A SERVICE ORIENTED REFLECTIVE MIDDLEWARE FOR PERVASIVE COMPUTING

by

Ahmet Volkan GÜREL

August 2008

A SERVICE ORIENTED REFLECTIVE MIDDLEWARE FOR PERVASIVE COMPUTING

by

Ahmet Volkan GÜREL

A thesis submitted to

The Graduate Institute of Sciences and Engineering

of

Fatih University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Engineering

August 2008
Istanbul, Turkey

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Cevdet MERİÇ
Head of Department

This is to certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Halûk GÜMÜŞKAYA
Supervisor

Examining Committee Members

Assoc. Prof. Dr. Halûk GÜMÜŞKAYA

Assist. Prof. Dr. İhsan Ömür BUCAK

Assoc. Prof. Dr. Onur TOKER

It is approved that this thesis has been written in compliance with the formatting rules laid down by the Graduate Institute of Sciences and Engineering.

Assist. Prof. Dr. Nurullah ARSLAN
Director

August 2008

A SERVICE ORIENTED REFLECTIVE MIDDLEWARE FOR PERVASIVE COMPUTING

Ahmet Volkan GÜREL

M. S. Thesis - Computer Engineering
August 2008

Supervisor: Assoc. Prof. Dr. Halûk GÜMÜŞKAYA

ABSTRACT

Pervasive Computing is getting much practical and applicable to daily life due to the technological advances in mobile computing. The needed hardware technology is available as advanced small mobile communication devices, short and long distance wide band wireless networking technologies and smart sensors. Along with these technological advances, new software techniques and mobile services for mobile devices, and indoor and outdoor distributed and wireless environments are required.

In this thesis, software architectures for small mobile communication devices are proposed for system independent and portable mobile applications and systems for a context-aware system. The wireless network packet measurements and architecture performance analyses are given for the proposed architectures. In these analyses, using basic TCP socket connections, RMI distributed object technology and service oriented approaches, which are the representatives of three important generations in distributed systems, client/server architectures are studied and the time analyses for different wireless network connections are presented. As a result of these analyses, software architectures and technologies which can be used on today's mobile devices for different applications are evaluated.

Keywords: Pervasive computing, service oriented architecture, middleware, small mobile devices, wireless networks, TCP, RMI, Web Services, performance evaluation.

YAYGIN HABERLEŞME İÇİN SERVİS ODAKLI AKSETTİRİCİ ARAKATMAN

Ahmet Volkan GÜREL

Yüksek Lisans Tezi – Bilgisayar Mühendisliği
Ağustos 2008

Tez Yöneticisi: Doç.Dr. Halûk GÜMÜŞKAYA

ÖZ

Son yıllarda mobil bilgi işleme (mobile computing) alanındaki teknolojik gelişmeler sayesinde, yaygın bilgi işleme (pervasive computing) çok daha pratik olmuş ve günlük hayata uygulanabilir hale gelmiştir. Gelişmiş taşınabilir haberleşme cihazları, kısa ve uzun mesafe geniş bant kablosuz ağ teknolojileri ve akıllı sensörler gibi ihtiyaç duyulan donanımlar artık kolayca ulaşılabilir durumdadır. Bu gelişmeler ışığında taşınabilir cihazlar ile bina içi ve bina dışı dağıtık kablosuz ortamlara yönelik yeni yazılım teknikleri ve mobil servislerin geliştirilmesi gerekmektedir.

Bu tez çalışmasında, sistemden bağımsız çalışan, taşınabilir ve ortamdan haberdar mobil uygulamalar ve sistemler için yazılım mimarileri sunulmaktadır. Bu mimariler için kablosuz ağ paket ölçümleri ve mimari performans analizi verilmektedir. Bu analizlerde, dağıtık sistemler ve sunucu istemci mimarisindeki üç önemli neslin temsilcisi olan temel TCP soket bağlantısı, RMI dağıtık nesne teknolojisi ve servis odaklı yaklaşım kullanılarak farklı kablosuz ağlar için zaman analizi yapılmıştır. Bu analizler sonucunda günümüzün mobil cihazlarında kullanılabilecek yazılım mimarileri ve teknolojileri değerlendirilmektedir.

Anahtar Kelimeler: Yaygın haberleşme, servis odaklı mimari, arakatman, küçük mobil cihazlar, kablosuz ağlar, TCP, RMI, Web Servis, performans ölçümü.

DEDICATION

To my parents

ACKNOWLEDGEMENT

I am grateful to my advisor, Assoc. Prof. Dr. Halûk GÜMÜŞKAYA, for giving me the opportunity to study and conduct research for constitution of this thesis. I appreciate his patience, understanding and encouragement throughout my thesis study. As a mentor his level of commitment to research and teaching, as well as his energy, creativity, and enthusiasm are inspirational.

Next, I would like to thank the examining committee members of my thesis, Assist. Prof. Dr. İhsan Ömür BUCAK and Assoc. Prof. Dr. Onur TOKER for their time on evaluating my work and their valuable comments.

And I would like to thank all my friends and colleagues at Fatih University, for their friendship and collaboration and companionship throughout this journey in graduate school life.

Finally, I owe sincere thanks to my parents for their support throughout the years.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

LIST OF SYMBOLS AND ABBREVIATONS

SYMBOL/ABBREVIATION

AP	Access Point
API	Application Programming Interface
BT	Bluetooth
CORBA	Common Object Request Broker Architecture
CPU	Central Processing Unit
EDR	Enhanced Data Rate
FCAPSYS	Frameworks for Context-Aware Pervasive Systems
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile communication
GUI	Graphical User Interface
HCI	Human Computer Interaction
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IrDA	Infrared Data Association
ISM	Industrial Scientific Medical
J2ME	Java Platform, Micro Edition
JCAF	Java Context Aware Framework
JDK	Java Development Kit
JRE	Java Runtime Environment
JVM	Java Virtual Machine
LAN	Local Area Network
MAC	Media Access Control
MIT	Massachusetts Institute of Technology
PARC	Palo Alto Research Center
PC	Personal Computer
PDA	Personal Digital Assistant
RMI	Remote Method Invocation
RPC	Remote Procedure Call
RTP	Real-time Transport Protocol
RTSP	Real Time Streaming Protocol
SDK	Software Development Kit
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SOC	Service Oriented Computing
SOWCAS	Service Oriented Wireless Context Aware System
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UMPC	Ultra-Mobile PC
UMTS	Universal Mobile Telecommunications System
USB	Universal Serial Bus
Wi-Fi	Wireless Fidelity

WiPoD	Wireless Position Detector
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WS	Web Services
XML	Extensible Markup Language

CHAPTER 1

INTRODUCTION

The use and capabilities of small mobile communication devices that include many all-in-one features, such as Pocket PCs, smart mobile phones, and smart wireless sensors, that have become embedded in everyday life are growing rapidly. In addition, wireless network solutions, such as IEEE 802.11abg/n, Bluetooth, ZigBee, Wimax are now available that enable such devices to integrate, with each other and with larger scale computing devices, into pervasive computing systems.

As a result of these developments, distributed systems have become widely pervasive, and are having a tremendous impact on various domains of human activity. Today's distributed systems range from ad-hoc networks comprising of tiny sensor devices, to overlay networks such as peer-to-peer systems, to massive web farms of powerful servers.

Advances in mobile devices, wireless access technologies, and distributed systems result in new developments in pervasive computing and context-aware systems. All these advances require new software techniques and mobile services for mobile devices, and indoor and outdoor distributed and wireless environments.

In this study, software technologies and architectures for system independent and portable network programs and context-aware pervasive distributed systems that can be used for Pocket PCs and mobile phones are presented and performance analyses are given. In these analyses, using TCP (Transmission Control Protocol) socket connections, RMI (Remote Method Invocation) distributed object technology and service oriented approaches, which are the representatives of three important generations in distributed systems, client/server architectures are studied and the time analyses for different wireless network connections are presented.

1.1 THESIS ORGANIZATION

This thesis is structured into six chapters. This chapter, Chapter 1, gives an introduction to the thesis subject.

The next chapter, Chapter 2, begins with a literature survey of Pervasive Computing which explains the history, principles and evolution of this new era from the beginning to the present, and it continues with a concise background of the Service Oriented Architecture (SOA).

Chapter 3 gives information about a number of important projects in Pervasive Computing, and explains how we used them in our system.

Chapter 4 introduces a context aware system called Service Oriented Wireless Context Aware System (SOWCAS). This system was designed and developed over the years for the Fatih University Campus.

Chapter 5 presents a detailed performance analysis of TCP, RMI and Web Services, which are the representatives of three important generations in distributed systems, on several wireless networks and hardware configurations. The main contribution of this thesis to the SOWCAS is the perfomance evaluation of different single-hop wireless network connections and understanding of currently available software networking technologies for small mobile devices.

Finally Chapter 6 summarizes what we have done in this thesis.

This study was supported by Fatih University Research Fund as a part of Frameworks for Context-Aware Pervasive Systems (FCAPSYS) Project (Grand No. P50050702).

CHAPTER 2

BACKGROUND AND MOTIVATION

2.1 UBIQUITOUS COMPUTING AND PERVASIVE COMPUTING

Ubiquitous computing began in the Electronics and Imaging Laboratory of the Xerox Palo Alto Research Center (PARC) (Weiser et al., 1999). Mark Weiser, Chief Technologist of the Xerox PARC, coined the term in 1988 to describe a future in which invisible computers, embedded in everyday objects, replace PCs. He wrote some of the earliest papers on the subject, largely defining it and sketching out its major concerns, both alone and with John Seely Brown (Wikipedia Ubiquitous Computing, 2008).

In his seminal 1991 paper (Weiser, 1991), Weiser envisioned a world in which computing is so pervasive that everyday devices can sense their relationship to us and to each other. They could, thereby, respond so appropriately to our actions that the computing aspects would fade into the background. Underlying this vision is the assumption that sensing a broad set of physical phenomena, rather than just data input, will become a common aspect of small, embedded computers and that these devices will communicate with each other (as well as to some more powerful infrastructure) to organize and coordinate their actions (Estrin et al., 2002).

This was a vision too far ahead of its time, because the hardware technology needed to achieve it simply did not exist. The implementation attempted by Weiser and his colleagues at Xerox PARC fell short. After nearly two decades of hardware progress, many critical elements of ubiquitous computing that were not available in 1991 are now available as handheld and wearable computers, wireless networks and smart sensors (Satyanarayanan, 2001).

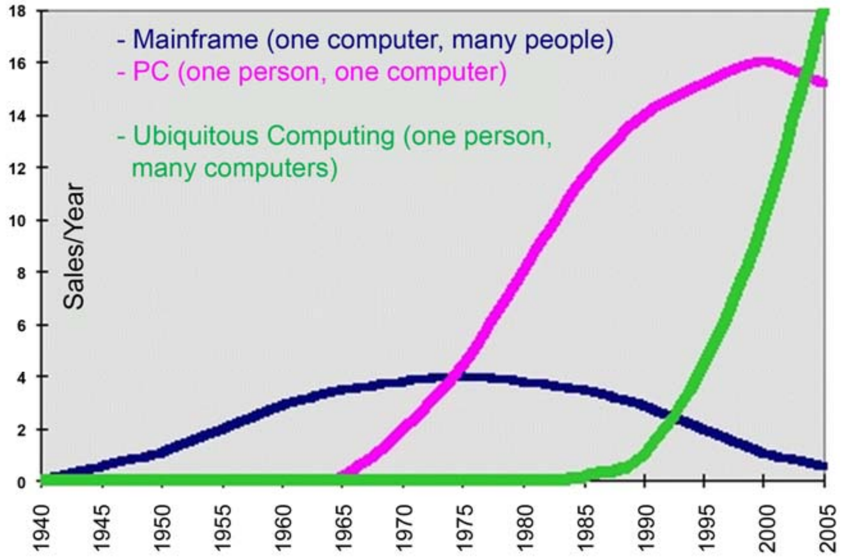

Figure 2.1 The major trends in computing.

As shown in the Figure 2.1, Weiser and his colleagues at PARC believed that ubiquitous computing would gradually emerge as the dominant mode of computer access over the next 20 years. Like the personal computer, ubiquitous computing would produce nothing fundamentally new, but by making everything faster and easier to do, with less strain and fewer mental gymnastics, it would transform what is apparently possible. Ubiquitous computing names the third wave in computing. First were mainframes, each shared by lots of people. Now we are in the personal computing era, person and machine staring uneasily at each other across the desktop. Next comes ubiquitous computing, or the age of calm technology, when technology recedes into the background of our lives.

2.1.1 Ubiquitous Computing vs. Pervasive Computing, and Research Areas

Five years later than Xerox PARC's research labeled as Ubiquitous Computing, the term Pervasive Computing emerged from research at IBM during 1996-97, embracing the vision of computing services available anytime, anywhere and on demand (Schechter, 1999). Initial focus of Ubiquitous Computing was not on making infrastructure available everywhere, but preached ubiquity as a notion similar to the availability of natural resources and utilities such as electricity, water and air. Today a convergence of themes is noticed such that the technical infrastructure advancement principles of Pervasive Computing complement the user centric opinions of the Ubiquitous Computing community. The major difference in philosophies has been that

Pervasive Computing was started with the initiative to exploit the existing wide scale deployment of computing technology, while Ubiquitous Computing's initiatives were to effectively make this complex mass of technology transparent to the human user's, especially those with limited technical "know-how".

