aware pervasive service composition allowing a pervasive system to provide the user service compositions that are sensitive to contextual information from networks, devices, and users. These applications can be flexible and adaptive to change. To investigate the potential for such services, the author starts by examining the nature of context-aware pervasive service composition, and then designs an architecture for context-aware service composition in which composition applications (i.e. compositions) are classified into three types: a) context-aware peer coordination, b) process service adaptation, and c) utility service adaptation. This part of the research is conducted through the China-Finland PSC project, which is funded by the MOTIVE Ubiquitous Computing and Diversity of Communication program.

The third basis for pervasive service computing is the ever-growing interconnectivity of telecommunication networks, in which a service “internetwork” is emerging. In the service internetwork, services from various providers are interlinked with public interfaces which are exposed to requestors. Using service composition, one can collaborate and coordinate primitive services to achieve complex activities, by smoothly integrating transaction, voice call, messaging, and content delivery services. Service composition facilitates applications development and deployment so that organizations can improve their agility and automation. In this study, the author examines a generic user activity

---

model, and categorizes service composition into service coordination and collaboration models, and studies service composition descriptions and service composition in the Cloud. This part of the research is also based on the China-Finland PSC project.

1.4 Contribution of the thesis

A brief overview of the contributions is provided below, while the detailed contributions are presented in Chapter 3.

The first contribution is to present a vision, a conceptualization, and a reference model of Pervasive Service Computing. In Paper I, the author suggests Pervasive Service Computing (PSC) as a Web service-centric solution to facilitate modern daily activities, with an emphasis on service composition, building applications from services, and controlling the execution of these applications. In Paper II, the author categorizes characteristics of pervasive service computing and identifies challenges to the development of this field by examining four typical service-oriented computing scenarios under the umbrella name of “Pervasive Campus.” In Paper III, the author proposes a generic Reference Model of Pervasive Service Composition (PSC-RM) to guide pervasive service computing architecture design and implementation.

The second contribution of this thesis is to explore the potential for Service-oriented Community Coordinated Multimedia (SCCM). This involves addressing research questions concerning the vision, conceptualization, modeling, and implementation of service-oriented, community coordinated multimedia. In Paper IV, the author explores how human experience is being extended and enhanced by collaboratively utilizing electronic and networked content and multimedia-intensive services. In Paper V, the author applies a service orientation approach with Web services technology to establish a service-oriented CCM architecture. This architecture enables us to tackle the requirements of scalability, discoverability, composability, layered abstraction, Quality of Service (QoS), and agility in distributed collaborative multimedia management. In Paper VI, the author examines methodologies for metamodeling community coordinated multimedia, and introduces a community coordinated multimedia metamodel. This model was co-developed as a part of ITEA2 CAM4Home Metadata Framework. The community coordinated multimedia metamodel aims to provide a primitive constructor for instantiating, classifying, capturing, indexing, searching for information on program entities, and to provide supports for user
decision-making concerning multimedia delivery options. In Paper VII, the author introduces a concept of multimedia applications as Web services, and then describes a multimedia content annotation end-user service.

The third contribution of the thesis is to expand the scope for context-aware service composition. In Paper VIII, the author broadens the definitions of context and context-awareness in terms of pervasive service composition. The thesis then classifies context-aware service composition into adaptations at three levels: peer coordination, process service adaptation, and utility service adaptation. Next the author specifies a design process for building context-aware pervasive service composition applications, then designs and develops the context-aware pervasive service composition architecture.

The fourth contribution of the thesis is to explore service composition modeling, description, and implementation in the Cloud. In Paper IX, the author examines and formalizes service composition, with categories of service collaboration and coordination. In Paper X, the author explores ontology methodologies in pervasive service computing, and generalizes the concept of ontology-driven pervasive service composition. This leads to the design of an ontology model for Pervasive Service Composition. In Paper XI, the author explores dynamic service composition in the Cloud, which aims to provide users with a Cloud-based middleware for dynamic service composition, which can support on-demand service composition over the Cloud.

In the process of researching pervasive service computing, the author has participated extensively in publications of magazine articles, book chapters (Zhou et al. 2009a; Zhou et al. 2011a), technical reports and conference papers in various research areas, including digital television (Zhou et al. 2009b), P2P networks (Ou et al. 2009) and wireless sensor networks (Sun & Zhou 2008; Sun et al. 2010), emotion-oriented computing (Zhou et al. 2009c), and pervasive social computing (Zhou et al. 2011b). These studies supplement the research work presented in this thesis from their own perspectives.

1.5 Organization of the thesis

The thesis is organized as the follows. In this chapter, the background of the research topics, the motive for this research, the research problems, scope and methodology, as well as a brief overview of the contributions of this thesis are all briefly discussed.
Chapter 2 presents a literature overview of the research relevant to this thesis, including the subjects of community coordinated multimedia, context-aware computing, and service composition.

Chapter 3 summarizes the main contributions of the thesis, which come under four main headings: the pervasive service computing visions and challenges, community coordinated multimedia computing, context-aware computing, and service composition.

Chapter 4 draws conclusions, evaluates the research, and presents directions for future work.
2 Overview of Pervasive Service Computing

This chapter provides an overview of the research topics related to community coordinated multimedia, context awareness, and service composition. Section 2.1 briefly presents the evolution towards pervasive service computing. Section 2.2 defines concepts related to pervasive service computing in terms of community-coordinated multimedia, context-aware computing, and service composition. Sections 2.3, 2.4, and 2.5 present the state of the art on each key technique for accommodating community coordinated multimedia, context-aware computing, and service composition. Section 2.6 summarizes the chapter.

2.1 Evolution of Pervasive Service Computing

Pervasive service computing arises from developments along the path of distributed systems, mobile computing, and pervasive computing as identified by Satyanarayanan (2001) (See Fig. 2).

As summarized in (Smith et al. 2008), distributed systems arose at the intersection of personal computers and local area networks. A real distributed system consists of several autonomous computers that are connected by a computer network, and communicate with each other by sending messages. A distributed database may be stored in multiple computers located in the same physical location, or may be dispersed over a network. Distributed databases were developed during 1970s for a real distributed environment, and by now there are a growing number of distributed database algorithms, such as distributed locking, query processing, recovery, etc. Protocol layering was invented by IBM (SNA) (Friend et al. 1988) and generalized by ISO/OSI around 1980 (Grigonis 2000). This body of knowledge spans many other areas that are foundational to pervasive computing, such as telecommunication networks, fault tolerance, high availability, remote information access, security, privacy, etc.

Mobile computing arose in the process of building distributed systems with mobile clients. The results achieved can be grouped into the following areas: mobile networking, mobile information access, support for adaptive applications, system-level energy saving techniques, location sensitivity, etc.

---

Pervasive computing can be characterized as a system of embedded computing and communication capabilities, which is so gracefully integrated with users that it becomes a “technology that disappears” (Smith et al. 2008). Pervasive computing introduces a long-term vision of computational resources for providing information and services to people when and where desired. The terms “ubiquitous computing” and “ambient intelligence” have also been used to describe this paradigm (Hansmann 2003). Pervasive computing is being extensively studied and developed for the capabilities of invisibility, mobility, scalability, transition, location awareness, object awareness, operational awareness, etc. Specifically, pervasive computing incorporates three additional research thrusts as aspects of its agenda: effective use of smart spaces, invisibility, and localized scalability.

Fig. 2. Evolution of Pervasive Service Computing.

The drivers we see for the evolution of pervasive computing towards Pervasive Service Computing can be grouped into the following broad areas.

1. Service-oriented architecture (SOA) provides methods for system development and integration in which systems can group functionality around business processes, and then package them as interoperable services (Thomas 2005; Zhang 2006). SOA separates functions into distinct units, or services, which developers can make accessible over a network in order that users can combine and reuse them in the production of business applications. These
services communicate with each other by passing data from one service to another, or by coordinating an activity between two or more services.

2. Context-aware computing (Loke 2006) refers to a general class of mobile systems that can sense their physical environment (i.e. their context of use) and then adapt their behavior accordingly. For example, the AMBIENCE project,\(^{10}\) has aimed to empower people and improve their quality of life by providing a digital environment that is conscious of people’s presence, and is both sensitive and responsive to their needs, habits, gestures, and emotions. Context-aware computing emerged from the ubiquity of mobile terminals, which spurred a growing demand for context and for ever more sophisticated applications. Three important aspects of context are a) where you are, b) whom you are with, and c) what resources are nearby (Schilit \textit{et al.} 1994).

3. Community coordinated social media is a means of aggregating various media sources via the Internet, and is used to efficiently exchange and distribute information via social networks. Social networking makes individuals into groups as online communities, in which members can share common interests such as hobbies or social concerns. With the integration of social media services, community coordinated social media is playing an ever more significant role in changing society, as seen in the Arab revolutions and revolts in 2011,\(^ {11,12}\) One Cairo activist succinctly explained, “We use Facebook to schedule the protests, Twitter to coordinate, and YouTube to tell the world.”\(^ {13}\) Other terms being used for addressing this paradigm include online communities, online social networks, etc. One can also talk about mobile communities if the access to the community is through mobile terminals (SMS or online social media sites) (Veijalainen & Rehmat 2010).

4. Smart devices (Want & Borriello 2000) overlap in definition with information appliances (IA), embedded systems, mobile devices, wireless devices, and handheld devices. These are appliances specializing in information, or personal devices which are designed to perform specific activities such as playing music, taking photographs, or editing text in a simple and user-


friendly way. Typical examples are smart phones and personal digital assistants (PDAs).

2.2 Concepts related to Pervasive Service Computing

Pervasive Service Computing is regarded as a Web service-centric solution, to enrich, extend, and enhance people’s experiences through facilitating people’s everyday activities. These activities include community coordinated multimedia, context awareness, and service composition. With such services, people’s experience is being extended and enhanced collaboratively through utilizing networked or electronic content and services. The users’ everyday activities of communicating with peers, or requesting, receiving, processing, and exchanging information are empowered by these links between a service world and the users fulfilling a task.

Concepts related to service composition

Services in the community coordinated multimedia system are networked applications (e.g. IPTV, web games) that provide multimedia. In particular, the term “services” refers to multimedia-intensive end-user applications.

Service composition is a process of building applications from services, and controlling the execution of these applications. Pervasive Service Composition takes full advantage of both pervasive computing and service computing to ubiquitously provide users with mobile services for managing their everyday activities. In the user-generic activity model (Zhou et al. 2009), the author regards each user task as part of an interaction session (or task session) between the user and the surrounding service world. From the service world’s perspective, a task session includes two major phases: peer coordination, and service collaboration. Peer coordination seeks appropriate parties (i.e. service providers) to cooperate in the task session, and assembles the co-operators’ functions and competences to implement the task. When peer coordination is set down and services have been identified, service collaboration comprises the identified services that run on the coordinated service peers. A peer is an organization or party owning various computing resources, and a peer that provides services is a service peer. Peers can form groups, join groups, or resign from them.