Recently, the term Pervasive Computing is used more commonly than Ubiquitous Computing. Other terms used for Ubiquitous Computing are Calm technology, things that think, everyware, pervasive internet, ambient intelligence, proactive computing, and augmented reality. The term Pervasive Computing will be used for the rest of this work.

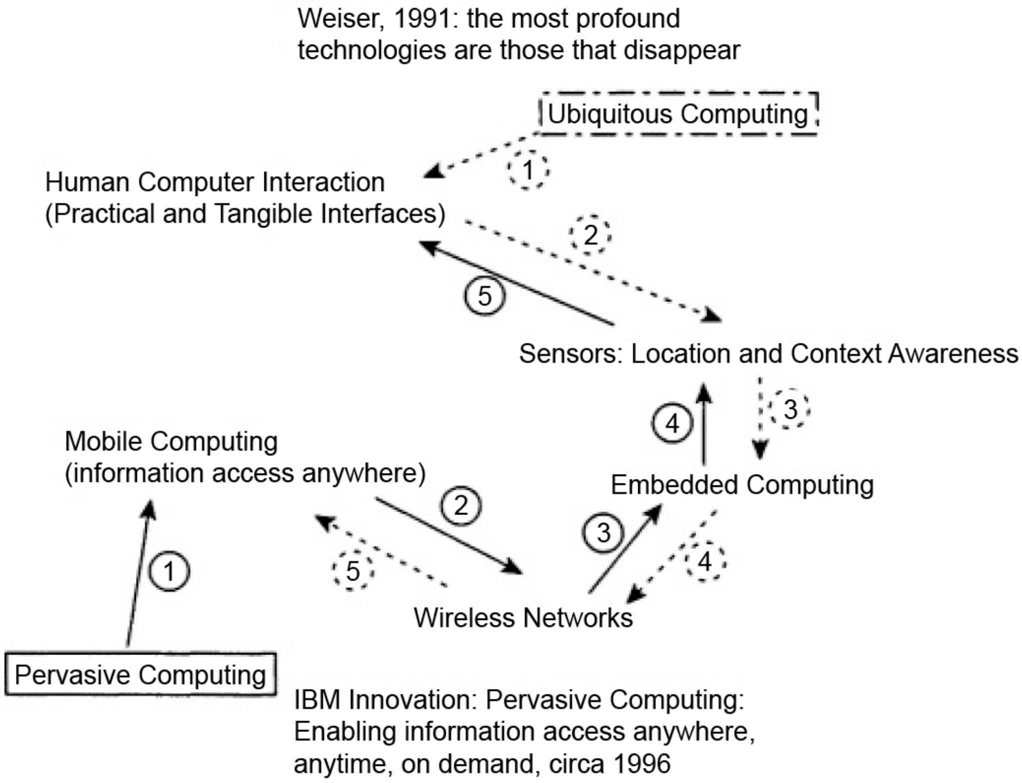

Figure 2.2 Convergence of Pervasive Computing and Ubiquitous Computing. Ubiquitous Computing was initiated with a user-centric methodology, while Pervasive was based on a bottom-up strategy for exploiting technology (Robinson et al., 2004).

Pervasive Computing is comprised of five common research areas: Mobile computing, wireless networking, embedded computing, context awareness with sensor technology, and human computer interaction (Robinson et al., 2004).

2.1.1.1 Mobile Computing

Mobile Computing or Nomadic Computing allows people to access information anywhere. Nowadays commercial examples of mobile computers like laptops, Pocket PCs, smart mobile phones, and UMPCs are quite affordable and they became widely pervasive. Other examples include wearable computers which are built into the garments of the user. There are some limitations of mobile computing: Insufficient bandwidth, security standards, power consumption, transmission interferences, potential health hazards, and human interface with device (Wikipedia Mobile Computing, 2008).

2.1.1.2 Wireless Networks

Wireless Networks have been developed to eliminate cabling and wiring (switches, adapters, plugs and connectors) while facilitating access to information. Emerging wireless networking technologies support long-range (e.g. GSM, GPRS, UMTS), local-area (e.g. IEEE 802.11 a/b/g/n), and short-range (e.g. IrDA, ZigBee, Bluetooth) communications.

2.1.1.3 Embedded Computing

An embedded computer is a special-purpose system designed to perform one or a few dedicated functions, often with real-time computing constraints. It is usually used as part of a complete device including hardware and mechanical parts. In contrast, a general-purpose computer, such as a personal computer, can do many different tasks depending on programming. Embedded systems control many of the common devices in use today (Wikipedia Embedded Systems, 2008). Embedded Computing has been considered as contributory to Pervasive Computing, while many pervasive systems are built by creating a distributed network of micro nodes each with a special purpose. A more powerful system is still needed to coordinate and make sense of the interaction between these small computers.

2.1.1.4 Context-Aware Computing

A pervasive computing system that strives to be minimally intrusive has to be context-aware (Satyanarayanan, 2001). Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves. A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user’s task (Dey, 2001).

A user’s context can be quite rich, consisting of attributes such as physical location, physiological state (e.g., body temperature and heart rate), emotional state (e.g., angry, distraught, or calm), personal history, daily behavioral patterns, and so on (Satyanarayanan, 2001).

2.1.1.5 Human Computer Interaction

Human–computer interaction (HCI) is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them (ACM SIGCHI). It is often regarded as the intersection of computer science, behavioral sciences, design and several other fields of study. Interaction between users and computers occurs at the user interface, which includes both software and hardware (Wikipedia Human-Computer Interaction, 2008). A basic goal of HCI is to improve the interaction between users and computers by making computers more user-friendly and receptive to the user's needs. A long term goal of HCI is to design systems that minimize the barrier between the human's cognitive model of what they want to accomplish and the computer's understanding of the user's task.

2.1.2 Evolution of Pervasive Computing

Mobile computing and pervasive computing represent major evolutionary steps in a line of research dating back to the mid-1970s. Figure 2.3 illustrates this evolution from a systems-centric viewpoint. New problems are encountered as one moves from left to right in this figure. In addition, the solutions of many previously-encountered problems become more complex — as the modulation symbols suggest, this increase in complexity is multiplicative rather than additive. It is much more difficult to design and implement a mobile computing system than a distributed system of comparable robustness and maturity; a pervasive computing system is even more challenging. As

Figure 2.3 indicates, the conceptual framework and algorithmic base of distributed systems provides a solid foundation for mobile and pervasive computing.

Mobile computing was born in the early 1990's with the advent of full-function laptop computers and wireless LANs. Although many basic principles of distributed system design continued to apply, four key constraints of mobility forced the development of specialized techniques:

- Unpredictable variation in network quality.
- Lowered trust and robustness of mobile elements.
- Limitations on local resources imposed by weight and size constraints.
- Concern for battery power consumption.

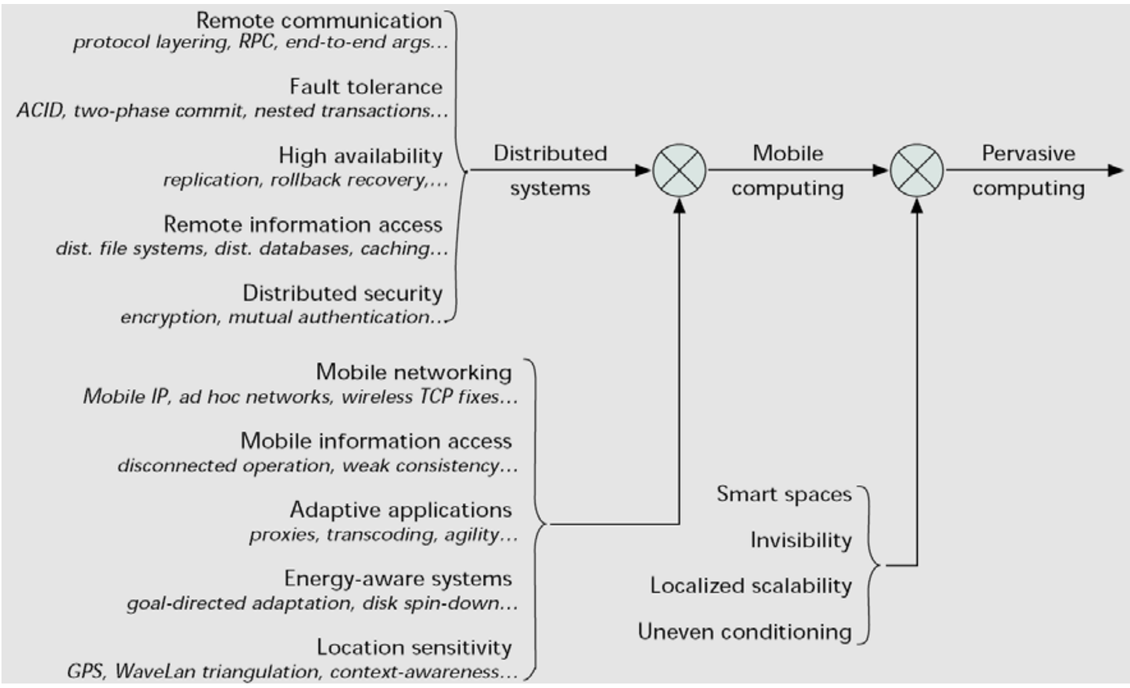

Figure 2.3 Evolution of Mobile and Pervasive Computing from Distributed Systems (Satyanarayanan, 2001).

Mobile computing is still a very active and evolving field of research, whose body of knowledge awaits codification. The results achieved so far can be grouped into the following broad topics:

- Mobile networking, including Mobile IP, ad hoc protocols, and techniques for improving.
- Mobile information access, including disconnected operation, bandwidth-adaptive file access, and selective control of data consistency.
- Support for adaptative applications, including transcoding by proxies and adaptive resource management.
- System-level energy saving techniques, such as energy-aware adaptation, variable-speed processor scheduling, energy-fair ad hoc networking, and energy-sensitive task and memory management.
- Location sensitivity, including location sensing and location-aware system behavior.

By 2000, mobile computing research began to touch upon issues that we now identify as the purview of pervasive computing. In the early 1990s, the hardware technology needed to achieve Mark Weiser's vision did not exist. Nearly 20 years later, computing and wireless communications technologies needed for its realization are becoming easily available.

As Figure 2.3 shows, pervasive computing shares many research themes in common with mobile computing. In addition, it addresses four key issues:

Smart Spaces: Embedding computing infrastructure in building infrastructure brings together two worlds that have been disjoint until now. The fusion of these worlds enables mutual sensing and control of these worlds.

Invisibility: the ideal expressed by Weiser is complete disappearance of pervasive computing technology from a user's conciousness. In practice, a reasonable approximation to this ideal is minimal user distraction. If a pervasive computing environment continuously meets user expectations and rarely presents him with surprises, it allows him to interact almost at a subconcious level.

Localized Scalability: as smart spaces grow in sophistication, the intensity of interactions between a user's personal computing space and its surroundings increases. This has severe bandwidth, energy and distraction implications for a wireless mobile user. Scalability, in the broadest sense, is thus a critical problem in pervasive computing. Like the inverse square laws of nature, good system design has to achieve scalability by severely reducing interactions between distant entities. This directly contradicts the current ethos of the Internet, which many believe heralds the “death of distance.”

Masking Uneven Conditioning: Uniform penetration of pervasive computing technology into the infrastructure is many decades away. In the interim, there will persist huge differences in the “smartness” of different environments. This large dynamic range of “smartness” can be jarring to a user, detracting from the goal of making pervasive computing technology invisible. One way to reduce the amount of variation seen by a user is to have his personal computing space compensate for “dumb” environments. As a trivial example, a system that is capable of disconnected operation is able to mask the absence of wireless coverage in its environment.

Figure 2.3 represents a focused perspective on the research ancestory of mobile and pervasive computing. Looking ahead, Figure 2.4 presents a broader perspective of the research challenges faced in this area. As this figure shows, mobile and pervasive computing share many research topics with other areas (Carnegie Mellon Web Site, 2008).

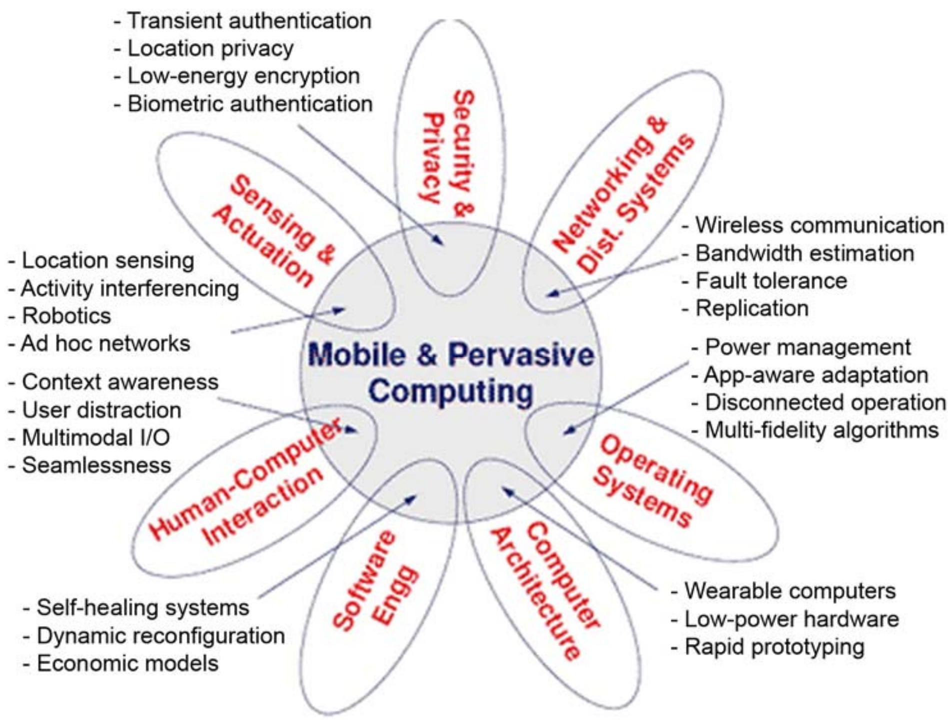

Figure 2.4 How Mobile and Pervasive Computing relate to other areas.
(Carnegie Mellon Web Site, 2008)

2.2 SERVICE ORIENTED ARCHITECTURE

Service-Oriented Computing (SOC) is a distributed computing paradigm based on the Service-Oriented Architecture (SOA), which is an architectural style for building software applications that use services (Erl 2005) (Papazoglou et al., 2007). SOC and SOA are not completely new concepts; other distributed computing technologies like CORBA and RMI have been based around similar concepts. SOA and SOC are merely extensions of the existing concepts and new technologies, like XML, and Web Services, are being used to realize platform independent distributed systems (Krafzig et al., 2005) (Erl, 2004). A SOA-based service is self-contained, i.e., the service maintains its own state. A service consists of an interface describing operations accessible by message exchange. Services are autonomous, platform-independent and can be described, published, dynamically located, invoked and (re-)combined and programmed using standard protocols. SOA promotes loose coupling between software components. The building block of SOA is the SOAP (Simple Object Access Protocol). SOAP is an XML-based messaging protocol defining standard mechanism for remote procedure calls. The Web Service Description Language (WSDL) defines the interface and details service interactions. The Universal Description Discovery and Integration (UDDI) protocol supports publication and discovery facilities. Finally, the Business Process Execution Language for Web Services (BPEL4WS) is exploited to produce a service by composing other services.

The SOA appears to be an ideal paradigm for mobile services. However, it is currently focused only on enterprise and business services. In addition, most of SOA research has been focused on architectures and implementations for wired networks. There are many challenges that need to be addressed by wireless middleware based on SOA. Wireless middleware will play an essential role in managing and provisioning service-oriented applications.

CHAPTER 3

PREVIOUS WORK

Since big advances in the technology allowed people to begin the quest for Mark Weiser's vision, Pervasive Computing projects have emerged at major universities and in industry. Project Aura at Carnegie Mellon University, Oxygen at Massachusetts Institute of Technology, Portalano at University of Washington, and Endeavour at University of California Berkeley were the first university examples. Works of AT&T Research Cambridge and IBM T.J.Watson Research Center were the first industry examples. There have been many outstanding projects in this area since then, and some of them like Place Lab, I-Find, JCAF, and Topiary were studied in detail in this thesis work, and they will be discussed in this chapter.

3.1 PLACE LAB

Place Lab project was developed at Intel Research Seattle to be a software providing low-cost, easy-to-use device positioning for location-enhanced computing applications. Place Lab tries to provide positioning which works worldwide, both indoors and out unlike GPS which only works well outside. Place Lab clients can determine their location privately without constant interaction with a central service unlike badge tracking or mobile phone location services where the service owns your location information.

The Place Lab approach is to allow devices like notebooks, PDAs and cell phones to locate themselves by listening for radio beacons such as 802.11 access points, GSM cell phone towers, and fixed Bluetooth devices that already exist in large numbers around us in the environment. These beacons all have unique or semi-unique IDs, for example, a MAC address.

Clients compute their own location by hearing one or more IDs, looking up the associated beacons' positions in a locally cached map, and estimating their own position referenced to the beacons' positions (Place Lab, 2008). We tried Place Lab as the location determination engine in our system, and it's explained in the next chapter.

3.2 IFIND

iFind was developed by researchers in SENSEable City Laboratory at Massachusetts Institute of Technology as a new social networking application that will make it possible for anyone on the Institute's huge campus to locate anyone else, via their laptop.

iFind gave all 20,000 members of the MIT community the ability to accurately calculate their location on campus, using Wi-Fi access points, and to choose if, when and with whom they want to share it with. With almost 3,000 Wi-Fi access points, the MIT campus is one of the most densely networked areas in the world. Such connectivity has changed the nature of social encounters on campus. Untethered to Ethernet cables, students, faculty and staff spend longer hours away from their offices and workstations. Cafes, lounges - sometimes just a lawn under a tree or a bench overlooking the Charles - are becoming normal workspaces.

iFind is unique compared with similar applications that are being developed for the market, in part because of the extreme precision of its positioning system. It uses Place Lab to estimate location context. More significantly, iFind has been built with particular attention to privacy and data storage issues. There is no centralized storage of data, and everything happens on encrypted peer-to-peer transmissions among users (Government Technologies, 2006).

3.3 TOPIARY

Topiary is a tool for prototyping location-enhanced applications. Location-enhanced applications make use of the location of people, places, and things to provide useful services. We see an increasing number of location-enhanced applications, particularly on mobile devices.

Topiary allows designers to quickly prototype location-enhanced applications using high-level abstractions such as maps, scenarios and storyboards, and test these application prototypes with real users in the field without having to deploy a location infrastructure (Hong and Landay, 2004).

As shown in Figure 3.1, we designed a digital version of Fatih University campus using Topiary. We explain how we used Topiary in our project in the next chapter.

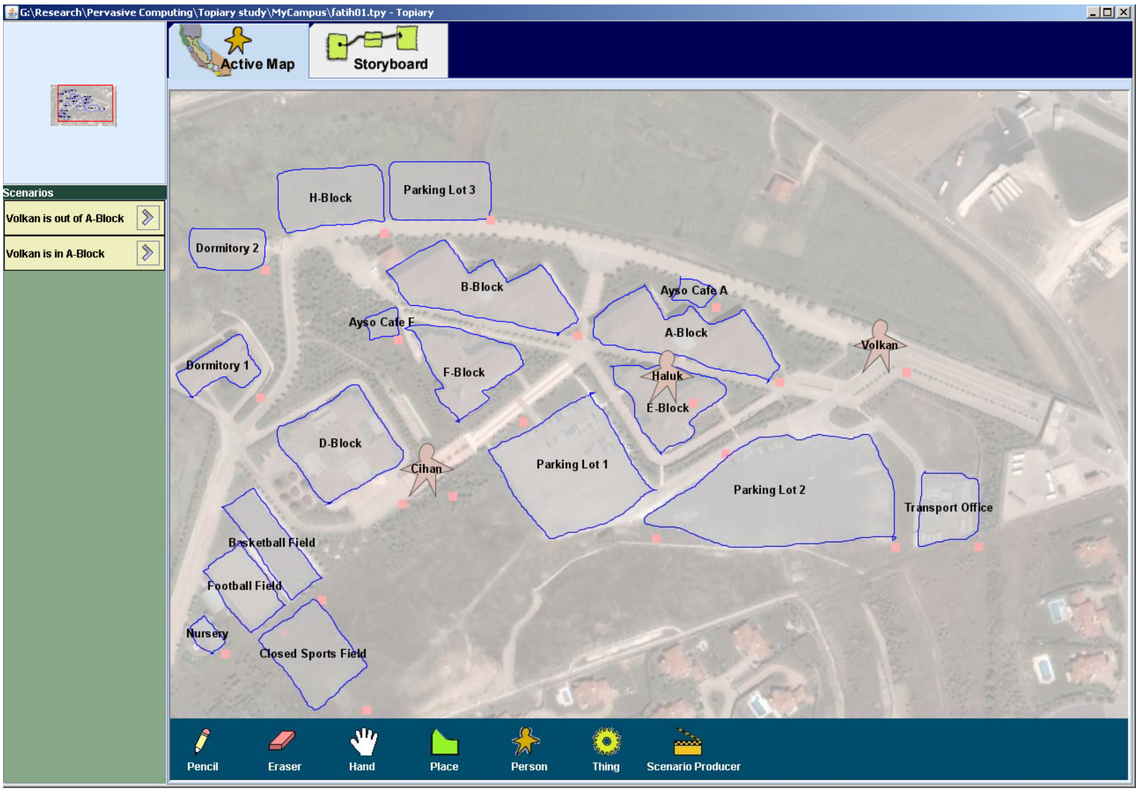

Figure 3.1 Fatih University campus designed in Topiary.

3.4 JAVA CONTEXT AWARENESS FRAMEWORK (JCAF)

Java Context-Awareness Framework (JCAF) is a Java-based set of APIs for creating context-aware application. JCAF is designed to support Context-Aware Computing and has come out of the work with the design of context-aware applications in a hospital environment (Bardram, 2004). We tried JCAF in our system, and it's explained in the next chapter.

CHAPTER 4

SERVICE ORIENTED WIRELESS CONTEXT AWARE SYSTEM

We have been developing a context-aware system, Service-Oriented Wireless Context-Aware System (SOWCAS) which leverages existing and emerging standards from both mobile devices and distributed enterprise systems (Gümüşkaya and Nural, 2007). Our primary focus in SOWCAS is to develop an indoor and outdoor context-aware system that benefits from the heterogeneity of wireless access technologies at a university campus. In SOWCAS, to collect and process context information we designed and deployed a distributed architecture composed of a central SOWCAS Server, Proxy PCs and mobile SOWCAS clients as seen in Figure 4.1.

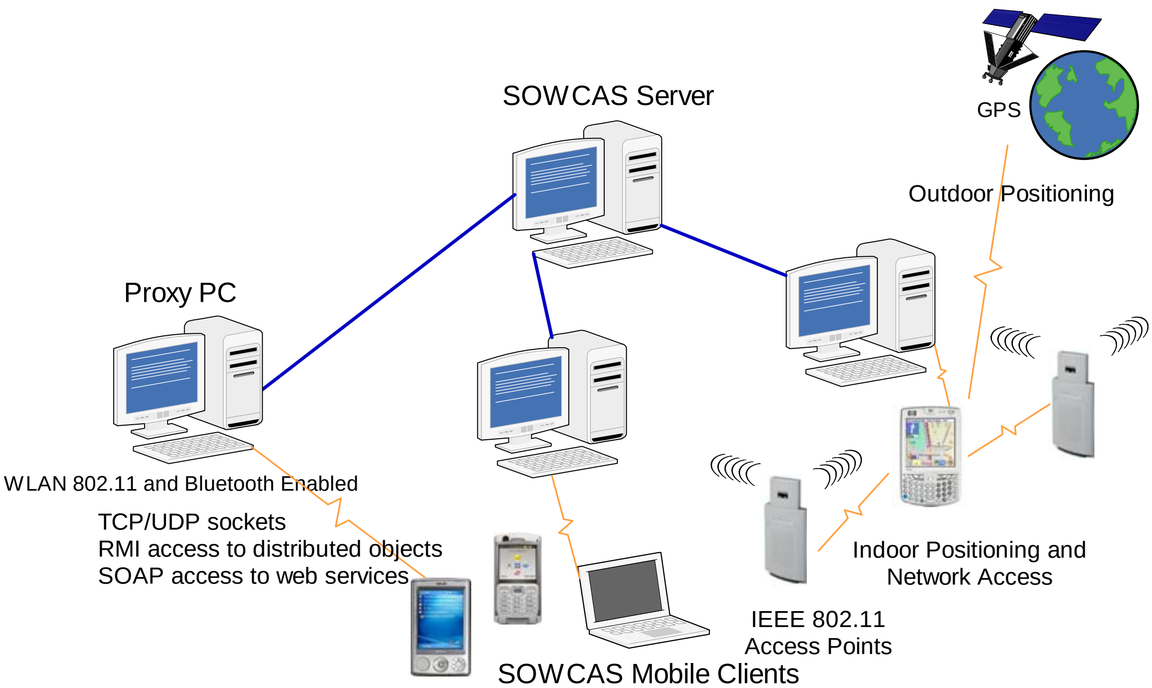

Figure 4.1. The SOWCAS system architecture.

The SOWCAS Proxy software runs on Proxy PCs located in classes, laboratories and some special locations in which students and professors are found frequently. These PCs are equipped with WLAN 802.11 and Bluetooth network cards. These PCs and wireless access points located in buildings are used to locate and track the positions of mobile users in the university.