Pervasive Service Composition Language (PSCL) aims to extend WSDL (Web Services Description Language) and provides the minimal set of concepts
and essential constructs necessary to specify peer coordination and service collaboration for automating everyday activities. PSCL is based on XML, and defined by an XML schema.

**Concepts related to context awareness**

Based on definitions given by Dey et al. (2001), Pascoe (1998) and Schilit (1995), the author extends the definition of context as the information characterizing an interaction between a user and his or her service world. However, context is more than information characterizing a participating entity. It is also information utilized for initiating, controlling, and maintaining a task session. As a result, the author defines context as follows: Context is any information characterizing the situation of a task or interaction between a user and his or her service world. Context is categorized into user context, peer context, process context, physical context, and service context. Schmidt et al. (2002) list three categories of user context: information about the user (e.g. knowledge of habits or emotional states), information about the user’s social environment (e.g. co-location of others, social interaction, and group dynamics), and information about the user’s tasks (e.g. spontaneous activities, engaged tasks, or general goals). Peer context (i.e. service-provider context) includes information on the peer’s services and social roles. Process context includes information about initiating, controlling, and maintaining task sessions. Physical context includes information on physical surrounding environments (e.g. absolute position, relative position, and co-location) and physical conditions (e.g. noise, light, or pressure). Service context includes managerial and technical information about the service, such as service functions, network connectivity, communication bandwidth, or service cost.

Based on definitions given in Schilit and Theimer (1994) and Pascoe (1998), the author defines context-aware computing as the process of collecting, using, and managing the various types of context mentioned above during a task. Context-awareness enables computer systems to adaptively initiate and manage an interaction with the user by correctly modeling and reasoning with the various types of context.

**Concepts related to community coordinated multimedia**

Community Coordinated Multimedia (CCM) systems support virtual communities for the creation and use of member-created content, or content from
registered multimedia services. The system provides tools for enhancement of community experience, such as applications for sharing multimedia content or interacting with registered services via fixed or mobile devices. The content can be created, annotated, and aggregated by several end users, and associated with relevant service providers and their content offerings.

Multimedia is a synchronized presentation using any combination of media types, such as text, graphics, image, audio, video, or animation. Multimedia systems carry complex information such as movies encoded into video data. Multimedia processing is a computationally intensive task (typically dealing with images and videos). The thesis emphasizes new applications of multimedia image and video services in context-aware systems.

Content is a collection of multimedia elements, consisting of either directly embedded or referenced (Universal Resource Identifier) multimedia data. Specifications of associations between content elements are necessary, especially when content element sources are distributed in the network. Content might be generated by the end users, or provided by companies that focus on multimedia distribution, e.g. the traditional broadcast (television and radio) industry.

Community, in the setting of community coordinated multimedia, is any virtual gathering of people with a shared interest, who interact primarily by uploading, authoring, annotating, or sharing content (e.g. audio, video, text documents). Such communities can also collectively subscribe to multimedia-intensive services (e.g. IPTV, movie, radio, games, maps). Communication between the community members happens primarily through multimedia content.

A model is a representation or description of an external reality (such as a person, system, phenomenon, or process) for a special purpose of answering questions about how the thing operates and interacts under various conditions. Metamodels provide a set of facilities to instantiate models. These facilities within a metamodel include metadata, meta operations, and meta semantics. Metadata is an overarching conceptual model that provides a framework to organize concepts and relationships governing operations and identify existent features of within a system (e.g. CD title, CD author). Metadata is expressed in language understandable to people rather than in computer coding. A meta operation is a set of declarations directing behavioral features of metadata to provide to support for enriched application development. For example, MPEG-7 Java API (Steiger et al. 2003) allows developers to enrich interface functionality for certain descriptors. Meta semantics can be represented by formal languages and also be interpreted by computers. The formal languages representing meta
semantics are regarded as schema. The schema language examples are RDF Schema and XML Schema (Decker et al. 2000, Kim et al. 2006). Meta semantics mainly serves to enable interpretation by computers.

An ontology is a description of the contents or parts of a system and of how they relate. Design of ontologies serves to enable knowledge sharing and reuse (Gruber 1993; Staab & Studer 2009; Veijalainen 2008). Based on a generic view, the author regards an ontology as a data model defining the primitive concepts, relations, and rules comprising a topic of knowledge, in order to capture, structure, or enlarge the explicit or tacit topic knowledge to be shared between people, organizations, computers, and software systems (Zhou & Dieng-Kuntz 2004). From the viewpoint of a data model, an ontology is an hierarchical arrangement of metadata.

Metamodelling is an activity of specifying and defining metadata, meta operations, and meta semantics as a functional whole. There is no consensus on unified metamodelling methodologies. But there exist various modeling methodologies for various purposes, e.g. the Object-Oriented method for software modeling, and the ontology-based CommonKADS (Bienvenido & Flores-Parra 2002) for knowledge base modeling. CCM metamodelling is viewed as a set of activities of using existing modeling methodologies for building CCM metamodel. Metamodelling methodology is viewed as a modeling method specific to the metamodel development. In this sense, it provides methods and tool suites for analyzing, designing, and building metamodels. CCM metamodel is structured for facilitating multimedia users to develop, manage, and exchange multimedia within the context of CCM. Concretely, this metamodel consists of CCM metadata, CCM meta semantics, and CCM meta operations. The metadata specifies constructs of concepts, attributes, relationships, and rules for classifying, capturing, and conveying information on CCM multimedia. The meta semantics specifies the usage of metadata, i.e. the principles, and rules for constructing a validation CCM document. The formalized CCM meta semantics in a certain formal language becomes a CCM schema for describing CCM documents. The documents can be exchanged between not only humans, but also computers. The meta operation specification provides application developers with application programming interfaces (e.g. operations of creating and deleting CCM metadata) to support rich multimedia application development, e.g. multimedia annotation, etc.
2.3 Community coordinated multimedia

Community coordinated multimedia involves sensing and interpreting ongoing activities in the social environment. Kolvenbach et al. (2004) claim that community portal solutions must integrate an awareness environment for capturing users’ presence or actions in their communities, and for notifying the communities’ members of the other members’ activities. This kind of community coordination can be described in two different ways (Zhao & Stasko 2002). First, it concerns the degree to which people know about each other, share social norms, accept roles within their community, or share concerns about issues affecting the community. Second, community coordination is about understanding the common interests of a community, and providing applications that support these common concerns.

One important characteristic of a community is sharing common interests and preferences, but not necessarily common goals (Sumi & Mase 2000). Community coordination can be understood from the perspective of building and managing a virtual community. Virtual communities are computer-mediated spaces resembling real life communities, through which individuals interact with each other to extend the experiences of everyday life. Through these communities, they can cross geographical and political boundaries in order to pursue mutual interests or goals. Many definitions of virtual community have been proposed (Lee et al. 2003). Most definitions describe it as a shared platform for individual members to do things, such as participate in public discussions. In these forums, participants have discussions with one another, share opinions, knowledge, feelings, or common topics of interest. Virtual communities also enable personal relationships, since with sufficient time the participants develop self-sustaining relationships amongst themselves. The communities support member-generated content, which refers to the data, information, discussion, expression, and feelings expressed in discussions led by members.

Requirements for the support of community coordination are analyzed in Kolvenbach et al. (2004). According to these authors, community coordination should be user-friendly in helping members to define personal awareness profiles, and access the range of shared services. Community coordination should also provide adequate contextual information for the current situation of the users. It should be able to infer the context or meaning, and share contextually aware programs. Community coordination should enable users to reach a common understanding, and keep users informed about activities in related communities.
Increasingly, virtual communities manage user-based content generation and support with the help of existing social networking systems. The author sees a future of enriching social media services, providing community coordinated content annotation, and aggregating end-user services.

To enable computerized community coordination is challenging, because people in a community are often neither organized as a group, nor united by common goals. For computational purposes, community coordination can be interpreted in two different ways. First, it can be interpreted as facilitation of awareness among physically separated communities (Zhao & Stasko 2002). This type of application uses technology to bridge physically dispersed people. One of these applications is 2nd Life, which recreates the real world inside the digital world. Second, community coordination can be interpreted as facilitating the needs of a community (Sumi & Mase 2000). This is a challenging task, because it is necessary to find a way to collect data from the community before making any decisions about the group’s needs. Current applications collect data by asking users to create their own profiles, through inputting their personnel information and interests.

2.4 Context-aware computing

Context-aware computing enables applications to automatically adapt their operations to dynamic changes in context data. Apart from earlier research on context definitions (Dey 2000; Schilit et al. 1994) and context-aware applications (Asthana et al. 1994; Cheverst et al. 1999; Gellersen et al. 1999; Henricksen et al. 2002; Oppermann et al. 1996; Want et al. 1992), recent studies are rapidly evolving in terms of context definition (Mancini et al. 2009), context accuracy (Lee et al. 2009), security-relevant contexts (Johnson 2009), context-management platforms (Schmidt et al. 2009), extended context-aware applications such as battery management (Ravi 2008), safety-critical systems (Bardram & Norskov 2008), and context verification (Schmidtke & Woo 2009).

To extend the understanding of context awareness, Janowicz et al. (2008) defined context as any kind of additional information which has an impact on similarity judgments at execution time. These authors distinguished six kinds of contexts, including user context, undesired or desired context, application context, discourse context, representation context, and interpretation context. Each kind of context has its own impact on the resulting similarity values, and on their interpretation. To address emerging contexts in mobile privacy, Mancini et al.
focused on a notion of context in relation to privacy. They defined contextual privacy as an emerging subjective sphere with its own socio-cultural knowledge, functions, relations, and rules, which is called a “place,” as opposed to the objectively defined notion of context defined by physical and factual elements, which is called a “space.”