SOWCAS uses novel approaches and some of the improved design ideas developed in our old projects. We proposed a meta software architecture and some design principles (Gümüşkaya, 2006). We applied these in our WLAN indoor positioning system WiPoD (Wireless Position Detector) (Gümüşkaya, 2005). We investigated several implementations of WLAN positioning on WiPoD using different algorithms and looked for the answer of if the implementation could be achieved at a level of acceptable accuracy to be used in real-life cases. After WiPoD, we developed a generic framework, Service-Oriented Reflective Wireless Middleware (SORWiM) (Gümüşkaya, 2006), for indoor and outdoor mobile applications. It provides a set of basic and composite Web Services for efficient and reliable information discovery and dissemination for mobile computing.

Many context-aware systems are built over positioning systems extending them to provide task specific context-aware information. SOWCAS is currently using WiPoD as its positioning engine. WiPoD uses well known K-Nearest Neighbor and Triangulation algorithms to estimate the position of a mobile user for indoor and outdoor environments respectively. WiPoD locates and tracks the user having an IEEE 802.11, Bluetooth and GPS supported device at the Fatih University campus. It operates by processing received signal information from multiple WLAN APs and GPS data. Besides indoor and outdoor location context information, we also keep other context information such as status, activity, weekly schedules, detailed personal information for students and faculty members in the central database at the SOWCAS Server.

4.1 ARCHITECTURES FOR MOBILE DEVICES

We use middleware approach in both the client and server parts of SOWCAS (Gümüşkaya, 2006). The SOWCAS middleware is decomposed into multiple layers such as host infrastructure middleware (Java Virtual Machine), distribution middleware (such as RMI and SOAP), common middleware services, and domain-specific middleware services.

The new state of the art Pocket PCs and smart phones have heterogeneous wireless technologies such as GSM/UMTS, GPRS/EGDE, WLAN, Bluetooth, and GPS. The programming access to these wireless technologies is generally provided by the third party products and software development environments as C/C++ APIs (Application Programming Interfaces) which are system dependent. This communication layer is shown as the Operating System (OS) Adaptation Layer in the SOWCAS client architecture in Figure 4.2 (a).

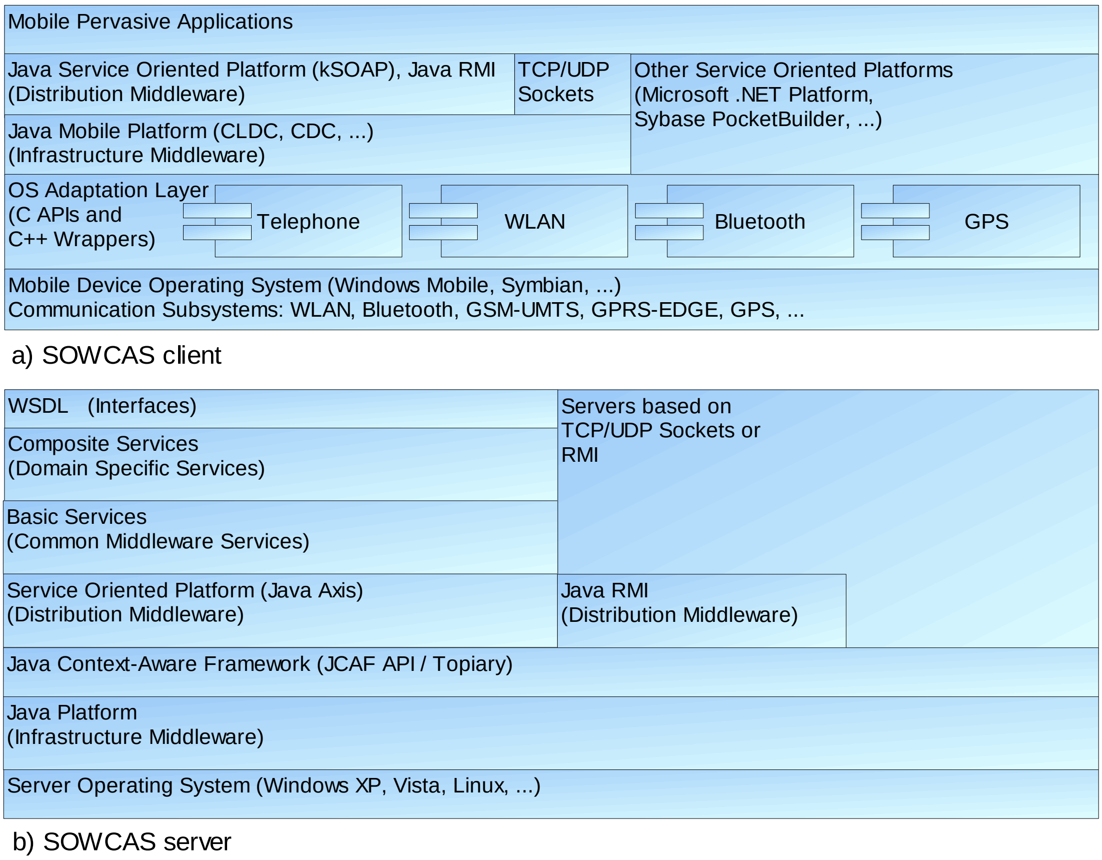

Figure 4.2. The SOWCAS client and server architectures.

We use Java for portability and development reasons in SOWCAS as much as possible. When it is not feasible or difficult to use Java for some mobile devices, we use other software development platforms such as Microsoft's .NET and Sybase's PocketBuilder to develop client applications.

We used JCAF which is a Java-RMI based context-awareness framework in the first SOWCAS prototype (Bardram, 2004). The RMI feature makes the JCAF client Java language dependent. Since we have one more layer, Web Services layer on top of JCAF as shown in Figure 4.2 (b), our system is platform independent. Since JCAF has several restrictions, we have started to develop our own context-awareness framework using some of the ideas and software components developed in the open source Topiary project (Li et al., 2004).

4.2 WIRELESS NETWORK CONNECTIONS FOR SMALL MOBILE DEVICES

4.2.1 Wireless Short-Range Access Technologies

Bluetooth is a low cost short range wireless communication technology which was designed to replace the cables connecting portable or fixed devices (Bluetooth Basics Web Site 2008). By using the Bluetooth technology which works on 2.4 GHz ISM frequency band, it is theoretically possible to create Wireless Personal Area Networks (WPANs) with a maximum range of 100 meters, and 1 Mbps (BT v1.2) – 3 Mbps (BT v2.0 + EDR) bandwidth (Wikipedia Bluetooth Web Site, 2008).

Wi-Fi (Wireless Fidelity) (IEEE 802.11) is a short range wireless communication technology which was designed for particularly mobile devices to make access to information easier, ensure compatibility and co-existence, and eliminate cabling and wiring (Wikipedia Wi-Fi Web Site, 2008). Wi-Fi provides better range and bandwidth than Bluetooth, but its power usage is quite higher. Wi-Fi specifications are defined by IEEE 802.11 standards. Current devices mostly support at least one of IEEE 802.11 a/b/g/n standards. Depending on the supported standard, Wi-Fi works on 2.4 or 5 GHz ISM frequency band. By using Wi-Fi technology, it is theoretically possible to create Wireless Local Area Networks (WLANs) with a maximum range of 250 meters, and 11 Mbps (802.11b) - 54 Mbps (802.11a/g) - 300 Mbps (802.11n).

Current small communication devices mostly have Bluetooth technology, but very few of them can match Bluetooth v2.0 and Enhanced Data Rate (EDR) specifications which provide better range and bandwidth. Also these devices mostly have Wi-Fi support, but very few of them can match IEEE 802.11g and/or IEEE 802.11n specifications which provide much better range and bandwidth.

4.2.2 Software Technologies for Wireless Short-Range Connections

Reflective middleware presents a comprehensive solution to deal with ubiquitous computing (Roman et al. 2001). Reflective middleware system responses are optimized to changing environments, including mobile interconnections, power levels, network bandwidth, latency/jitter, and dependability needs. Reflection is the ability of a program to observe and possibly modify its structure and behavior. A reflective middleware server should be able provide several network connection alternatives (such as TCP/UDP (User Datagram Protocol) sockets, RMI, and SOAP) to different network applications demanding different bandwidths, data reliability and time sensitivity requirements. In addition, based on the clients' conditions, such as hardware resources (like CPU, memory, and I/O capabilities), battery powers, and the network quality, the server should give different services to its clients.

When developing the SOWCAS architecture, client/server architectures based on basic TCP/UDP socket connections, RMI distributed object technology and service oriented approaches were evaluated. We developed client and server test programs in these four technologies and performed detailed analyses on wireless network connections between a mobile device and a fixed server.

UDP is a connectionless unreliable transport layer protocol generally used for streaming multimedia applications which are loss tolerant and rate sensitive. We developed a small Java multimedia server based on UDP which distributes movie clips to its clients using the application layer protocols RTP (Real Time Protocol) and RTSP (Real Time Streaming Protocol). We did not use any library; the programs handle packetizing using RTP and client/server interactions using RTSP. The performance of the Java RTSP client application on 416 MHz PDA was acceptable. These results are not presented in the next section. Since the transport layers of RMI and SOAP use the TCP protocol, we use the TCP protocol as a reference base when comparing the RMI and SOAP implementations. The UDP test results for mobile devices give us a very useful insight for real time multimedia applications.

The SOA appears to be an ideal paradigm for mobile services (Yurday and Gümüşkaya, 2006). However, it is currently focused only on enterprise and business services. In addition, most of SOA research has been focused on architectures and implementations for wired networks. There are many challenges that need to be addressed by wireless middleware based on SOA. Wireless middleware will play an essential role in managing and provisioning service-oriented applications. One of the primary goals of this research is to investigate how the construction of mobile services can benefit from the SOA paradigm.

CHAPTER 5

PERFORMANCE ANALYSES

There are a number of research studies that evaluate the performance of Web Services and SOAP compared to middleware distributed object technologies, including RMI and CORBA, for wired and wireless networks. In these studies, a comparison is made based on the performance of a web service implementation that calls a method and an RMI/CORBA implementation that calls the same method. Several studies showed that the available web service solutions were significantly worse than RMI/CORBA implementations (Jagannadham et al., 2007). But some earlier studies also showed that Web Services could outperform distributed objects for proper implementations (Cook and Barfield, 2006).

(Elfwing et al., 2002) was one of the first performance evaluation studies that examine Web Services compared to distributed objects. They have performed tests of CORBA and Web Services with measurements of single method invocations, on desktop computers.

(Gray, 2004) have compared RMI, CORBA and Web Services. They provided detailed measurements including memory usage, CPU times, latency times and packet counts. They used fixed PCs and 100 Mbit wired network.

(Juric et al. 2004) and (Juric et al. 2007) have compared the local and network performance of RMI and Web Services, considering network overhead, firewalls and proxy secured networks, using fixed PCs and 100 Mbit wired network.

(Demarey et al., 2005) have compared 20 different middleware implementations, including RMI and Web Services, using measurement of an empty “ping” method invocation on the server, which was a fixed PC.

(Kim and Han, 2006) have provided performance comparison of DCOM, CORBA and Web Services. They measured the round trip performance using the simple data types on local and remote machines of fixed PCs on a wired network.

(Cook and Barfield, 2006) have compared the performance of Web Services and RMI, using desktop and laptop PCs, and wireless networks.

(Juric et al., 2006) have studied RMI and Web Services, and their secure variants, WS-Security and RMI-SSL. They identified functional differences, and compared the performance on two operating systems, Windows and Linux, using fixed PCs and 100 Mbit wired network.

(Jagannadham et al., 2007) have studied on performance comparison and functional analysis of Java sockets, RMI, Servlet, CORBA, XML-RPC, and Web Services, using fixed PCs and wired network.

Our performance studies have many important differences than previous work. The first difference is that our performance tests were performed for small mobile devices. Limited resources, heterogeneity, and a high degree of dynamism are the most common properties that usually exist in mobile devices such as Pocket PCs and mobile phones. Mobile devices don't have powerful CPUs, large amount of memory and high-speed networking capabilities compared to desktop PCs. Conventional middleware platforms such as RMI and CORBA, and some new technologies like Web Services may not be appropriate for many mobile devices and wireless applications.

The second important difference of our performance study is that the tests were conducted for different wireless network connections. The previous performance studies were done on wired networks for powerful fixed desktop computers.

The third difference of our performance study is the use of Java on mobile devices for all network software communication technologies including Web Services and RMI. Although in recent years, the better new generation implementations for Web Services have emerged for desktop computers, there are few SOA Java packages for small mobile devices. There are also not many RMI implementations for mobile devices and they are limited and not as functional as they are for desktop computers.