For determining reliable contextual information in smart spaces, Lee et al. (2009) proposed a pragmatic context classification and a generalized context modeling scheme, which is based on sensor fusion techniques. For achieving effective security in a mobile computing environment, Johnson et al. (2009) coined the term “security-relevant context,” and proposed the notion of shrink-wrapped security, which couples a user’s situation with security applications. Li et al. (2008) also introduced trust computing technology into pervasive computing, and proposed terminal equipment architecture. For providing high interoperability for pervasive computing, Schmidt et al. (2009) introduced a generic context service as a standard Web service interface, which allows adding context components and their sensors at runtime. Moreover, low-level forms of context such as process context and high-level context are described in this paper. Hu et al. (2008) introduced context management, and proposed an autonomic context management system (ACoMS). This system supports the dynamic configuration and re-configuration of context information which is required by applications, and achieves fault tolerance without relying on redundancy. Gu et al. (2008) applied a P2P approach to context reasoning for deriving and obtaining additional context data from low-level context data, which may be spread over multiple domains in pervasive computing environments. To provide a uniform view of multiple-distributed context sources, Xue et al. (2008) studied schema matching for context-aware computing, and proposed schema-matching middleware, in which local schemas of context data from individual sources can be matched into a set of global schemas.

In the extension of context-aware applications, Ravi et al. (2008) proposed a system for context-aware battery management. The system extends the use of context information for battery management. This enables awareness of factors such as location in order to predict the next charging opportunity, discharge speedup factors, or operate crucial applications such as telephony. Similarly, to reduce power consumption in wearable computing systems, Murao et al. (2008) proposed a context-aware system that changes the combination of accelerometers that consider energy consumption. The proposed system can manage sensors to achieve a high accuracy of activity recognition, with low energy consumption.
To enable enterprises to be more flexible, adaptive, and to survive within an increasingly dynamic market, Wieland et al. (2008) introduced workflow technology into pervasive production environments with the notion of smart workflow. Smart workflow is a context-aware process which can adapt to changes in the physical context. This kind of workflow awareness can be applied to machines, tools, or workers in production environments. Bardram et al. (2008) extended context-awareness technologies in building safety-critical systems. They presented a context-aware patient safety system called CAPSIS for operating rooms. This system has a general awareness of the working context inside the operating room, with sensitivity to the staff, the patients, equipment, and medical material. The CAPSIS system can provide the surgical team with important clinical data at the appropriate moment, as well as detect potentially critical safety issues.

To provide location-specific information for mobile but visually impaired users, Stewart et al. (2008) presented an urban orientation system called “Talking Points.” This is based on the idea that an individual’s walking journey can be enhanced by a system providing contextual information about points of interest (POIs) along their route. The research revealed that the system could be enhanced by accessing several kinds of contextual information, such as orientation (left, right, ahead, behind) information, annotations for barriers and obstruction markers, porting for access to mobile phone platforms (such as Google Android); and support for on-the-go users with content contributed directly from mobile devices.

This thesis extends the definitions of context and context awareness in the field of pervasive service composition. It develops a context-awareness architecture for pervasive service composition, and then applies it to the multimedia annotation domain.

2.5 Service-oriented computing

The aim of service composition is to facilitate application development. As an ongoing, emerging technology, service composition attracts much attention and interest in terms of frameworks, specifications or representations of service compositions, semantics, and automatic service compositions (Li et al. 2007; Qiu et al. 2006). Specifically, many research efforts aim to improve the characteristics of service composition such as reliability, adaptability, flexibility, representability, semantics, reusability, availability, security, visual-ability, atomicity, and
coordination. These efforts take account of service accessing networks and dynamic composition environments. These challenges in the development of service-oriented computing are detailed as follows (Zhou et al. 2009).

Reliability is the capability of a composite service to perform its required functions under stated conditions, for a specified period of time. In reliability studies, a service composition is conceptualized as a process of setting up a service path (e.g. service A uses/calls B) within a network. Unreliable services and unreliable connections both result in unreliable service compositions. The reliability mechanisms are particularly necessary for ubiquitous service composition, e.g. service composition via mobile ad hoc networks (MANET). Backup service composition (Zhou et al. 2008) and re-composing service paths (Jiang et al. 2009) are approaches for improving service composition reliability.

Adaptability is the ability of a service composition to adjust itself efficiently to changed environments (e.g. a newly available service, a newly unavailable service, or a changed business rule). The adaptability factor allows a service composition to change participant services, composition logic, and data dependencies at running time. Karastoyanova et al. (2005) proposed a new construct, <find_bind> to BPEL, for enabling the dynamic binding of Web services at run time. Lins et al. (2007) presented an approach for improving adaptability by extending the composition engine. Zhai et al. (2008) proposed a reflective framework to improve the adaptability of BPEL-based Web service composition.

Flexibility is the ability of a service composition to respond to potential internal or external changes. In the field of ubiquitous robotics, flexible service composition is able to automatically generate a plan for ubiquitous robotic services. Linne et al. (2007) proposed an approach which automatically generates a flexible plan for service composition by optimizing both the number of services and their parameters which appear in the composition. Linne et al. (2007) proposed a typical scenario for assisting an elderly person by composing ubiquitous robotic services (e.g. numeric services and a robot control service). Similar research on automatic service composition includes personalized automated-service composition (Li et al. 2008) and a program for automatically converting Web service composition plans into BPEL4WS (Gu et al. 2007).

Representability is the ability to describe all composition relationships, e.g. control flow, data flow, participant services, composition logic, data dependencies, rules for data transfer, etc. Orriens et al. (2003) developed ServiceCom, a tool for service composition specification, construction, and execution that supports
complex Web service composition specifications. The directional graph-based, and semantic-based methods are often used to describe the composition relationship. Zhou and Mao (2008) proposed a description approach to the Web services composition relationship, based on the extended directional graph. Zhou and Zeng (2006) addressed semantic specification of grid service composition, with the expansion of the pi-calculus to depict the dynamic, concurrent, and interactive characteristics of grid service composition.

Semantic service composition uses the principles of the semantic Web to provide more concrete service composition, by considering the semantic properties of services. While non-semantic compositions identify only the structure of the messages exchanged, semantic compositions also interpret the message content (Kapitsaki et al. 2007). Li et al. (2007) proposed a method for semi-automatic composition of Web services, combining BEPL with process ontology. Another method, based on description logics and AI planning, was proposed to achieve a semantic specification of service composition in (Kuropka et al. 2008; Qiu et al. 2006).

Reusability can be defined as the likelihood that a composite service can be used again with slight or no modification. Zhou et al. (2007) envisions that current service-oriented architecture (SOA) can focus on service composition for application development (i.e., application composition which is volatile, such as “compose one time and use one time”). Zhou et al. (2007) proposed a method to improve reusability of service composition, focusing on service dependency management or a dependency-aware service composition.

Availability refers to ensuring the continuity of a service in the fluctuating composing environment. This involves creating a fall-back mechanism, based on an alternative atomic service. If one of the composing services is not available or fails during execution, a dispatcher component needs to be able to dynamically switch to alternative Web services that provide equivalent functionality in order to fulfill the consumer request. The availability mechanisms are especially required in ubiquitous service composition, to ensure continuous operation of the composition. In availability research, Lakhal et al. (2004) proposed an architecture with an hierarchy of arbitrary-nested transactions for Web service compositions. Dahlem et al. (2007) presented a mediation model, by transparently selecting (and executing) an alternative composition at runtime for improving the availability of composed services.

Security involves protecting composition service confidentiality, integrity, and authentication. The security policy specifies authentication mechanisms
(username/password pairs, binary certificates), encryption algorithms, digital signatures, etc. These security mechanisms are particularly necessary in ubiquitous service composition. The security system needs to ensure that confidentiality and integrity are supported by a partner Web service before writing the BPEL process that invokes the partner service. In security studies, Charfi and Mezini (2005) proposed a framework implemented by a set of aspects in AO4BPEL in order to secure BPEL compositions using WS-Security and WS-Policy. Song et al. (2006) designed flexible security policies involving little runtime overhead to construct complex services through combining available service components on request.

Visibility is the property that provides the visible service creation environment. Visible service composition is required in ubiquitous service composition environments in order to provide a better approach to composite Web service. In visibility studies, composition templates use abstract descriptions of reusable compositions containing several placeholders, which enable services to provide visual composition. Braem et al. (2006) presented a high-level, visible service creation environment to provide service composition templates, verification of compatibility, guideline conformity, and advanced separation of concerns through aspect-oriented software development. Oberleitner and Dustdar (2003), proposed a component workbench in which components from different models can be composed and combined with multi-party Web services.

Atomicity requires service composition to follow an “all or nothing” rule (i.e., in a series of operations on composing services, either all occur, or nothing occurs). When “nothing occurs” in a composition of services, it means that if one of the operations fails, then all the other operations will be invalid. This is a highly desirable property for achieving application consistency in service compositions (Ye et al. 2009). To achieve such an atomicity property, Hagen and Alonso (2000) proposed an approach based on the notion of an atomicity sphere, or a structural criterion for the backend processes of involved services. If a service satisfies the atomicity sphere, it implies that this service is atomic. Moreover, Ye et al. (2009), proposed a process algebraic framework for publishing atomicity-equivalent public views from the backend processes. In their framework, these authors presented algorithms to construct atomicity-equivalent public views, and to analyze the atomicity sphere for a service composition. The atomicity mechanisms are particularly necessary in ubiquitous service compositions such as those used in supply chain and insurance domains.
Coordination refers to the methods and techniques for federating basic Web services into composite Web services—in a more adaptable manner than traditional programming. In coordination studies, service compositions are conceptualized through coordination-theoretic modeling, with a top-down analysis. The coordination mechanisms are particularly required in ubiquitous service composition. An adaptive agent coordination framework (Sun et al. 2007) and a coordination-theoretic modeling (Qian et al. 2008) are proposed to improve the coordination of Web service composition. Conventionally, service-oriented computing and service composition are applied in business application developments. And the author proposes such service composition in support of people’s common daily activities. This research pays particular attention to the issues of service coordination, composition description, and composition availability.

2.6 Summary

This chapter presents the ICT evolution towards Pervasive Service Computing, and the concepts related to community coordinated multimedia, context awareness, service-oriented computing, and service composition, which will be utilized in this thesis. The chapter gives an overview on the state of the art in accommodating community coordinated multimedia, context-aware computing, and service composition. The next chapter presents the author’s research contributions in detail.
3 Research contribution

In this chapter, the contributions of this thesis and the original publications preceding it are presented in detail. These contributions fall into four main areas: (1) the vision, conceptualization, and reference model of Pervasive Service Computing (PSC), (2) the conceptualization and modeling of Service-oriented Community Coordinated Multimedia (SCCM)—along with the service-oriented content annotation prototype implementation as an end-user application, (3) the design and prototype implementation of context-aware pervasive service computing, and (4) the modeling and description of service composition and service composition in the Cloud. Papers I, II, and III describe the PSC vision, conceptualization, and reference model. Papers IV, V, VI, and VII deal with SCCM-related issues. Paper VIII addresses context awareness and adaptive service composition. Papers IX, X, and XI address the research issues related to service composition, modeling, description, and implementation in the Cloud.

3.1 Pervasive Service Computing: vision, significance, conceptualization, and reference model

This section presents the motivation for studying Pervasive Service Computing, examines the significance of the research, conceptualizes and formalizes the research problems, and presents the reference model for an approach to Pervasive Service Computing. Papers I, II, and III describe contributions in these areas. There are mountains of research and numerous studies of pervasive computing, service-oriented computing, social networking, and social media. However, less research is reported on the paradigm or the practice of Pervasive Service Computing. It is especially rare to find studies which deal with all the three dimensions of Pervasive Service Computing, namely community coordinated multimedia, context awareness, and service-oriented computing.

Paper I. Pervasive Service Computing toward Accommodating Service Coordination and Collaboration

This paper suggests Pervasive Service Computing (PSC) as a Web service-centric solution to facilitate modern daily activities, with an emphasis on service composition, building applications from services, and controlling the execution of
these applications. The author had the main responsibility for the technical content and writing of this paper.

This research is a response to the worldwide challenge of migrating our economies from economic models dominated by industry, to those based on information and knowledge, and finally to a service-intensive economy.\textsuperscript{14,15} This new economy will be characterized by innovation, community coordination, and super-productivity. And this research suggests tackling the challenges of this transition to a service intensive world by combining pervasive computing, service-oriented computing, and community coordinated multimedia into Pervasive Service Computing. The vision of Pervasive Service Computing is illustrated as Fig. 3. This illustration shows the context of service domains which will surround the user’s everyday activities. The domains, presented as concentric circles around the user, are multimedia, device, network, discoverable service, and space.


Pervasive Service Computing facilitates the user’s everyday activities, e.g. learning at a campus, medication at a hospital, or shopping at a marketplace. It does this by providing service collaboration and service coordination frameworks, which intelligently facilitate the user to engage services smoothly by using mobile handsets through the above intra- and inter-domains.

Generally, the significance of service computing lies in managing and delivering services on the air to support user’s everyday activities, productivity, and innovation—anytime, anyplace, on any device, and on any network. Pervasive Service Computing evolves from pervasive computing, and is an enhanced technology towards promoting ever-expanding service-intensive user activities. Pervasive Service Computing unifies conventional pervasive computing by the explicit adoption of service orientation and a convergence of advanced ICT with service-oriented computing, mobile communication, multimedia informatics, and social networking. The accommodated service
collaboration framework strengthens the capacity for collaboration in pervasive systems which facilitate basic user activities. And the accommodated service coordination framework strengthens the capabilities for coordination between pervasive systems that facilitate the more complex user activities. The existing pervasive systems are monolithic, application-specific, and platform-dependent. They have constraints in terms of Internet-wide access, availability, and integration. This thesis leverages pervasive computing to take account of emerging service-oriented computing. It introduces a novel concept: Pervasive Service Computing, which integrates community coordination, context awareness, and service composition into a next generation of pervasive system developments.

Paper II. Pervasive Service Computing: Visions and Challenges

This paper designs a generic computing and communication architecture for Pervasive Service Computing (Fig. 4). This architecture categorizes the characteristics, and identifies the challenges of pervasive service computing through examining four typical service-oriented computing scenarios under the umbrella name of “Pervasive Campus.” The author had the main responsibility for technical content and writing of this paper.
The proposed computing architecture comprises the following six main components:

1. Data sensing allows various sensing systems to collect physical data anywhere, at anytime, and provide this data to the system’s upper layer for data modeling. For example, wireless sensor networks can be used for data
sensing in transportation systems (Chen et al. 2009). Also, physiological data sensors can inform networked systems for health care (Ryoo et al. 2005).

2. Data modeling applies formal descriptions and techniques for structuring data models and presenting them to the next program layer, of knowledge reasoning.

3. Knowledge reasoning attempts to provide the answers to a user’s query, or to produce data for supporting the next level up - service composition.

4. Service composition accommodates service collaboration and coordination to assemble services which facilitate the user’s activities.

5. Context-awareness refers to capability of mobile systems to sense their physical environment, or the situation in which they are used, and adapt their behavior accordingly.

6. Security management involves controls governing how these programs can be altered. Such security needs to meet four main requirements, namely confidentiality, integrity, availability, and authenticity. Confidentiality requires that the information be accessed only by authorized parties. Integrity requires that system assets can be modified only by authorized parties. Availability requires that system assets are available to the authorized parties. And authenticity requires that a host or service is able to verify the identity of a user.

The characteristics and structures of PSC are outlined as follows:

Service-orientation. A service-orientation approach to programming is based on the idea of composing applications by discovering and invoking network-available services, rather than by building new applications, or by using previously accessed applications to accomplish some task. PSC employs a service-oriented approach to develop and integrate (i.e. compose) applications from independent services, and implements them in a pervasive environment.

Explicit description of user’s activity. Daily activities involving questions such as “what to do?” “when to start?” and “how to do?” have to be explicitly described in Pervasive Service Composition for automatic management. Making an explicit activity description involves not only discovering services and composing services within one peer account, but also discovering and composing services within a range of service peer networks.

Pervasive nature. It is assumed that the user’s activities occur in a pervasive computing environment. The pervasive nature affects the way the service peers and the Web services are accessed, and the way they form peer networks.
**Semantics.** Service composition involves various kinds of information exchange. These include not only technical information for describing service input/outputs and interactions, but also managerial information for service peers and roles. A shared understanding and interpretation of this information is needed to smooth everyday activities within a pervasive computing environment. And this requires enabling semantic service discovery and composition, along with service peer discovery and coordination.

**P2P-based.** Traditional business integration emphasizes centralized service composition, and less peer-oriented composition. However, in pervasive computing all kinds of services, including networks, appliances, devices, and software, surround the users. The users expect these services to be distributed and connected, but they also expect that the system owners (i.e. service peers) are relatively decentralized and independent. Service peers are likely to run on their own policies and agreements with peer collaboration, and these policies are often flexible rather than rigid. Therefore decentralization is the significant feature in P2P-based composition.

**Trust-aware.** Trust management is defined by Sun and Denko (2007) as the activity of creating systems or methods that allow mutually reliant parties to make assessments and decisions regarding the dependability of potential transactions. These systems allow risk assessment, allow players or system owners to increase their own reliability, and allow them to correctly represent their own and their system’s reliability. In Pervasive Service Composition, trust management is concerned with aspects of Web services and with service peers. It attains a balance between availability and optimization of Web services and peer relations, and also a balance between automation and privacy control of applications.

**Wireless sensor network-based.** Pervasive Service Composition involves various devices which are placed in the user’s surroundings, sensing user activities, location, etc. Sensing and interacting with those entities efficiently via wireless sensor networks promises to make Pervasive Service Composition a reality in the context of application adaptations and automation.

Besides the notable challenges listed by Papazoglou et al. (2007) and by Dustdar and Papazoglou (2008), the author specifies the following impending research and implementation challenges for service composition in a pervasive environment.

1. **Context sensing in PSC.** Context sensing is the ability to collect appropriate contextual information. Information about a specific context is usually
gathered by various sensors and techniques, for example with GPS receivers, Bluetooth beacons, and RFID readers. The accuracy and fidelity of context are important for the context modeling and context-aware adaptation.

2. **Modeling in PSC.** Modeling in PSC aims to present an information representation mechanism for the Pervasive Service Computing environment, for example capturing data on persons, objects, locations, devices, networks, or services.

3. **Reasoning in PSC.** Context reasoning is the ability to compose context units and deduct higher-level context data. The context reasoning problem is discussed on two different levels (Sun & Wu 2006), namely context inference/recognition and context reasoning. The reasoning technology is rooted in AI. Commonly used reasoning technologies include rule-based logics and machine learning methods.

4. **P2P-based service composition.** P2P computing is changing the way people use the Internet by enabling a shift from the traditional client-server computing model to a decentralized model. The adoption of P2P into Pervasive Service Computing is expected to make better use of heterogeneous resources, speed up data access, eliminate single points of failure, improve scalability, and reduce network bottlenecks.

Paper III. PSC-RM: Reference Model for Pervasive Service Composition

This paper presents a generic Reference Model of Pervasive Service Composition (PSC-RM) for guiding PSC architecture design and implementation. The author had the main responsibility for the technical content and writing of this paper.

To design a model for PSC-RM, the author first investigates and presents a model for tracing a user’s generic activity, and then analyzes characteristics of PSC to envision PSC applications. Based on these applications, the author presents requirements and an initial design of PSC-RM. Then, the activity model is depicted as follows (Fig. 5):
Fig. 5. A Generic User Activity Model (III, published by permission of IEEE).

This model indicates a process to be followed in a pervasive service system. The phases of this process include the following:

1. A user’s generic activity starts with the goal planning phase. A goal usually consists of a set of tasks.
2. The task decomposition phase partitions the initial task into atomic entities, or subtasks, which represent operable task units. Each task unit is automated by one or more service units.
3. Following task decomposition, the consequent step is peer coordination or service collaboration. If a goal is simple, and only consists of small tasks which can be served by one service peer, then there is no need for peer coordination. Otherwise, peer coordination will respond to locate service peers. In this step, a global contract (e.g. definition of behavior, conditions, or constraints) and an observable process will be agreed among the involved parties. The agreement will concern matters such as peer roles and peer ownership. In dealing with these questions, we say that a service is owned by
a peer (an entity, a person, or an organization), and that it involves consuming a resource owned by a peer. Peers with the same rights to control services belong to a peer group.

4. Once peer coordination is set and services are identified, service collaboration comprises identified services that run for the coordinated service peers. Once composed, service logics are executed (step 5) as specified in service collaboration.

The author also designs PSC-RM (Fig. 6), which abstracts functional modules. This reference model is composed of three layers: an application layer, a PSC system layer, and a PSC enabling or enhancing layer.

The PSC-RM application layer illustrates the envisioned supporting applications, including Web service-oriented composition, P2P-based composition and service provision, context-aware and semantic service composition, multimodal user interface-based composition, and trust-aware service composition. Those applications are expected to facilitate the user’s daily activities in the context of pervasive service computing.

Fig. 6. Reference Model for Pervasive Service Computing (III, published by permission of IEEE.)
The PSC-RM system layer abstracts basic functional modules for meeting the needs of pervasive service composition, which include multimodal user interface, context management, core modules of peer coordination and service collaboration, service provisioning, providing a service repository and database, and using a Pervasive Service Composition Language (PSCL). The Multimodal User Interface (MUI) module enables the user to operate MUI transference in a pervasive computing environment, with responsiveness to the user’s preferences, privacy, and location. The context management module enables the user to compose applications which are flexible and adaptive to changes in the environment. Peer coordination and service collaboration are two more core modules in PSC. Peer coordination enables the user to compose applications within peer networks. Service collaboration enables the user to implement primitive service compositions in the context of service description, service discovery, service execution, etc. The service provisioning module provides backend engines for application composition and execution. The module will probably be built upon the existing service-oriented architecture, in order to back up and enhance core peer coordination and service collaboration in the pervasive computing environment. The service repository and database are responsible for storage and retrieval of services and data respectively.