5.1 TEST EQUIPMENTS

The specifications of the devices which were used for performing our benchmarking tests are given below:

- HP Compaq nx8220 Laptop
  - 1.86 GHz Intel Centrino CPU
  - 1 GB Ram
  - Intel 2200 802.11 BG Wireless Adapter
  - NetXtreme Gigabit Ethernet
  - Bluetooth v1.2
  - Windows XP Professional SP2

- Toshiba A100-192 Laptop
  - 2.0 GHz Intel Core Duo CPU
  - 2 GB Ram
  - Intel 3945 802.11 ABG Wireless Adapter
  - Intel Pro 100 Ethernet
  - Bluetooth v2.0 + EDR
  - Windows XP Home SP2

- Asus MyPal A636n PDA
  - 416 MHz Intel Xscale CPU
  - 128 MB Flash ROM, 64 MB SDRAM
  - 2 GB FLASH SD Storage Card
  - WLAN 802.11 BG
  - Bluetooth v2.0 + EDR
  - Windows Mobile 6.0

- Surecom EP-808SX 8 Port 10/100M Ethernet Mini Switch

5.2 TESTING PROCEDURE

We performed benchmarking tests to measure the performance of a mobile client wireless connection to see if its performance is acceptable for a wireless application based on new SOA and RMI implementations running on small mobile devices. The performance tests of small mobile devices utilized a fixed server and a mobile hand-held client device.

The network traffic analysis was conducted using the Wireshark network packet analysis tool to capture packets transferred over the 54 Mbit WLAN link (IEEE 802.11g) and 700 Kbit (Bluetooth v1.2) / 3 Mbit (Bluetooth v2.0+EDR) WPAN (Wireless Personal Area Network) wireless links and 100 Mbit wired Ethernet link between the server and the client device.

The server side software technologies were Jakarta Tomcat 5.5, Axis 1.3, and Java 1.5 SDK running on Windows XP Home/Professional with Service Pack 2.

For the wireless network tests, the client side software technologies were NSICom CrEme JVM 4.11, NSICom RMI package and kSOAP library running on Windows Mobile 6.0. For the wired network tests (100Mbit Ethernet), we used laptop PC on the client side, so that the software configurations were the same as the server side. We developed a Java client test program having the same GUI shown in Figure 5.1 for each wireless TCP, RMI and SOAP connections.

The test measures include CPU-times for both client and server, overall latency time for a single request and for sequences of requests, total byte transfers, and packet counts. The first tests were performed on small (10 bytes), medium (100 and 1000 bytes) and large (10000 and 50000 bytes) data sets many times in TCP, RMI and SOAP accesses, and the average client CPU times were calculated for each test.

Before each tests, all unnecessary programs and services were disabled on client and server devices, and the operating systems of devices were restarted. On PDA, power save option was disabled and CPU power was increased to maximum level.

The simplified remote interface for performance tests is given below:

```
public interface SOWCASTest extends Remote {
  public String get10Bytes() throws RemoteException();
  public String get100Bytes() throws RemoteException();
  public String get1KBytes() throws RemoteException();
  public String get10KBytes() throws RemoteException();
  public String get50KBytes() throws RemoteException();
}
```

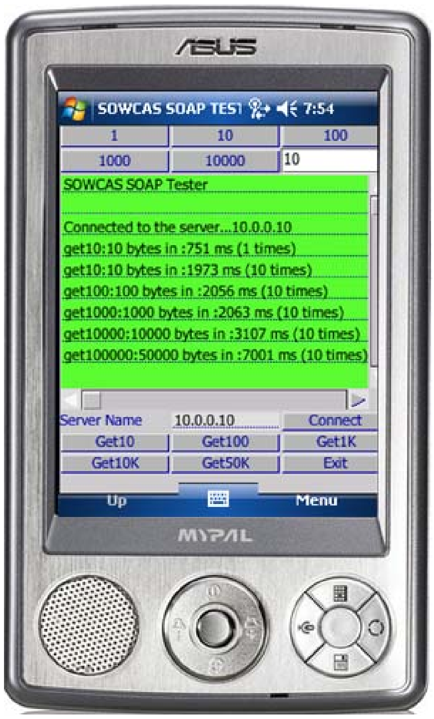

Figure 5.1 The performance testing program GUI on ASUS Pocket PC.

5.3 TEST RESULTS

We performed five different tests on wireless and wired network configurations. In this section, we give test setup and configuration settings for both client and server devices, and then measured results for each connection. At the last section, we give a comparison of all connections.

5.3.1 Bluetooth v1.2 Tests

5.3.1.1 Client

PDA (Asus MyPal A636N, Bluetooth ver2.0+EDR support)
IP Address: 169.254.21.118
Subnet Mask: 255.255.0.0

5.3.1.2 Server

Laptop PC (HP Compaq nx8220, Bluetooth ver1.2 support, Windows BT Stack)
IP Address: 169.254.208.240
Subnet Mask: 255.255.0.0

5.3.1.3 Results

On PDA, power save option was disabled and CPU power was increased to maximum level. Network access services between laptop and PDA were activated, and a 700Kbps Bluetooth WPAN was established.

During first tests, distance between two devices was about 1 meter, and signal strength was at “Good” level. When the distance reached to 3 meters, signal strength stayed the same, but there was a 25% performance loss. At the distances of 6 meters and more, some exception errors were seen depending on losing wireless ad-hoc network connection. Data transfer duration was increased up to 2.5 times during data transfers at more than 6 meters.

Measured average CPU times and calculated performance rates are given in Table 5.1, and illustrated in Figure 5.2 and Figure 5.3. We give only results of 1 meter distance, as they are the best average results for Bluetooth v1.2.

Table 5.1 The average CPU times for TCP, Web Service and RMI implementations, and performance rates on Bluetooth v1.2 WPAN connection.

	Client Average CPU Times (ms)				
# bytes	10	100	1K	10K	50K
TCP	29,2	42,0	46,4	245,0	802,1
RMI	32,3	37,6	52,8	196,8	888,2
WS	307,2	312,1	318,8	544,5	1433,7
	Performance Rates				
WS/TCP	9,5	8,3	6,0	2,8	1,6
RMI/TCP	10,5	7,4	6,9	2,2	1,8
WS/RMI	1,1	0,9	1,1	0,8	1,1

K = 1000 bytes

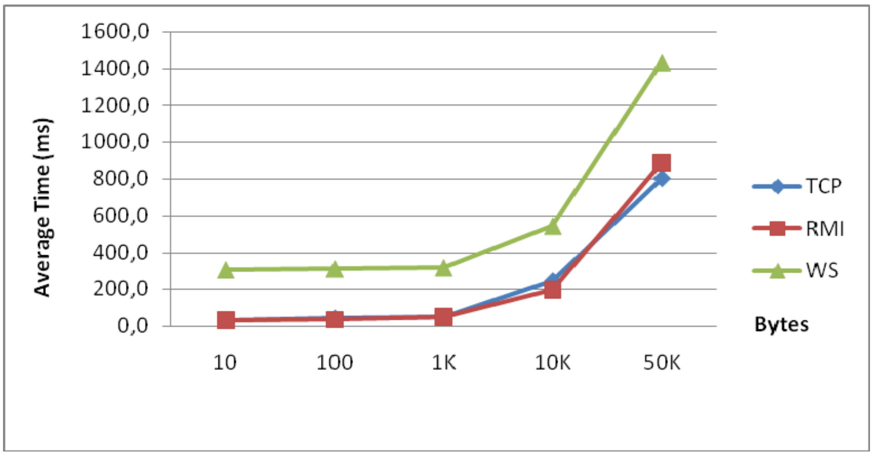

Figure 5.2 The CPU times of TCP, RMI and Web Service implementations on Bluetooth v1.2 WPAN connection.

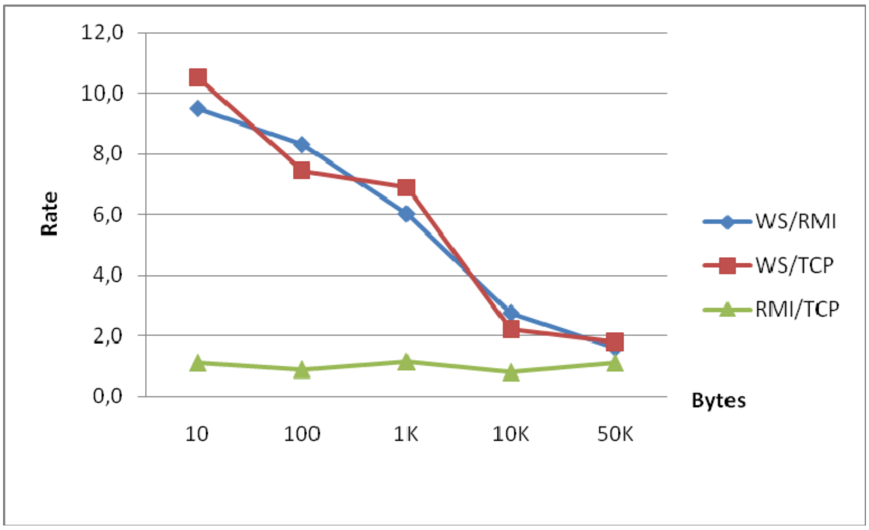

Figure 5.3 The performance rate of TCP, RMI and Web Service implementations on Bluetooth v1.2 WPAN connection.

5.3.2 Bluetooth v2.0 Tests

5.3.2.1 Client

PDA (Asus MyPal A636N, Bluetooth ver2.0+EDR support)
IP Address: 169.254.21.118
Subnet Mask: 255.255.0.0

5.3.2.2 Server

Laptop PC (Toshiba A100-192, Bluetooth ver2.0+EDR support, Toshiba BT Stack)
IP Address: 169.254.217.233
Subnet Mask: 255.255.0.0

5.3.2.3 Results

On PDA, power save option was disabled and CPU power was increased to maximum level. Network access services between laptop and PDA were activated, and a 3Mbps Bluetooth WPAN was established.

During the first tests, the distance between two devices was about 1 meter, and signal strength was at "Good" level. When the distance reached to 3 meters, signal strength stayed the same. Thanks to better range and higher bandwidth provided by Bluetooth v2.0 and Enhanced Data Rate (EDR) technology, obtained results for 3 meters were very close to the ones for 1 meter. At 6 meters, although signal quality was decreased seriously and the connection became unstable, results were close to the shorter range tests.

The results of the first tests which were taken at 1 meter distance are given in Table 5.2, and illustrated in Figure 5.4 and Figure 5.5.

Table 5.2 The average CPU times for TCP, Web Service and RMI implementations, and performance rates on Bluetooth v2.0 WPAN connection.

	Client Average CPU Times (ms)				
# bytes	**10**	**100**	**1K**	**10K**	**50K**
TCP	23,0	24,8	32,9	86,6	313,1
RMI	24,7	25,9	33,8	91,2	344,0
WS	276,5	288,6	298,1	448,9	943,9
	Performance Rates				
WS/TCP	11,2	11,1	8,8	4,9	2,7
RMI/TCP	12,0	11,6	9,1	5,2	3,0
WS/RMI	1,1	1,0	1,0	1,1	1,1

K = 1000 bytes

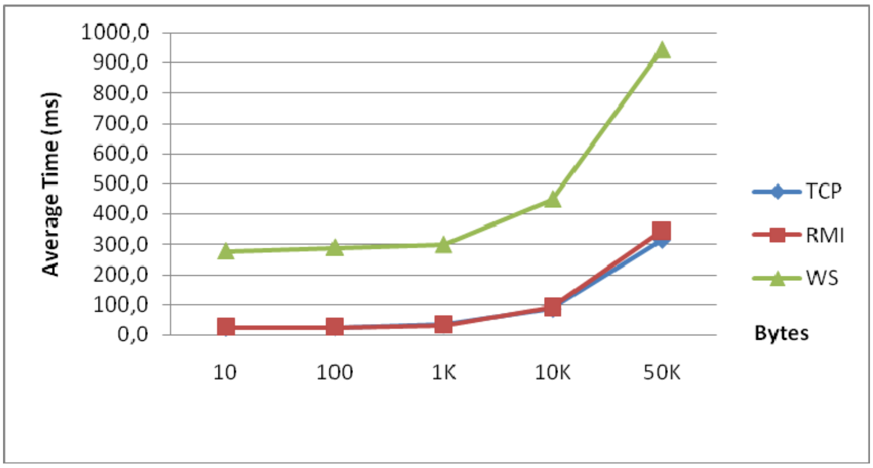

Figure 5.4 The performance of TCP, RMI and Web Service implementations on Bluetooth v2.0 WPAN connection.

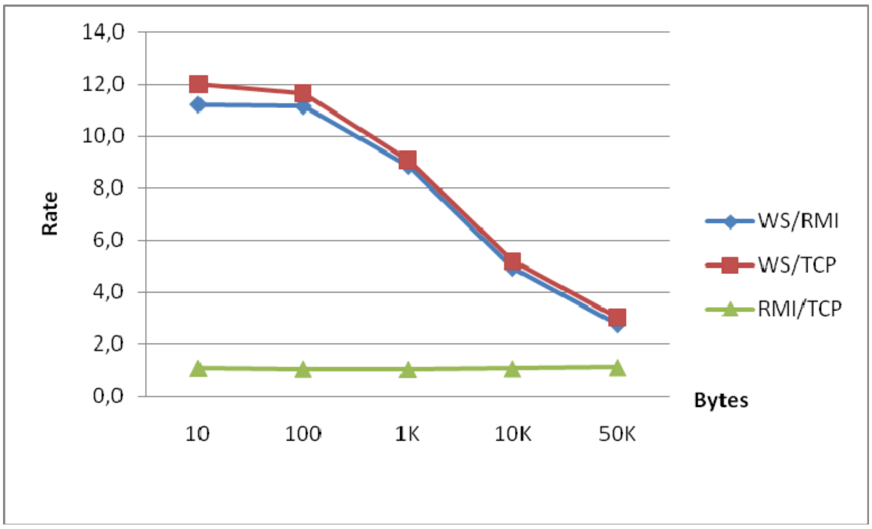

Figure 5.5 The performance rate of TCP, RMI and Web Service implementations on Bluetooth v2.0 WPAN connection.