Pervasive Service Composition Language (PSCL) provides and defines the minimal set of concepts and essential constructs necessary to specify peer coordination and service collaboration. PSCL is based on XML, and is defined by XML schema.

The PSC enabling and enhancing layer includes the following techniques.

Service-oriented PSC composes applications based on primitive SOA, contemporary SOA, and specific SOA. Primitive SOA (such as service description, registration, and discovery) promises to enable service collaboration. Contemporary SOA (such as service orchestration and choreography) promises to support peer coordination. Specific SOA is expected to extend SOA specifications for customizing Pervasive Service Computing, and supporting user’s everyday activity management.

P2P-based PSC is regarded as peer-coordinated service collaboration. P2P-based PSC emphasizes messaging management, service peer formation, and management. Messaging management is crucial for peer formation, as it provides the basic communication methods based on the messaging models. Peer management takes care of the formation of the peer group, the scale of the community, plus enabling the joining or leaving of peers.
Context-aware PSC facilitates the gathering of information from sources such as sensors and resource monitors. It performs interpretation of data, and carries out propagation of contextual information to interested parties in a scalable and timely fashion. Context-aware PSC will make use of information on activities in which the users are involved, their proximity to devices or services, plus their location, time of day, weather conditions, etc. This enables the application composition to adjust within changing environments, such as changing networks and devices. Ontology engineering and knowledge engineering are important in implementing context-aware PSC for user’s activity representation, semantic service discovery, and logic reasoning.

Mobility-aware PSC enables applications to detect changes happening to the available mobile resources, and to adopt corresponding approaches to deal with those changes. Connection and resource management are emphasized in mobility-aware PSC. Connection management should adopt different connection and reconfiguration mechanisms from fixed or from mobile distributed systems. Resource management monitors the limitations of mobile devices, such as their battery levels, CPU, memory, etc. Therefore specifications in PSC should be lightweight enough to avoid overloading the mobile devices.

Trust-aware PSC enables the user to share, interact with, and exchange data, resources, and services both frequently and invisibly, anytime and anywhere. Thus the open and dynamic environment raises challenges for security and trust management. Trust-aware PSC expects to enable services and service peers to cooperate effectively and reliably.

MUI-aware PSC allows the ubiquitous access to computing applications which users in pervasive computing environments demand. This raises possibilities for unanimously available user interfaces such as touch screens and gesture recognition. Currently, three major UIs are set in PSC interaction, i.e. touch-based UI, gesture-based UI, and speech-based UI.

Wireless sensor networked PSC targets enhancement of user wellbeing by facilitating the user’s everyday activities in a pervasive computing environment. PSC application compositions significantly depend on sensing and interpreting the user’s social signals, and the media in which the user engages.

3.2 Service-oriented Community Coordinated Multimedia

This section deals with the research issues related to service-oriented community coordinated multimedia, in terms of vision, conceptualization, modeling, and
implementation. Four publications, Paper IV, Paper V, Paper VI, and Paper VII contribute to this research. Already, popular social websites such as Wikipedia, Flickr, YouTube, and Joost provide functionalities of community management, multimedia content generation, and publishing. Now, further research on social network communities intends to discover communities (Bhukya 2010; Ichise et al. 2006), and detect or track communities in social networks (Yang et al. 2009).

Compared to these previous studies, the present research focuses on the support of multimedia services such as content annotation and aggregation. In addition, this research incorporates service-oriented computing into multimedia service design and development, and also introduces multimedia applications as web services. Compared with existing heavyweight multimedia processing tools such as the VideoAnnEx Annotation Tool (ANVIL), 16,17 and the General Architecture for Text Engineering (GATE), 18 the MAWS services are lightweight, platform-independent, and accessible in a Web-based, discoverable, and composable way.

**Paper IV. Community Coordinated Multimedia: Converging Content-Driven and Service-Driven Models**

This paper explores how human experience is being extended and enhanced by collaboratively consuming electronic or networked content, and multimedia-intensive services. The author had the main responsibility for the technical content and writing of this paper.

Recently, many web-based multimedia systems have emerged where the end user can upload, view, and share multimedia content (e.g. image, text, audio, video), or can consume traditional broadcast services (e.g. TV, movie, radio, music) over the Internet and mobile devices. These trends suggest a growing enrichment of experiences as people enhance and extend their consumption of multimedia-intensive content and services over online virtual communities. The author generalizes a vision for Community Coordinated Multimedia (Fig. 7).

Community Coordinated Multimedia presents a global scenario for addressing how multimedia experiences are extended and enhanced through consuming content and multimedia-intensive services within a community. As shown in Fig. 7, the initiated multimedia resource is progressively aggregated,

---

delivered, shared, and consumed within a CCM community, as shown in the diagram’s numbered order. The basic steps involved are (1) viewing a subscribed TV service, (2) creating a bundle with the aggregation of a clip of video, (3) sharing the bundle with members, (4) progressively creating the bundle by binding application software, and (5) broadcasting the bundle to members.

Fig. 7. Community Coordinated Multimedia Vision (IV, published by permission of IEEE).

Next, the author designs the CCM model (Fig. 8) converging content-driven and service-driven models in CCM. (1) In the content-driven CCM model, the fixed and mobile end users’ experiences are enhanced by collaboratively sharing user-generated content. The CCM system empowers the end user mainly in the context of content management, community management, and rights management. Concerning content management, the CCM interface allows end users to upload, annotate, aggregate, collect, and share content. Concerning community management, it allows community managers to authenticate end user memberships. Concerning rights management, the CCM interface involves content access control and subscription control. Other functionalities include user-side content rating, news feeds, and filtering of illegal, pornographic, or other offensive content.
(2) This service-driven CCM model also allows end users to enhance their experience by consuming content which is not directly owned or provided by CCM, but rather by a third party, such as a service provider. The service provider might be a professional media company or an amateur media producer. In this model, the CCM system takes the role of a service registry.

The author recognizes CCM as having the following characteristics:

1. **Web accessible.** By providing Websites and tool suites, the CCM system enables all end users to access content and services with the desired quality and functionality.

2. **Mobile.** In the CCM system, content can be consumed, created, analyzed, aggregated, or transmitted over mobile devices. Also, services can be requested, discovered, invoked, or interacted over mobile devices, including mobile computers, PDAs, or mobile phones.

3. **Content-driven.** The CCM system promises to support the end user in content creation, annotation, aggregation, and sharing in order to enrich and extend shared content. Also content-based automatic or semi-automatic applications (e.g. automatic content classification and annotation) in CCM are expected to facilitate content management.
4. **Service-driven.** The CCM system employs service-oriented computing and content delivery networking for delivering broadcasting content (e.g. music, TV, radio).

5. **Adaptable.** Content and services are consumable by any type of terminal devices.

6. **Participatory.** The end users aggregate content with information that is relevant within a common topic of interest, and send it to friends in the community who share that interest.

7. **Social network-oriented.** The CCM builds an online community for end users and service providers by providing the means for managing community memberships and members’ contacts.

8. **Personal.** The CCM attempts to provide the end user with individually customized content and service by managing the CCM user’s preferences and profiles.

Furthermore, the author addresses the challenges of implementing the CCM system with respect to community, content, services, networks, and Quality of Service (QoS) management. These challenges include the following:

1. **Attractive content delivery.** To make CCM community attractive means to promote the CCM system by grouping the right people, aggregating and distributing the content to the right focus groups, or to groups whose QoS parameters fit.

2. **Content authoring.** Enabling participants to generating their own content through writing, annotation, or aggregation, plays an important role in collaborative and aggregative community media.

3. **Content metadata management.** Content metadata is the extra descriptive data about the content, used for content indexing or categorization.

4. **Incorporation with service-oriented computing (SOC).** The service-driven CCM model is assumed to employ service-oriented architecture (SOA) as its basic infrastructure.

**Paper V. SCCM: Service-Oriented Community Coordinated Multimedia Architecture**

This paper applies a service orientation approach along with Web services technology. The result is to establish a service-oriented CCM architecture (SCCM) for tackling the requirements of scalability, discoverability, composability, layered
abstraction, QoS, and agility in distributed collaborative multimedia management. The author had the main responsibility for the technical content and writing of this paper.

Service-oriented CCM (SCCM) refers to an architectural approach to modeling the CCM system, which is analyzed and designed in compliance with principles of service orientation modeling (e.g. service discoverability, service autonomy, service compositability). This system is implemented by utilizing Web services technology.

The author outlines a design for two CCM service layers, i.e. CCM business service and application service layers (Fig. 9).

This layered structure enables the building of services that take CCM business, application, and agility into consideration. This structure involves the following layers:

1. The CCM business service layer represents the most fundamental CCM building block, encapsulating a distinct set of multimedia business logics within a well-defined functional boundary, and bringing the representation of corporate business models into the web services arena.
2. The CCM application service layer establishes the ground-level foundation that exists to express technology-specific multimedia processing functionality. CCM application services are responsible for representing technology and application logic.
Fig. 9. CCM Service Modeling Layers (V, published by permission of IEEE).

**Paper VI. Metamodeling for Community Coordinated Multimedia and Experience on Metamodel-Driven Content Annotation Service Prototype**

This paper introduces the CCM metamodel. The author had the main responsibility for the technical content and writing of this paper.

The CCM metamodel aims to provide a primitive constructor for instantiating, classifying, capturing, indexing, searching for information about the entities, and providing supports making for various multimedia delivery-related decisions. In the study of metamodeling methodology activities, the author makes the following observations:

1. *Unified Modeling Language* (UML) is used in designing models for human-to-human communication and for semantic transformation to enable software automation with wide vendor support.

2. The benefit of knowledge engineering approaches, such as CommonKADS, lies in a series of interrelated activities and techniques for ontology development.
3. The benefit of a Web resource modeling approach, such as Resource Description Framework (RDF or RDFS), lies in its flexibility in representing any data resource.

To develop a CCM metamodel (Fig. 10), the author proposes a combined metamodeling approach, which puts its emphasis on applying three kinds of methodology to the different phases of CCM metamodeling. These methodologies are (1) Knowledge engineering methodology, which is adopted for CCM ontology conceptualization during the analysis phase; (2) Model-Driven Architecture (MDA)/UML methodology, which is adopted for CCM metamodel development during the design phase; (3) An RDF modeling approach, which is adopted for CCM Schema generation during the implementation phase.

Fig. 10. An Overview of the CCM Metamodel (VI, published by permission of IEEE).
This paper introduces a concept of multimedia applications as Web services, and describes the multimedia content annotation end-user service. The author had the main responsibility for the technical content and writing of this paper.

Multimedia application as a Web service (MAWS) is a novel approach, consisting of analyzing, modeling, and implementing conventional multimedia applications in line with emerging Web services. The approach of MAWS aims to regard a conventional multimedia application as a hosted service, accessible to customers across the Internet. By partitioning the conventional heavyweight multimedia applications into lightweight pieces, and then providing them as Web services, MAWS takes advantages of mobile device access, heterogeneous network convergence, and system scalabilities. This alleviates the customer’s burden of multimedia system costs for maintenance, ongoing operation, and support.

Further, the author has participated in the design of a content annotation end-user service. The focus of this service is to bring semi-automation to the annotation process, and distribute computation over the Web. This content annotation program consists of a user interface and two web services, namely a structure extraction service and a people detection service. These last two services were developed in the CAM4Home project, and are utilized to develop a content annotation end-user prototype. The user interface invokes these two web services using Simple Object Access Protocol (SOAP)\textsuperscript{19} to carry out the automatic part of the annotation. Then the structure extraction service is used for extracting the shot information from a video file, which is utilized by the people detection service when detecting people-related concepts. These two Web services operate on CCM schema file, and produce a CCM document in a XML format. Fig. 11 presents the interaction between services in a sequence diagram.

3.3 Context-aware service composition

This section addresses the research issues related to context awareness and adaptive service composition. Paper VIII gives the contributions made in this research. Compared with existing non-web service-based context-aware applications, such as the context-aware battery management system (Ravi et al. 2008), safety critical systems (Bardram & Norskov 2008), or Talking Points (Stewart et al. 2008), this research introduces context awareness into service composition. It then investigates a general context-aware service composition architecture. In addition, this research categorizes three types of context-aware service composition applications, i.e. peer coordination adaptation, process service adaptation, and utility service adaptation.
Paper VIII. Context-Aware Pervasive Service Composition and its Implementation

This paper proposes the concept of Context-Aware Pervasive Service Composition (CAPSC), which aims to enable a pervasive system to provide user service compositions that are relevant for users in any situation. The author had the main responsibility for the technical content and writing of this paper.

This research is concerned to answer two questions: What is context-aware pervasive service composition? and how can we accommodate context-awareness in pervasive service composition?

The author broadens definitions of context and context-awareness in term of Pervasive Service Composition as the following.

Context encompasses not only information that can be used to characterize the situation of a user (namely user context) or of a peer (namely peer context), but also information that can be used to characterize the situation of a service (namely service context) or an activity (namely process context).

Context-aware computing is defined as the process of collecting, using and managing the various types of contextual information during a task session.

Context-awareness enables computer systems to adaptively initiate and manage an interaction session with the user by correctly modeling and reasoning with the various types of context.

Context-aware Pervasive Service Composition aims to support applications that respond to changes in context. Context-aware service composition can be classified into the following three adaptations:

1. Peer coordination, which is required when the group of participating service peers needs to be changed. To accomplish a complex task in a Pervasive Service Composition, it is necessary to seek and to group peers before composing and executing services. Many changes can occur in the midst of this process, such as peer join, peer disconnection, and peer leave. Context-aware peer coordination is expected to manage those changes in the peer coordination level.

2. Process service adaptation is required when the process services need to be changed. A process service introduces a parent level of abstraction to manage interaction, and ensures that service operations are executed in a specific sequence. In the midst of this process, many changes in process logic can
take place, such as time constrains, exception handling, or structured activities.

3. Utility service adaptation is required when the utility services need to be changed. This adaptation and reaction takes place in the utility service layer. Utility service reacts to context changes, such as changes in location, temperature, weather, proximity, bandwidth, etc.

To enable such applications, the author specifies the design process for building context-aware pervasive service composition (CAPSC) applications as follows.

Step 1, context specification, addresses the context-aware user activities and interactions occurring in a pervasive computing environment, and determines which context is required for participating in the consequential context-aware pervasive service composition.

Step 2, context acquisition, determines which sensors are required for reading and collecting the specified context.

Step 3, context modeling, interprets and structures the acquired data into a context model in a holistic and coherent way.

Step 4, context reasoning, is initiated by user queries or context changes. This process accesses and reasons (deductively or inductively) concerning the context model, and produces new knowledge for facilitating service composition.

Step 5, context-aware service composition, plays the key role in pervasive service composition. It is responsible for grouping service peers, for example through peer coordination, or service collaboration, and in assembling services.

Furthermore, the author designs the CAPSC architecture (Fig. 12). This architecture consists of the following modules: (1) context sensing, for perceiving contextual information and transmitting it to the rest of the system; (2) context modeling, for acquiring and working with the context to convert it into a usable form through interpretation; (3) context reasoning, for making decisions by examining contextual information or user requests against the predefined context model; (4) context storage, for storing current and historical context information; (5) composition adaptation, for choosing and performing context-aware service composition with respect to peer coordination, process adaptation, or utility service adaptation.
Peer coordination takes care of peer management within this composition process. This feature emphasizes processing and exchanging peer context through an efficient composition application. Process service adaptation considers the changes in the process logic, and adapts process services for maintaining a service composition session. Utility service adaptation examines the service context to detect changes in the situation, and indicate appropriate responses.

In the prototype implementation, all utility Web services are implemented with Java and deployed on the Glassfish 2.1 Application Server. Glassfish is chosen as a server because of its support of Enterprise JavaBeans (EJB) technology, and the BPEL Service Engine. The Glassfish server runs on 3GHz PCs, with 2 GB of memory. All utility Web services implement an identical Web Service Description Language (WSDL) interface, so they can be easily accessed as a part of a service composition. Information on user context, such as mood, event, music, etc. is mainly defined and processed for triggering adaptive composition. XML Schema is used to represent such context information. The reasoning engine is the Web service, which implements the logic of the prototype. It is implemented with WIN-Prolog,\(^\text{20}\) and wrapped with Java to the Web service. The reasoning engine is aware of the semantics for application, utility Web service addresses, and content. This reasoning engine has a knowledge base containing all information related to utility Web services, such as their addresses, and content.

rendered pictures, or music files. The knowledge base contains facts, and a set of rules.

In the successive research on context-aware computing, the author co-designed a Web services-based context-aware computing system—the Perception Framework for Supporting Development of Context-Aware Web Services (Gilman et al. 2011). This system provides conflict resolution, robustness and consistency support, clear separation between sensors and actuators, and a general reasoner. The Perception Framework is functionally divided into three main layers (Fig. 13).

![Fig. 13. The Conceptual Architecture of Perception Framework Gilman et al. 2011 (published by permission of Emerald Group Publishing Limited).]

The lowest, or sensing layer, provides mechanisms to retrieve environmental data using sensors and manual input. The semantic layer offers mechanisms for
context modeling and context reasoning. The highest, or control layer, offers maintenance functionality, such as resolving possible conflicts.

The Perception Framework prototype is implemented to have the following characteristics (Gilman et al. 2011): (1) It separates concerns with context sensing, context modeling or reasoning, conflict resolving, and managing history. (2) The prototype has a service-oriented design and allows subscribing or notifying context. (3) It is proactive and reactive, with an embedded context reasoner to deal with rule files. (4) The accommodated conflict resolver handles two possible types of conflicts, i.e. multiple-request conflicts, and multiple-usage conflicts. (5) The prototype provides an ontology-based context model, used for all system components. (6) The history manager registers the successful reasoning for system tracking.

Compared with existing prototypes proposed by other researchers (Bardram 2005; Dey 2000; Kim & Choi 2007; Román et al. 2002; Schmidt 2002), this Perception Framework is innovative in accommodating a unique general reasoner and conflict resolver. The general reasoner can handle various logic approaches. Thus, developers can select their preferred languages to construct their rules from various forms of contextual information. The conflict resolver guarantees consistency, as this mechanism tackles the challenge of large-scale deployment of pervasive services.

3.4 Service composition modeling, description, and implementation in the Cloud

This section addresses the research issues related to service composition modeling, description, and implementation in the Cloud. Papers IX, X, and XI offer a series of contributions to this field. Compared to the previous efforts in service composition reviewed in section 2.5, this thesis offers several innovations related to Cloud computing.

First, this research explores Web services and service composition in support of people’s daily activities in a pervasive computing environment—not just business applications in a stationary situation. Second, the author addresses the challenges of improving the characteristics of service federation, representability, and availability in service composition. Third, this research models service composition for solving and optimizing service coordination and service collaboration. Existing description languages (e.g. BPEL) for service composition are not comprehensive enough, and are more or less functionality-focused.
Therefore the author introduces an ontology methodology into service composition description, and proposes an Ontology-Driven Pervasive Service Composition (ODPSC) method for service-composition description. In addition, this thesis explores the availability of service composition via the Cloud, and introduces the concept of dynamic service composition in the Cloud. The author proposes a CM4SC middleware for accelerating service composition in the Cloud. This Cloud-based solution increases service composition availability, and reduces the users’ burden in using and managing Web services.

**Paper IX. Modeling Service Composition and Exploring Its Characteristics**

This paper examines and formalizes service composition and coordination. It investigates service composition as a process of value-added activities. The author had the main responsibility for the technical content and writing of this paper.

In an abstract description of a value-added process, people’s activities target certain goals. A goal is achieved by executing a task, and a complex task is achieved by executing a plan consisting of a sequence of subtasks, each achieving a sub-goal. When a task corresponds to one or more services, service composition creates the task network to achieve the goal, and controls execution by the task network. In collaboration, the task network is built up by one service provider. In coordination, the network is built up by multiple service providers. A service collaboration is a service composition occurring within one service party. And a set of service collaborations is defined as a two-tuple, \( \text{SCB} = \langle \text{Collaborator}, S_c, \rangle \), \( \text{Collaborator} \) is a set of logic units which specify relationships between services in \( S_c \). An instance of service collaboration \( \text{scb} \) in \( S_c \) is \( \langle \text{collaborator}, \{ s_i \} \rangle \), where \( \{ s_i \} \subseteq S_i \), \( S_i \) is the set of services administrated by \( S_c \). A service coordination is a service composition occurring among multiple service parties. The set of service coordinations is also defined as a two-tuple: \( \text{SCR} = \langle \text{Coordinator}, S_r, \rangle \), \( \text{Coordinator} \) is a set of logic units which specify relationships between services in \( S_r \). An instance of service coordination \( \text{scr} \) among \( S_r \) is \( \langle \text{coordinator}, \{ s_i, s_j, s_k \} \rangle \), where \( s_i \in S_i \), \( s_j \in S_j \), \( s_k \in S_k \).