5.3.3 Wi-Fi Tests

5.3.3.1 Client

PDA (Asus MyPal A636N, IEEE 802.11 g support)
IP Address: 169.254.21.118
Subnet Mask: 255.255.0.0

5.3.3.2 Server

Laptop PC (HP Compaq nx8220, IEEE 802.11 bg support)
IP Address: 169.254.208.240
Subnet Mask: 255.255.0.0

5.3.3.3 Results

On PDA, power save option was disabled and CPU power was increased to maximum level. Network access services between laptop and PDA were activated, and a 54Mbps ad-hoc WLAN (Wireless Local Area Network) was established.

In this test, devices connected each other via IEEE 802.11g. Distance between two devices was about 3 meters, and signal strength was at the highest level during the tests. Wi-Fi connection gives relatively high performance, if there are no walls, objects, people or devices between or around the client and the server devices. There is no significant performance loss on 6 meters of distance.

Measured CPU times and calculated performance rates are given in Table 5.3, and illustrated in Figure 5.6 and Figure 5.7.

Table 5.3 The average CPU times for TCP, Web Service and RMI implementations, and performance rates on Wi-Fi WLAN connection.

	Client Average CPU Times (ms)				
# bytes	**10**	**100**	**1K**	**10K**	**50K**
TCP	3,4	3,5	5,6	20,7	90,0
RMI	5,1	5,4	7,6	35,0	164,7
WS	197,0	199,4	206,4	308,3	685,6
	Performance Rates				
WS/TCP	38,6	36,9	27,2	8,8	4,2
RMI/TCP	57,9	57,0	36,9	14,9	7,6
WS/RMI	1,5	1,5	1,4	1,7	1,8

K = 1000 bytes

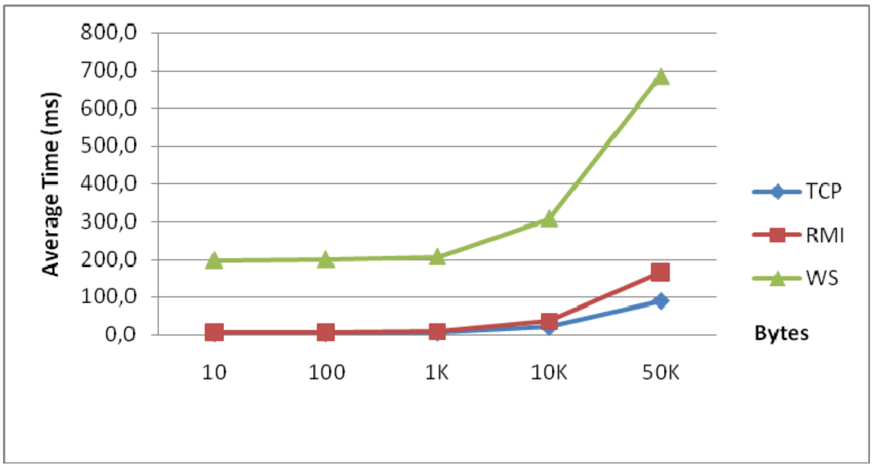

Figure 5.6 The performance of TCP, RMI and Web Service implementations on Wi-Fi WLAN connection.

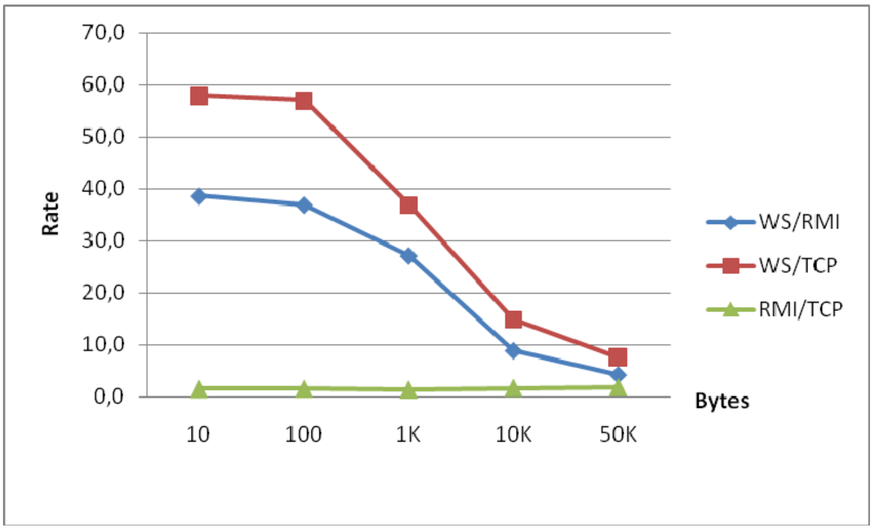

Figure 5.7 The performance rate of TCP, RMI and Web Service implementations on Wi-Fi WLAN connection.

5.3.4 Wired Network (10Mbps) Tests

5.3.4.1 Client

PDA (Asus MyPal A636N, USB 1.1)
IP Address: 169.254.21.118
Subnet Mask: 255.255.0.0

5.3.4.2 Server

Laptop PC (HP Compaq nx8220, USB 2.0)
IP Address: 169.254.208.240
Subnet Mask: 255.255.0.0

5.3.4.3 Results

On PDA, power save option was disabled and CPU power was increased to maximum level. Network access services between laptop and PDA were activated, and PDA was connected to the laptop PC via the USB synchronization cable. A 10Mbps LAN was established by the wired connection. Measured CPU times and calculated performance rates are given in Table 5.4, and illustrated in Figure 5.8 and Figure 5.9.

We have two reasons to perform this test: First, we have a more stable network, since we established a wired network connection in which we don't have to think about things like performance drops due to the interference of radio signals in wireless connections, and etc. Second, we compare the performances of the small communication devices, in this case a Pocket PC, with the regular laptop PCs. As we mentioned before, there are several important reasons for a Pocket PC to perform quite slower than a laptop.

The next section gives the results of the test which we performed on a pair of laptop PCs and a 100Mbps wired Ethernet connection.

Table 5.4 The average CPU times for TCP, Web Service and RMI implementations, and performance rates on a 10 Mbps LAN connection.

	Client Average CPU Times (ms)				
# bytes	**10**	**100**	**1K**	**10K**	**50K**
TCP	2,4	3,1	5,1	26,0	131,6
RMI	5,1	5,3	8,4	43,4	194,1
WS	199,1	198,7	201,1	303,7	718,9
	Performance Rates				
WS/TCP	39,3	37,4	23,8	7,0	3,7
RMI/TCP	83,0	64,7	39,7	11,7	5,5
WS/RMI	2,1	1,7	1,7	1,7	1,5

K = 1000 bytes

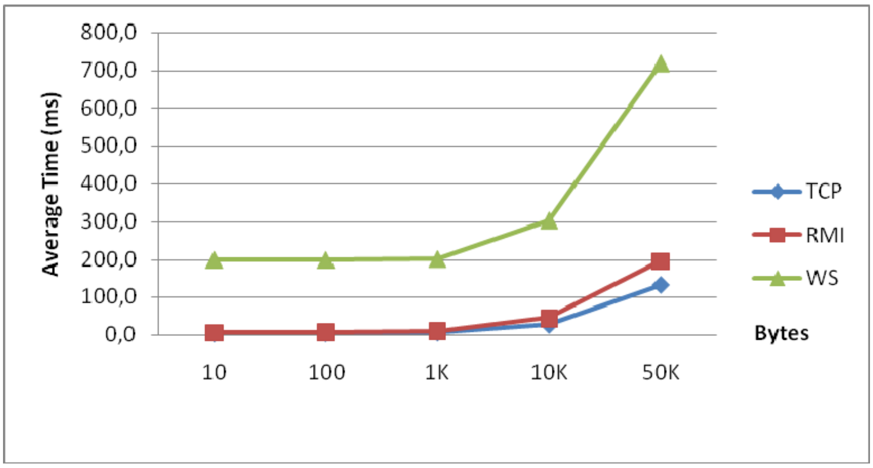

Figure 5.8 The performance of TCP, RMI and Web Service implementations on a 10 Mbps LAN connection.

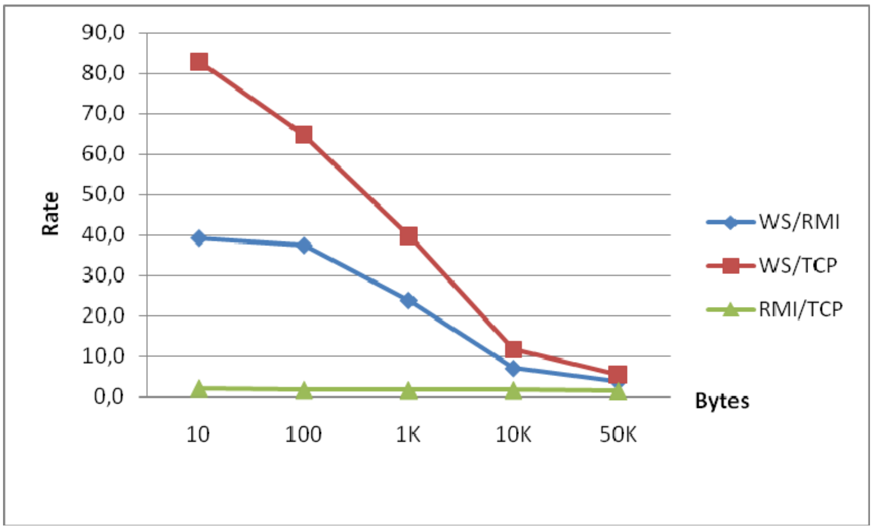

Figure 5.9 The performance rate of TCP, RMI and Web Service implementations on a 10 Mbps LAN connection.

5.3.5 Wired Network (100Mbps) Tests

5.3.5.1 Client

Laptop PC (HP Compaq nx8220, NetXtreme Gigabit Ethernet with 1Gbps support)
IP Address: 169.254.208.240
Subnet Mask: 255.255.0.0

5.3.5.2 Server

Laptop PC (Toshiba A100-192, Intel Pro 100 Ethernet with 100Mbps support)
IP Address: 169.254.217.233
Subnet Mask: 255.255.0.0

5.3.5.3 Results

This test setup was selected to be a reference point for our performance tests which are based on wireless networks of a laptop PC server and a PDA client. We used laptop PC as the client, which has stronger CPU, larger memory capability, and better networking features than our PDA client. Also Java Virtual Machine and SOAP implementation in the PDA have quite limited functions as mentioned in the earlier chapters.

In order to setup a wired LAN, we used a SureCom 8 port 100Mbps Ethernet Hub/Switch and a pair of straight-through CAT5 cable. Measured CPU times and calculated performance rates are given in Table 5.5, and illustrated in Figure 5.10 and Figure 5.11.

Table 5.5 The average CPU times for TCP, Web Service and RMI implementations, and performance rates on a 100 Mbps LAN connection.

	Client Average CPU Times (ms)				
# bytes	**10**	**100**	**1K**	**10K**	**50K**
TCP	0,2	0,3	0,4	0,9	4,8
RMI	0,5	0,6	0,7	1,5	6,2
WS	4,8	5,3	6,0	9,4	32,9
	Performance Rates				
WS/TCP	9,0	9,0	9,0	6,1	5,3
RMI/TCP	28,5	21,3	14,9	10,6	6,8
WS/RMI	3,2	2,4	1,7	1,8	1,3

K = 1000 bytes

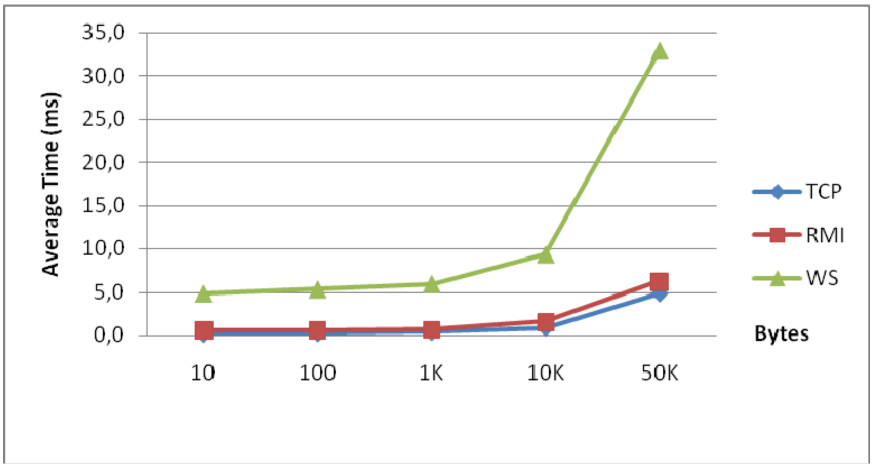

Figure 5.10 The performance of TCP, RMI and Web Service implementations on a 100 Mbps LAN connection.