Consequently the author formalizes the four solutions to service composition as the following:
Problem 1. Is there an available service collaboration $sch_i$ which consists of a set of services $\{s_i\}$ within one party $sp_i = \langle adm_i, S_i \rangle$ for achieving a goal $g_i$, i.e.

$$\exists sch_i = \langle collaborator, \{s_i\} > \{s_i\} \subseteq S_i \rightarrow g_i .$$

Problem 2. Is there an optimal service collaboration $sch_i$ which consists of a set of services $\{s_i\}$ within one party $sp_i = \langle adm_i, S_i \rangle$ for achieving a goal $g_i$, taking account of a set of constraints $c_i$ defined on service properties $\{p_i\}$, i.e.

$$I_{opt} = \arg\max_i F(sch_i)[c_i \geq A_{sup}] .$$

$A_{sup}$ is the value set as the user expected, e.g. satisfaction in service cost, accessing speed, etc.

Problem 3. Is there an available service coordination $scr_i$ which consists of a set of services $\{s_i\}$ among multiple parties $\{sp_i\}$ for achieving a goal $g_i$, i.e.

$$\exists scr_i = \langle coordinator, \{s_i\} > \exists i, j, s_i \in S_n ; S_j \in S_n \rightarrow g_i .$$

Problem 4. Is there an optimal service coordination $scr_i$ which consists of a set of services $\{s_i\}$ among multiple parties $\{sp_i\}$ for achieving a goal $g_i$, taking account of a set of constraints $c_i$ defined on service properties $\{p_i\}$, i.e.

$$I_{opt} = \arg\max_i F(scr_i)[c_i \geq A_{sup}] .$$

Paper X. Ontology-Driven Pervasive Service Composition for Everyday Life

This paper generalizes the concept of Ontology-Driven Pervasive Service Composition (ODPSC). This concept envisions an ontology-centric solution to flexibly facilitate everyday activity management by forming service peers, consuming individual Web services, and integrating Web services within or across service peers in a pervasive computing environment. The author had the main responsibility for the technical content and writing of this paper.

First, the author describes Ontology-Driven Pervasive Service Composition (ODPSC) as an advanced means of managing complex everyday activities in a pervasive computing environment. ODPSC can be classified as pervasive, task-
based, semantic, and P2P-based computing. The nature of pervasive computing requires ODPSC not only to discover and choose services before composing, but also to discover and regroup service peers dynamically when the situation changes. ODPSC is also expected to support user’s everyday activities by maintaining Web services and service compositions in a pervasive computing environment. This requires describing the tasks users want to accomplish and the composing services for performing these tasks. That is, it requires task-based computing. Some ODPSC services involve only a single party, while others involve multiple service parties, namely peers.

Second, the author analyzes requirements for Ontology-Driven Pervasive Service Composition, including requirements for peer coordination, such as relationship management, interaction management, and peer management, plus the requirements for service collaboration, for service description, for interaction description, message description, and for context description.

Third, the author develops a conceptual level ODPSC ontology (Fig. 14) of Pervasive Service Composition for capturing everyday activities, including business activity. The top concept composition contains two major workflow logic concepts, namely peer coordination and service collaboration, which make up a plan. Peer coordination specifies working peers, their roles and relationships, and their interactions for achieving planned goals in a decentralized and pervasive computing environment. Service collaboration specifies interactions with Web Services within a service peer for an executable process. Basic and structured activity elements, defined in BPEL specification, can be used to develop service collaboration ontology in ODPSC ontology.
Paper XI, Cloud Architecture for Dynamic Service Composition

This paper introduces dynamic service composition in the Cloud for Pervasive Service Computing environments. The author had the main responsibility for the technical content and for writing of this paper.

First the author investigates the situation of traditional service composition, which is inadequate in offering large-scale, on-demand service composition. Traditional service also suffers from high cost in service management and maintenance. Consequently, it cannot offer dynamic service composition in a pervasive service computing system. By taking advantage of Cloud computing, the author explores dynamic service composition in the Cloud. This can provide users with a Cloud-based Middleware for Dynamic Service Composition (or CM4SC) for supporting on-demand service composition over the Cloud.

Second, the author proposes an accelerated Cloud architecture for service composition in the Cloud, and designs the CM4SC “Composition as a Service” middleware layer, between the application layer and the platform layer. Fig. 15 presents the functional modules in the CAM4SC middleware, and its relations to the surrounding layers of software—the application layer above, and the platform layer below. The CM4SC middleware layer provides supporting services for the application layer to accelerate service composition. And the platform layer provides storage and registration services for the CM4SC middleware to realize loose-coupled service interaction and discovery.
The author next elaborates the CM4SC middleware design (Fig. 16). The CM4SC middleware encapsulates sets of fundamental services for executing the users’ service requests, and for performing service composition. These services include client service, process planning, service discovery, process generation, reasoning engine service, process execution, and monitoring. Together, these services form the “Composition as a Service” that is hosted in the Cloud.
Client service allows users to describe complex activities and task requirements in a high-level language. A process planner service transforms high-level task descriptions into formal and precise sets of activity descriptions and their associations. These activities are represented by Web services in the Cloud. A service search engine takes activity descriptions as input, and acquires the status of a corresponding service, including its availability, functionality, and interaction interface. The process generator service seeks to fulfill user tasks by composing Cloud services which are advertised by service providers. The process generator takes the activity description and the service discovery results as inputs, and then outputs a process model that describes a composite service. The process model describes a set of selected Cloud services, plus the control flow and the data flow among these services. The reasoning engine service takes contextual information from various resources as input, processes this information against the CM4SC rules in the storage, and provides decision information for the process planner and generator. The process execution service provides a composition execution.
environment. The process monitor service enables users to monitor the control and data flow in a composition.

The CM4SC middleware prototype was implemented utilizing Windows Azure Cloud platform, Windows Communication Foundation (WCF), and Windows Workflow Foundation (WF). The implementation verifies that the CM4SC middleware serves to reduce the burden of costs, and decreases risks for users and providers in using and managing these components. Meanwhile, CM4SC gives the user more opportunities to get computational services, and provides on-demand dynamic service composition. Cloud environments also leverage CM4SC services to automatically scale workloads. CM4SC accelerates rapid dynamic service composition. It increases the availability and scalability of service composition in ways that are hard to guarantee through conventional composition approaches.

3.5 Summary

This chapter presents the main contributions to the vision, conceptualization, and reference model of Pervasive Service Computing. It offers an end-user prototype for service-oriented community coordinated multimedia, with service-oriented content annotation. It proposes a design and prototype implementation of context-aware pervasive service computing, with modeling and description of service composition in the Cloud. The original papers on which this research is based are reprinted at the end of this thesis (in its printed form), with the permission of the original publishers. In the electronic version, these papers are not included, but are available via the original publishers.

This research results from the author’s work in two major projects. The first project was the CAM4Home project, which was funded by the Tekes (Finnish Funding Agency for Technology and Innovation) as part of the EUREKA ITEA 2 program (Information Technology for European Advancement). The second project was the Pervasive Service Computing (PSC) project, which was funded by the Academy of Finland as part of the MOTIVE Ubiquitous Computing and Diversity of Communication program.
4 Discussion, future work, and conclusions

4.1 Discussion

Computer technology has been essential to promote innovation and productivity through facilitating people’s everyday activities. At present, pervasive computing represents the major developmental trend in computer technology. This presents a new computing paradigm, which is characterized by integrating and embedding the technology of computers, digital multimedia, and communication into the user’s living environment. The potential arises for a new kind of invisible and ubiquitous computing, which allows the users to focus on the subject matter of a complex task itself, rather than on the complexities of operating computer system technology. Service-oriented computing introduces a new model for improving and increasing business productivity through building standardized and modularized applications, integrating those applications across heterogeneous platforms, and flexibly establishing a new value chain. These ever-growing Web services fit into the structure of service-oriented architecture. And these services are facilitating complex business management via the Internet to move forward from data management to process management.

The author recognizes how society’s transition to a service-centric economy involves embracing powerful service-oriented computing. And to realize a pervasive computing paradigm, the author introduces a novel Web service-centric solution to pervasive computing, called Pervasive Service Computing (PSC). This approach enables encapsulating resources into Web services, and building versatile Web-service sets that support and adapt to the user’s activities. Users can access these Web services anytime, anyplace, on any device, and on any network. In response to the questions defined in section 1.2, the author has conducted research and development to find solutions. The questions and answers are given below.

Q1. What is Pervasive Service Computing, what is its paradigm, and why is it emerging as a significant aspect of pervasive computing?

Answer: Pervasive Service Computing represents the next generation ICT, and is the major technological force for realizing a service-centric society, and for empowering people to improve their quality of life. Pervasive Service Computing involves unitizing the capabilities of community coordinated multimedia, context awareness, and service composition. These three capabilities empower Pervasive
Service Computing to deliver on-the-air, multimedia-rich services which support user activities—anytime, anyplace, on any device, and on any network.

To explore the Pervasive Service Computing paradigm, the author first creates scenarios for a “pervasive campus” in the context of pervasive computing and Web services. These scenarios examine characteristics of Pervasive Service Computing, analyze the requirements for implementing pervasive service applications, and also discuss the challenges of implementing Pervasive Service Computing. The developed reference model and a user’s generic activity model guide research and development towards Pervasive Service Computing.

Q2. What is community coordinated multimedia, what is Multimedia Application as a Web Service (MAWS), and how do we apply such Web services into community coordinated multimedia application developments?

Answer: Community coordinated multimedia aggregates collective intelligence through accelerating multimedia process, and presents users with multimedia-rich information via a virtual community. Such community coordinated multimedia extends and enhances the user’s experience through providing various multimedia services for annotating and aggregating content. Multimedia application as a Web service breaks down conventional heavyweight, monolithic multimedia processing tools into pieces, and encapsulates those pieces into Web services. The MAWS method facilitates multimedia service and application developments by embracing features exhibited by Web services, e.g. Web-based accessibility, discoverability, and composability. The developed CCM service model and metamodeling approach are applied into the multimedia content annotation end-user service development.