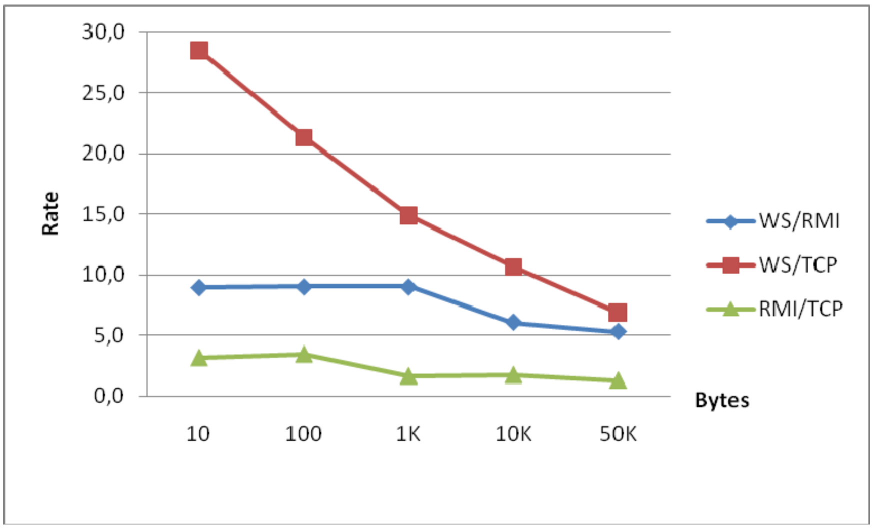

Figure 5.11 The performance rate of TCP, RMI and Web Service implementations on a 100 Mbps LAN connection.

5.3.6 Comparison of All Tests

In the previous sections, we presented each test results individually. We found it useful to present a comparison of the results as shown in the Table 5.6, Figure 5.12 and Figure 5.13 below.

Table 5.6 The average CPU times for TCP, Web Service and RMI implementations, and performance rates on all connections.

	Client Average CPU Times (ms)				
# bytes	**10**	**100**	**1K**	**10K**	**50K**
TCP (BT v1.2)	29,2	42,0	46,4	245,0	802,1
TCP (BT v2.0)	23,0	24,8	32,9	86,6	313,1
TCP (Wi-Fi)	3,4	3,5	5,6	20,7	90,0
TCP (10Mbps)	2,4	3,1	5,1	26,0	131,6
TCP (100Mbps)	0,2	0,3	0,4	0,9	4,8
RMI (BT v1.2)	32,3	37,6	52,8	196,8	888,2
RMI (BT v2.0)	24,7	25,9	33,8	91,2	344,0
RMI (Wi-Fi)	5,1	5,4	7,6	35,0	164,7
RMI (10Mbps)	5,1	5,3	8,4	43,4	194,1
RMI (100Mbps)	0,5	0,6	0,7	1,5	6,2
WS (BT v1.2)	307,2	312,1	318,8	544,5	1433,7
WS (BT v2.0)	276,5	288,6	298,1	448,9	943,9
WS (Wi-Fi)	197,0	199,4	206,4	308,3	685,6
WS (10Mbps)	199,1	198,7	201,1	303,7	718,9
WS (100Mbps)	4,8	5,3	6,0	9,4	32,9
	Performance Rates				
RMI/TCP (BT v1.2)	1,1	0,9	1,1	0,8	1,1
RMI/TCP (BT v2.0)	1,1	1,0	1,0	1,1	1,1
RMI/TCP (Wi-Fi)	1,5	1,5	1,4	1,7	1,8
RMI/TCP (10Mbps)	2,1	1,7	1,7	1,7	1,5
RMI/TCP (100Mbps)	3,2	2,4	1,7	1,8	1,3
WS/RMI (BT v1.2)	9,5	8,3	6,0	2,8	1,6
WS/RMI (BT v2.0)	11,2	11,1	8,8	4,9	2,7
WS/RMI (Wi-Fi)	38,6	36,9	27,2	8,8	4,2
WS/RMI (10Mbps)	39,3	37,4	23,8	7,0	3,7
WS/RMI (100Mbps)	9,0	9,0	9,0	6,1	5,3
WS/TCP (BT v1.2)	10,5	7,4	6,9	2,2	1,8
WS/TCP (BT v2.0)	12,0	11,6	9,1	5,2	3,0
WS/TCP (Wi-Fi)	57,9	57,0	36,9	14,9	7,6
WS/TCP (10Mbps)	83,0	64,7	39,7	11,7	5,5
WS/TCP (100Mbps)	28,5	21,3	14,9	10,6	6,8

K = 1000 bytes

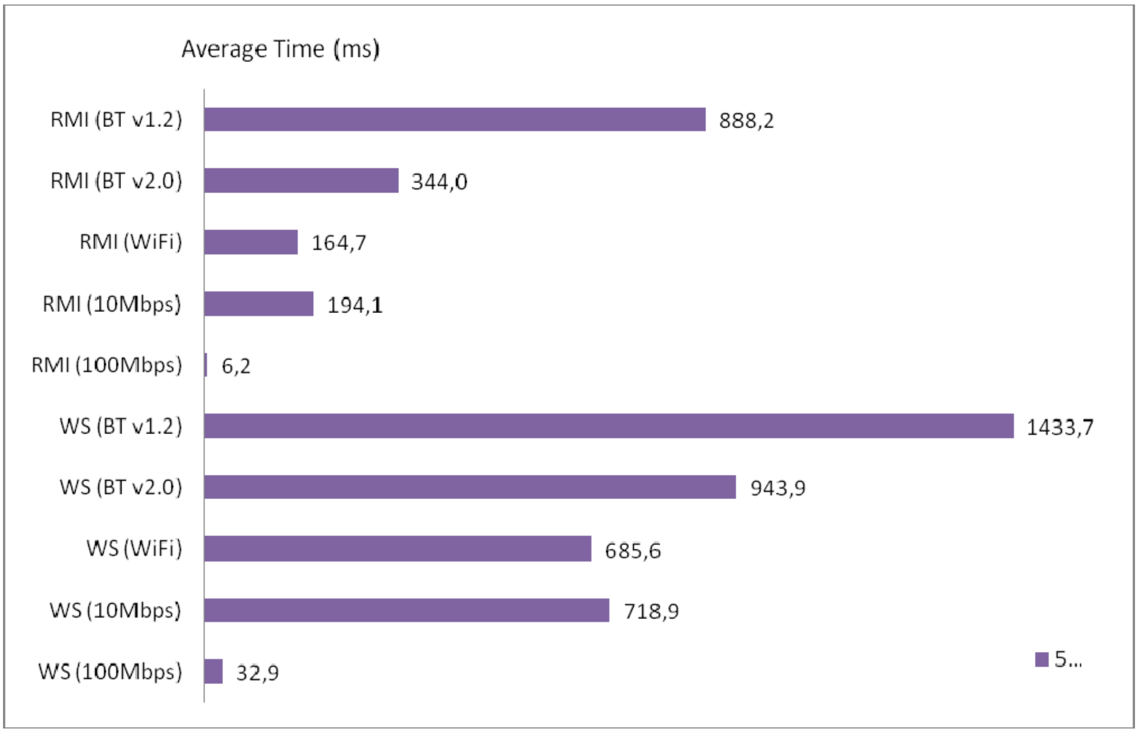

Figure 5.12 The performance of TCP, RMI and Web Service implementations on all connections.

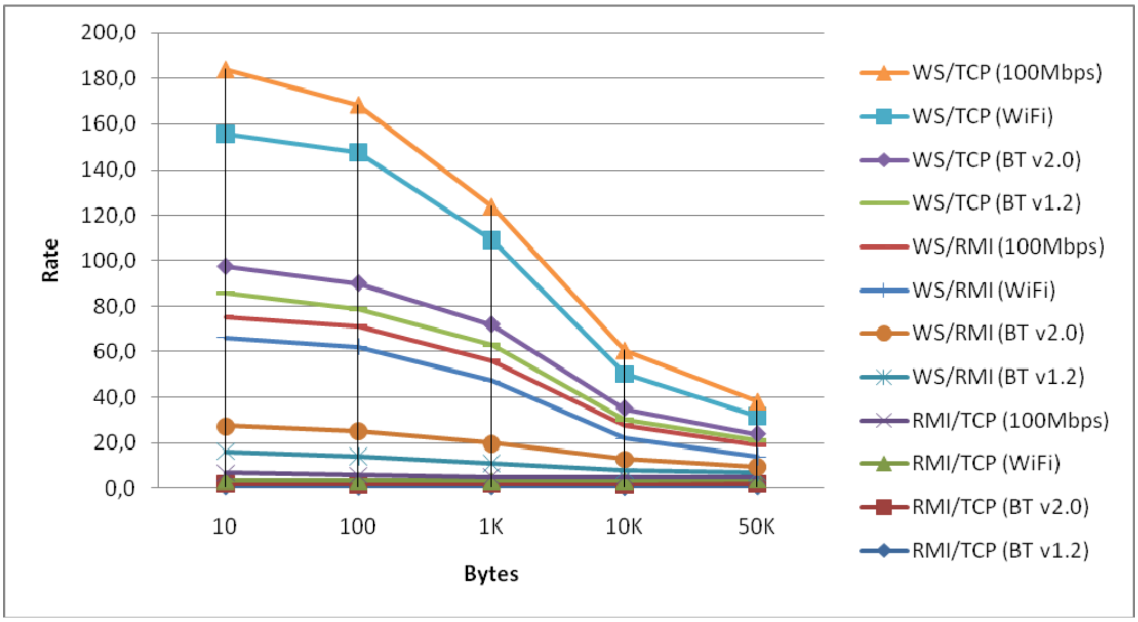

Figure 5.13 The performance rates of TCP, RMI and Web Service implementations on all connections.

CHAPTER 6

CONCLUSIONS

As seen from our test results, the Web Services implementations perform slower than RMI and TCP socket implementations. The RMI implementation is faster for small batches of documents and low-latency networks, but performance degrades rapidly with larger batches and higher latency. The Web Services have a high initial cost, but show little or no change with larger batches. Higher latency creates a greater initial cost, but performance is still independent of batch size. As network latency increases, in our tests as decreasing from a 100 Mbit Ethernet connection, to 54 Mbit Wi-Fi, and 3 Mbit and 700 Kbit Bluetooth connections, the performance benefits of the document-oriented approach in Web Services increase significantly. This is relevant when in some real world wireless communication scenarios, latency may even be minutes for disconnected or asynchronous operations. The performance studies generally measure RPC-style communication and do not consider the possibilities of document-oriented designs that show the strengths of Web Services. Web Services are not intended to be used RPC-style like other distributed object technologies. If the document-centric nature of Web Services is used in implementations, Web Services can outperform other traditional implementations.

The large amount of XML metadata contained in SOAP messages is the main reason that Web Services will require more network bandwidth and CPU times than RMI. All numeric and other data in Web Services are converted to text. Meta-data, defining structure, are provided as XML mark-up tags. XML parsers allow client and server implementations to construct their distinct but equivalent representations of any data structures. The use of HTTP, and XML text documents, supports increased interoperability but also represents a significant increase in run-time cost for Web

Service solutions. The XML formatted documents are inherently more voluminous than the binary data traffic of the other approaches. More data have to be exchanged across the network, and more control packets are required.

When considering performance alone, Web Services provide value when the overhead of parsing XML and SOAP is outweighed by the business logic or computation performed on the server. Although Web Services generally don’t provide value in performance, but do provide a convenient way to provide user interface, automatic firewall protection, mobility and heterogeneity of applications, transparent proxies, and thin clients. The most natural designs for distributed objects are easy to use but scale poorly, whereas Web Services have good scaling properties.

REFERENCES

Abowd G.D. and Mynatt E.W., “Charting Past, Present, and Future Research in Ubiquitous Computing”, *ACM Transactions on Computer Human Interaction*, vol.7, pp.29-58, March 2000.

ACM SIGCHI Curricula for Human-Computer Interaction.

Bardram J.E., “Applications of Context-Aware Computing in Hospital Work – Examples and Design Principles”, *Proceedings of the 2004 ACM Symposium on Applied Computing 2004 (ACM SAC 2004)*, pp.1574-1579, 2004.

Bardram J.E., “The Java Context Awareness Framework (JCAF) – A Service Infrastructure and Programming Framework for Context-Aware Applications”, *Lecture Notes in Computer Science, Munich Germany, Springer Verlag*, vol.3468, May 2005.

Barkhuus L. and Dourish P., “Everyday Encounters with Context-Aware Computing in a Campus Environment”, *In Proceedings of UbiComp 2004, Nottingham, UK*, 2004

Barkhuus L., Dourish P., “Everyday Encounters with Context-Aware Computing in a Campus Environment”, *ACM Proceedings of the 44th Annual Southeast Regional Conference*, pp.494-499, 2006.

Birman K.P., “Like it or not, web services are distributed objects”, Communications of the ACM, vol.47, pp.60-62, December 2004.

Bluetooth Technology Official Web Site, “Basics”, 2008 http://www.bluetooth.com/Bluetooth/Technology/Basics.htm

Carnegie Mellon Web Site: “Research Areas - Mobile and Pervasive Computing Research in the Computer Science Department at Carnegie Mellon”, 2008 http://www.csd.cs.cmu.edu/research/areas/mopercomp/

Chen S., Yan B., Zic J., Liu R., Ng A., "Evaluation and Modeling of Web Services Performance", *IEEE International Conference on Web Services (ICWS'06)*, 2006.

Cook D.J., Das S.K., “How smart are our environments? An updated look at the state of the art”, *Pervasive and Mobile Computing*, vol.3, pp.53-73, 2007.

Cook W.R., Barfield J., "Web Services versus Distributed Objects: A Case Study of Performance and Interface Design", *Proc. of the IEEE International Conference on Web Services (ICWS)*, 2006.

Davies N. and Gellersen H.W., “Beyond Prototypes: Challenges in Deploying Ubiquitous Systems”, *IEEE Pervasive Computing*, vol.1, pp.26-35, 2002.

Demarey C., Harbonnier G., Rouvoy R., Merle P., “Benchmarking the Round-Trip Latency of Various Java-Based Middleware Platforms”, Studia Informatica Universalis Regular Issue, vol.4, no.1, pp.724, May 2005.

Dey A.K.,“Understanding and Using Context”, *Personal and Ubiquitous Computing,* vol.5 pp.4-7, February 2001.

Edwards W.K., “Discovery Systems in Ubiquitous Computing”, *IEEE Pervasive Computing*, vol.5, pp.70-77, 2006.

Ejigu D., Scuturici M., and Brunie L., “Hybrid Approach to Collaborative Context-Aware Service Platform for Pervasive Computing”, *Journal of Computers*, vol.3, No. 1, January 2008.

Elfwing R. Paulsson U., Lundberg L., “Performance of SOAP in Web Service Environment Compared to CORBA”, Proceedings of the Ninth Asia-Pacific Software Engineering Conference (APSEC’02), pp.84-93, 2002.

Erl, T. Service-Oriented Architecture: Concepts, Technology, and Design, Prentice Hall, 2005.

Erl, T., “Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services”, PrenticeHall, 2004.

Estrin D., Culler D., Pister K., and Sukhatme G., “Connecting the Physical World with Pervasive Networks”, *IEEE Pervasive Computing*, vol.1, no.1, pp.59-69, January 2002.

Gokhale A., Kumar B., Sahuguet A., “Reinventing the Wheel? CORBA vs. Web Services”, Proceedings of the International World Wide Web Conference, May 2002.

Goldstein H., “Mike Villas's World”, *IEEE Spectrum*, vol.41, pp.45-48, July 2004.

Government Technologies Digital Communities, "New MIT iFind Lets Students Locate Other Wi-Fi Laptop Equipped Friends", December 2006.

Gray N.A.B., "Comparison of Web Services, Java-RMI, and CORBA service implementations", *Fifth Australasian Workshop on Software and System Architectures*, 2004.

Griswold W. G., Boyer R., Brown S.W., Truong T.M., Bhasker E., Jay G.R., and Shapiro R.B., “Activecampus - sustaining educational communities through mobile technology”, *Technical Report CS2002-0714, UC San Diego, Department of CSE*, July 2002.

Griswold W.G., Boyer R., Brown S.W., Truong T.M., Bhasker E., Jay G.R., Shapiro R.B., “Using Mobile Technology to Create Opportunitistic Interactions on a University Campus”, *UbiComp 2002 Workshop on Supporting Spontaneous Interaction in Ubiquitous Computing Settings, Technical Report CS2002-0724, Computer Science and Engineering, UC San Diego*, September 2002.

Griswold W.G., Shanahan P., Brown S.W., Boyer R., Ratto M., Shapiro R.B., and Truong T.M., “ActiveCampus Experiments in Community-Oriented Ubiquitous Computing”, *IEEE Computer Society*, pp.73-81, October 2004.

Gümüşkaya H., “An Architecture Design Process Using a Supportable Meta-Architecture and Roundtrip Engineering”, *Lecture Notes in Computer Science (LNCS), Springer-Verlag*, vol.4243, pp.324-333, 2006.

Gümüşkaya H., Gürel A.V., Nural M.V., “Architectures for Small Mobile Communication Devices and Performance Analyses”, *First IEEE International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2008)*, August 2008.

Gümüşkaya H., Hakkoymaz H., “WiPoD Wireless Positioning System Based on 802.11 WLAN Infrastructure”, *Enformatika*, vol.9, pp.126-130, November 2005.

Gümüşkaya H., Nural M.V., “Service-Oriented Context-Awareness and Context-Aware Services”, *Third International Conference on Systems, Computing Sciences and Software Engineering (SCSS2007)*, December 2007.

Henderson T., Kotz D., “Measuring Wireless LANs”, Mobile, Wireless and Sensor Networks: Technology Applications and Future Directions John Wiley & Sons, New York, pp.5-27, 2006.

Hightower J., LaMarca A., Smith I.E., "Practical Lessons from Place Lab," *IEEE Pervasive Computing*, vol. 5, no. 3, pp. 32-39, Jul-Sept, 2006.

Jagannadham D., Ramachandran V., Kumar H.N.H., "Java2 Distributed Application development (Socket, RMI, Servlet, CORBA) approaches, XML-RPC and Web Services Functional analysis and Performance Comparison", *2007 International Symposium on Communications and Information Technologies (ISCIT 2007)*, 2007.

Jones Q., Grandhi S., Terveen L., and Whittaker S., "People-To-People-to-Geographical-Places: The P3 Framework for Location-Based Community Systems", *Journal of Computer Supported Cooperative Work, Kluwer Academic Publishers*, 2004.

Juric M.B., Hericko M., Welzer T., Rozman I., Sasa A., Krisper M., "Web Services and Java Middleware Functional and Performance Analysis for SOA", *2007 Inaugural IEEE International Conference on Digital EcoSystems and Technologies*, pp.217-222, Feb. 2007.

Juric M.B., Rozman I., Brumen B., Colnaric M., Hericko M., "Comparison of performance of Web services, WS-Security, RMI, and RMI–SSL", *Journal of Systems and Software*, vol.79, pp.689-700, May 2006.

Juric M.B., Kezmah B., Hericko M., Rozman I.,Vezocnik I., “Java RMI, RMI Tunneling and Web Services Comparison and Performance Analysis”, *ACM SIGPLAN Notices*, vol.39, pp.58-65, May 2004.

Juric M.B., Rozman I., Nash S., “Java 2 Distributed Object Middleware Performance Analysis and Optimization”, ACM SIGPLAN Notices, vol.35, pp.31-40, August 2000.

Kiener I., “Development of a Location Based Service for a University Campus; Case study of University of New South Wales”, *School of Surveying and Spatial Information Systems, The University of New South Wales, Sydney, Australia*, February 2003.

Kim S., Han S., "Performance comparison of DCOM, CORBA and Web service", The 2006 International Conference on Parallel & Distributed Processing Techniques & Applications, 2006.

Koo S.G.M., Rosenberg C., Chan H.H., and Lee Y.C., “Location-based E-campus Web Services: From Design to Deployment”, *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom 2003)*, 2003.

Kotz D., Essien K., “Analysis of a Campus-Wide Wireless Network”, *Wireless Networks*, vol.11, pp.115-133, January 2005.

Krafzig, D., Banke, K., Slama, D., “Enterprise SOA: Service-Oriented Architecture Best Practices”, Prentice Hall, 2005.

Kumar M., Das S.K., “Pervasive Computing: Enabling Technologies and Challenges”, Handbook of Nature-Inspired and Innovative Computing: Integrating Classical Models with Emerging Technologies, Springer, 2006.

LaMarca A., et. al., “Place Lab: Device Positioning Using Radio Beacons in the Wild” *In Proceedings of Pervasive 2005, Munich, Germany*, May 2005.

Li Y., Hong J.I., and Landay J.A., “ContextMap: Modeling Scenes of the Real World for Context-Aware Computing”, *In Conference Supplement to UbiComp 2003: Poster. Seattle, WA*, October 2003.

Li Y., Hong J.I., Landay J.A., “Design Challenges and Principles for Wizard of Oz Testing of Location Enhanced Applications”, *IEEE Pervasive Computing*, vol.6, pp.70-75, June 2007.

Li Y., Hong J.I., Landay J.A., “Topiary: A Tool for Prototyping Location-Enhanced Applications”, *ACM CHI Letters*, vol.6, no.2, October 2004.

Li Y., Hong J.I., Landay J.A., “Topiary: A Tool for Prototyping Location-Enhanced Applications”, *Symposium on User Interface Software and Technology - UIST'2004, Santa Fe, New Mexico*, 2004.

Li Y., Klemmer S., and Landay J., “Tools for Rapidly Prototyping Mobile Interactions”, *In Handbook of Research on User Interface Design and Evaluation for Mobile Technology*, January 2008.

Mohapatra D., Suma S.B., “Survey of Location Based Wireless Services”, *IEEE International Conference on Personal Wireless Communications (ICPWC 2005)*, pp.358-362, January 2005

Nammour F., Mansour N., "Comparative Evaluation of Object Request Broker Technologies", *ACS/IEEE International Conference on Computer Systems and Applications*, July 2003

Papazoglou, M. P., Heuvel, W. J. van den, “Service-Oriented Architectures: Approaches, Technologies and Research Issues”, Springer The VLDB Journal, vol.16, No.3, pp.389-415, July 2007.

Place Lab Web Site, 2008
www.placelab.org

Robinson P., Vogt H., Wagealla W., "Some Research Challenges in Pervasive Computing", *Springer Netherlands - Privacy, Security and Trust within the Context of Pervasive Computing*, vol.780, pp.1-16, December 2004.

Roman M., Kon F., Campbell R., “Reflective Middleware: From your Desk to your Hand”, *IEEE Distributed Systems Online*, vol.2, pp.5, 2001.

Roussos G., Marsh A.J., Maglavera S., “Enabling Pervasive Computing with Smart Phones”, *IEEE Pervasive Computing*, vol.4, pp.20-27, 2005.

Saha D., Mukherjee A., “Pervasive Computing: A Paradigm for the 21st Century”, *IEEE Computer, IEEE Computer Society Press*, pp. 25-31, March 2003.

Satyanarayanan M., “Pervasive Computing: Vision and Challenges”, *IEEE Personal Communications*, vol.8, pp.10-17, August 2001.

Schechter B.,”Seeing the light: IBM’s vision of life beyond the PC”, 1999.

Schillit B.N., “Mega-Utilities Drive Invisible Technologies”, *IEEE Computer*, vol.36, pp.97-99, February 2003.

Schwab D., and Bunt R., “Understanding the Use of a Campus Wireless Network”, Mobile, Wireless, and Sensor Networks Technology, Applications, and Future Directions, IEEE Press Wiley Interscience, 2006.

Schwab D., Bunt R. , “Understanding the Use of a Campus Wireless Network”, Mobile, Wireless and Sensor Networks: Technology Applications and Future Directions John Wiley & Sons, New York, pp.29-43, 2006.

Syukur E., and Loke S.W., "MHS Learning Services for Pervasive Campus Environments", Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07), 2007.

Vinge V., “Synthetic Serendipity”, *IEEE Spectrum*, vol.41, pp.35-44, July 2004.

Vogels W., “Web Services are not Distributed Objects”, IEEE Internet Computing, vol.7, pp.59-66, December 2003.

Want R., Pering T., Borriello G., and Farkas K.I., “Disappearing Hardware”, *IEEE Pervasive Computing*, vol.1, pp.36-47, 2002.

Weiser M., "The Future of Ubiquitous Computing on Campus," *Communications of the ACM*, vol.41, pp.41-42, 1998.

Weiser M., “The Computer for the 21st Century”, *Scientific American*, vol.265, no. 3, pp. 94-104, September 1991.

Weiser M., Gold R., and Brown J.S., “The Origins of Ubiquitous Computing Research at PARC in the Late 1980s”, *Systems Journal*, vol.38, pp.693-696, 1999.

Wikipedia Web Site, “Bluetooth”, 2008
http://en.wikipedia.org/wiki/Bluetooth

Wikipedia Web Site, “Embedded Systems”, 2008
http://en.wikipedia.org/wiki/Embedded_systems

Wikipedia Web Site, “Human-Computer Interaction”, 2008
http://en.wikipedia.org/wiki/Human_computer_interaction

Wikipedia Web Site, “Mobile Computing”, 2008
http://en.wikipedia.org/wiki/Mobile_computing

Wikipedia Web Site, “Ubiqutious Computing”, 2008
http://en.wikipedia.org/wiki/Ubiquitous_computing

Wikipedia Web Site, “Wi-Fi”, 2008
http://en.wikipedia.org/wiki/Wifi

Yurday B., Gümüşkaya H., “A Service Oriented Reflective Wireless Middleware”, *Lecture Notes in Computer Science (LNCS), Springer-Verlag*, vol.4294, pp.545-556, 2006.