Community coordinated multimedia enables collaborative activity, which empowers people to experience the real world and share it with others. In exploring the means to extend this service, the author first examines a scenario for collaborative multimedia sharing, and reviews the characteristics and challenges of CCM. This leads to proposals for a community coordinated multimedia model for converging the content-driven and service-driven models. Second, the author studies CCM metamodeling for end users and commercial providers, and proposes a hybrid CCM metamodeling approach which takes advantage of previously developed methodologies. In this approach, the author co-develops a metamodel for the CCM paradigm with a CCM bundle metadata model, and offers a CCM bundle metadata in RDF and CCM metamodel schemas. Furthermore, the author generalizes service-oriented CCM in designing a SCCM
model, which focuses on identifying and specifying CCM business and application services. The author emphasizes a concept of multimedia application as a Web service, which encapsulates traditional multimedia applications into Web services. Multimedia, as a service, assists users in the task of extracting, annotating, retrieving, and mining multimedia content by taking advantages of service discovery, service accessibility via the Internet, and service composition. The prototype of a CCM content annotation end-user service shows that the CCM enables and facilitates the extensive and emerging multimedia applications.

Q3. What is context awareness in Pervasive Service Computing, what is context-aware pervasive service composition, and how can we combine these features in pervasive service systems?

Answer: The research defines context as any information characterizing the situation of a task session or interaction between a user and his/her service world. Context awareness is composed of a series of computing processes—of collecting, using, and managing context during a task session. Context-awareness enables computer systems to adaptively initiate and manage an interaction session with the user, by correctly modeling and reasoning with the various types of context.

Context-aware Pervasive Service Composition supports application developments. And those applications are able to react to changing context. The CAPSC architecture intends to accommodate both context-aware computing and service composition. Respectively, context-aware computing comprises context sensing, context modeling, and context reasoning. Context-aware service composition is divided into three types of applications: peer coordination, process service adaptation, and utility service adaptation. Through software engineering, this thesis implements a Web service development environment involving the NetBeans IDE, BPEL service engine, and WIN-Prolog reasoning tools in order to produce a concept-proof prototype which demonstrates process service adaptation.

In this study of context awareness and service composition, the author first broadens the definition of context to include both the process context and context awareness, and then proposes context-aware pervasive service composition. Then the author categorizes context-aware service composition applications into peer coordination adaptation, process service adaptation, and utility service adaptation. After examining the requirements of these applications, the author proposes an architecture for context-aware pervasive service composition. An initial prototype is designed and implemented as the proof of the concept's workability. In the successive efforts on context-aware computing, the author co-designs the context
perception framework (Perception Framework). This framework clearly separates the activities of sensors, actuators, and a general reasoner. The framework is service-oriented, and provides conflict resolution, robustness, and consistency support. The Perception Framework is also verified with a prototype implementation.

**Q4. How do we describe service composition in the context of Pervasive Service Computing, what is dynamic service composition in the Cloud, and how do we enable pervasive service composition in the Cloud?**

Answer: This thesis defines service composition as a process of value added activities. The research deconstructs service composition into service coordination and collaboration. In service collaboration, the value-added chain is built by integrating services within one service provider. In service coordination, the value-added chain is built by integrating services within multiple service providers. This thesis then analyzes description requirements for service composition, in the contexts of peer coordination, service collaboration, service description, interaction description, message description, and context description. From this investigation, the author develops an ODPSC ontology model for Pervasive Service Composition.

Dynamic service composition in the Cloud aims to improve scalability and availability of service composition by embracing Cloud computing. The thesis introduces a method for composition as a service into Cloud architecture, and develops CM4SC middleware for dynamic service composition in the Cloud. The resulting concept-proof prototype presents a novel way for providing scalable and on-demand dynamic service composition.

In the study of service composition, the author first examines a scenario which takes account of service composition within one party and among multiple parties. Second, the author models and formalizes service composition with categories of service collaboration and coordination. To describe the user’s everyday activities, the author analyzes requirements specifying peer coordination, service collaboration, service description, interaction description, message description, and context description. This leads to the design of an ontology model for capturing everyday activity, with an overview of existing approaches to Web service composition and ontology methodologies.

In order to research and develop Cloud architecture for dynamic service composition, the author applies “Composition as a Service” into Cloud computing, and proposes a CM4SC middleware layer in the Cloud architecture for supporting
dynamic service composition. The implementation of the CM4SC prototype reduces the burden of costs and risks which users and providers face in managing those components. Meanwhile, CM4SC also gives the user more opportunities to access computational services and provides on-demand dynamic service composition.

This research leads to a final practical question:

How can we automatically and efficiently develop and manage service compositions to adaptively support community scale user activities?

Answer: We witness that people experience their world as mobile and dynamic, due to their increasingly complex surroundings and tasks. On one hand, software-intensive applications facilitate people in coping with complex daily activities. On the other hand, the traditional monolithic and platform-dependent applications are burdensome to use and manage. These applications are usually costly to install and maintain. But context-aware and pervasive computing embeds microcomputers into environmental objects, where they process data and interact with the user invisibly. To a certain degree, context-aware and pervasive computing allows the users to focus directly on tasks to be done, instead of being burdened by complex requirements for operating computing devices. However, existing practices and applications are insufficiently sophisticated to support complex user tasks. To overcome this, service-oriented computing enables software development, integration, and maintenance in a discoverable and composable way. Contemporary service composition facilitates large-scale application development by composition of lightweight atomic services. This forms a value-added chain for supporting complex user-task computing and information retrieval. Moreover, community coordinated multimedia enriches information retrieval through aggregating collective intelligence via virtual communities across the Internet.

In order to empower advanced context-aware computing, service-oriented computing, community coordinated multimedia, and to meet the requirements for rapid software development, the author creates a Pervasive Service Computing paradigm. This paradigm includes three aspects: community coordinated multimedia, context awareness, and service composition. For service composition the thesis introduces a powerful, rapid application development method. The study on service composition in the Cloud convinces us that CM4SC-accelerated Cloud architecture is a feasible means to realize automatic service composition via the Internet. Context-aware computing takes account of contextual
information during a user task session, and adaptively initiates an interaction session with the user. The developed context-aware service composition prototype demonstrates that service composition can adapt to changing user task requirements. Community coordinated multimedia introduces a way of enriching and extending information through aggregating collective intelligence. The developed multimedia content annotation end-user services enable users to generate, annotate, and aggregate multimedia content within a virtual community, so that user-desired information is enriched and extended for an enhanced user experience.

By accommodating the three properties of pervasive computing, Pervasive Service Computing allows rapid large-scale service/application developments for supporting user tasks. It allows services to adapt to specific user tasks. It allows users to focus on the content of complex tasks rather than on the complexities of operating the software. And it enables user communities to enrich desired information, and enhance their shared user experience. Therefore, Pervasive Service Computing is an emerging technology for promoting a service-centric society, improving the quality of life, and enhancing productivity.

### 4.2 Future work

Adoption of these new technologies and development of new applications will be ongoing tasks for the future of Pervasive Service Computing. Future work in the development of Pervasive Service Computing will focus on automatically, transparently, and intelligently managing integrated computing and communication resource services across networks, e.g. the Internet. This work will happen through developing and enhancing middleware technologies in the context of community coordinated multimedia, context awareness, and service composition. Particular aspects of this ongoing research will include the following:

- Research on multimedia as a service middleware for accommodating generic multimedia services, and for enhancing and enriching the user multimedia experience via networks, e.g. the Internet.
- Research on context-aware middleware for accommodating generic services for context sensing, modeling, and reasoning in Pervasive Service Computing. This research will explore the possibilities for collecting contextual information gathered by different sensors and techniques, and for exploring
appropriate information-representation mechanisms for modeling Pervasive Service Computing environments. These mechanisms can be designed to accommodate expanding capabilities for composition of context units and deduction of higher-level context data, such as “what, who, where” information about objects, and “how” information concerning actions and relationships between things. The results will be an ever-more comprehensive perspective which unites diverse sources of contextual information.

- Research on service composition middleware for accommodating the lifecycle of service composition in the Pervasive Service Computing environment, including user-oriented task description, autonomous workflow generation, QoS-aware service discovery, and adaptive-service composition.

- Research on semantics for intelligent pervasive service computing, including semantics-based community coordinated multimedia, semantics-based context awareness, and semantics-based service composition.

4.3 Conclusion

To explore Pervasive Service Computing, the author focuses on addressing its three essential characteristics—community coordinated multimedia, context awareness, and service composition. Community coordinated multimedia presents a global scenario for addressing how multimedia experiences are extended and enhanced, by accessing content and multimedia-intensive services within a virtual community. In this dissertation, the author introduces the concept of community coordinated multimedia, analyzes the requirements for implementing CCM, designs a community coordinated multimedia model for converging content-driven and service-driven services, and then develops a community coordinated multimedia metamodel. Moreover, the author introduces a concept of multimedia applications as Web services, and participates in the design of a multimedia content annotation end-user service. The prototype of a content annotation service shows that CCM enhances and extends the end user’s multimedia experiences by delivering multimedia-intensive content and services through online virtual communities.

In this thesis, the author regards context as more than information characterizing a participating entity. Context is also information utilized for initiating, controlling, and maintaining a task session. The author categorizes context into user context, peer context, process context, physical context,
service context. The proposed model seeks to collect, use, and manage all these aspects of context in a context-aware computing process.

Context-awareness in Pervasive Service Computing enables computer systems to adaptively initiate and manage interaction sessions with their users. The proposed model operates through an interaction between three-levels of adaptive composition: peer coordination, process service adaptation, and utility service adaptation. Services composition is the essential technique in service-oriented architecture, which is traditionally used to provide value-added business services through integrating basic Web services provided by other organizations. In the thesis, the author introduces service composition into pervasive computing for end users, in order to facilitate their everyday activities. This model formalizes service composition with the categories of service collaboration and coordination. It also explores ontology-driven pervasive service composition, and implements dynamic service composition in the Cloud.
References


Original papers


Reprinted with permission from IEEE (I-VII, IX), Springer (VIII, X) and IGI Global (XI)

Original publications are not included in the electronic version of the dissertation.
383. Lasanen, Kimmo (2011) Integrated analogue CMOS circuits and structures for heart rate detectors and other low-voltage, low-power applications


385. Kortelainen, Jukka (2011) EEG-based depth of anesthesia measurement: separating the effects of propofol and remifentanil

386. Turunen, Helka (2011) CO₂-balance in the atmosphere and CO₂-utilisation: an engineering approach

387. Juha, Karjalainen (2011) Broadband single carrier multi-antenna communications with frequency domain turbo equalization


396. Koskela, Timo (2011) Community-centric mobile peer-to-peer services: performance evaluation and user studies


398. Teirikangas, Merja (2011) Advanced 0-3 ceramic polymer composites for high frequency applications
Jiehan Zhou

PERVASIVE SERVICE COMPUTING: COMMUNITY COORDINATED MULTIMEDIA, CONTEXT AWARENESS, AND SERVICE COMPOSITION